Symbolic Computation and Theorem Proving in Program Analysis

Laura Kovács

Chalmers
Outline

Part 1: Weakest Precondition for Program Analysis and Verification

Part 2: Polynomial Invariant Generation (TACAS’08, LPAR’10)

Part 3: Quantified Invariant Generation (FASE’09, MICAI’11)

Part 4: Invariants, Interpolants and Symbol Elimination (CADE’09, POPL’12, APLAS’12)
Part 1: Program Analysis and Verification

Preliminaries

Weakest Precondition (WP) and Loop Invariants

Examples of Verification by WP
Preliminaries

Program Verification:

program satisfies its requirements (specification)

Precondition $P$: $(x \geq 0) \land (y > 0)$

Postcondition $Q$: $(quo \ast y + rem = x) \land (0 \leq rem < y)$

Program (code) $S$: $quo := 0; rem := x;$
while $y \leq rem$ do
$rem := rem - y; quo := quo + 1$
end while

Hoare triple (correctness formula): $\{P\} \ S \ {Q\}$
Preliminaries

Program Verification:
program satisfies its requirements (specification)

Example.
Given two natural numbers $x$ and $y$, with $y$ being non zero, compute the quotient ($quo$) and the remainder ($rem$) of the integer division of $x$ by $y$.

Precondition $P$: $(x \geq 0) \land (y > 0)$
Postcondition $Q$: $(quo \cdot y + rem = x) \land (0 \leq rem < y)$

Program (code) $S$: $quo := 0; rem := x$; $\text{while } y \leq rem \text{ do}$ $rem := rem - y; quo := quo + 1$ $\text{end while}$

Hoare triple (correctness formula): $\{ P \} S \{ Q \}$
Preliminaries

Program Verification:
program satisfies its requirements (specification)

Example.
Given two natural numbers $x$ and $y$, with $y$ being non zero, compute the quotient ($quo$) and the remainder ($rem$) of the integer division of $x$ by $y$.

Precondition $P$: $(x \geq 0) \land (y > 0)$
Initial states

Postcondition $Q$: $(quo \times y + rem = x) \land (0 \leq rem < y)$
Final states

Program (code) $S$: 

```
quo := 0; rem := x;
while y \leq rem do
    rem := rem − y; quo := quo + 1
end while
```

How

Hoare triple (correctness formula): $\{P\} S \{Q\}$
Program Verification:
program satisfies its requirements (specification)

Example.
Given two natural numbers \(x\) and \(y\), with \(y\) being non zero, compute the quotient \((quo)\) and the remainder \((rem)\) of the integer division of \(x\) by \(y\).

Precondition \(P\): \((x \geq 0) \land (y > 0)\)  
Postcondition \(Q\): \((\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem} < y)\)

Program (code) \(S\):

\[
\text{quo} := 0; \text{rem} := x;
\text{while } y \leq \text{rem} \text{ do}
\quad \text{rem} := \text{rem} - y; \text{quo} := \text{quo} + 1
\text{end while}
\]

Hoare triple (correctness formula): \(\{P\} \ S \ \{Q\}\)
Preliminaries

Program Verification:
program satisfies its requirements (specification)

Example.
Given two natural numbers \( x \) and \( y \), with \( y \) being non zero, compute the quotient \((quo)\) and the remainder \((rem)\) of the integer division of \( x \) by \( y \).

Precondition \( P: (x \geq 0) \land (y > 0) \) initial states
Postcondition \( Q: (quo \times y + rem = x) \land (0 \leq rem < y) \) final states

Program (code) \( S: \)
\[
\begin{align*}
&quo := 0; rem := x; \\
&while y \leq rem do \\
&\quad rem := rem - y; quo := quo + 1 \\
&end while
\end{align*}
\]

How

Hoare triple (correctness formula): \( \{P\} \ S \ \{Q\} \)
Preliminaries

Program Verification:

Program satisfies its requirements (specification \(P, Q\))

Program correctness

Example.

Given two natural numbers \(x\) and \(y\), with \(y\) being non zero, compute the quotient \((quo)\) and the remainder \((rem)\) of the integer division of \(x\) by \(y\).

Precondition \(P\): \((x \geq 0) \land (y > 0)\)

Postcondition \(Q\): \((quo \times y + rem = x) \land (0 \leq rem < y)\)

Program (code) \(S\):

\[
\begin{align*}
quo & := 0; rem := x; \\
\text{while } y \leq rem & \text{ do} \\
rem & := rem - y; quo := quo + 1 \\
\text{end while}
\end{align*}
\]

Hoare triple (correctness formula): \(\{P\} S \{Q\}\)
Considerations

Program statements:

- Assignments: $x := \text{expression}$
- Sequencing: $s_1; s_2$
- Conditionals: if $(\text{cond})$ then $s_1$ else $s_2$
- Loops: while $(\text{cond})$ do $s$ end while

Program: $S = s_1; s_2; \ldots; s_{n-1}; s_n$

Partial correctness of \{P\} $S$ \{Q\}:
Every computation of $S$ that:

- starts in a state satisfying $P$ and
- is terminating,
ends in a state satisfying $Q$. 
Considerations

Program statements:

- Assignments: \( x := expression \)
- Sequencing: \( s_1; s_2 \)
- Conditionals: \( \text{if } (cond) \text{ then } s_1 \text{ else } s_2 \)
- Loops: \( \text{while } (cond) \text{ do } s \text{ end while} \)

Program: \( S = s_1; s_2; \ldots; s_{n-1}; s_n \)

Partial correctness of \( \{P\} S \{Q\} \):
Every computation of \( S \) that:
- starts in a state satisfying \( P \) and
- is terminating,
ends in a state satisfying \( Q \).
Considerations

Program statements:

- Assignments: $x := \text{expression}$
- Sequencing: $s_1; s_2$
- Conditionals: if $(\text{cond})$ then $s_1$ else $s_2$
- Loops: while $(\text{cond})$ do $s$ end while

Program: $S = s_1; s_2; \ldots; s_{n-1}; s_n$

Partial correctness of $\{P\} \ S \ \{Q\}$:
Every computation of $S$ that:

- starts in a state satisfying $P$ and
- is terminating,
ends in a state satisfying $Q$. 
Weakest Precondition

Specification

Program

Verification Conditions

Proving
Weakest Precondition

Verification Conditions

Proving
Weakest Precondition Strategy

\( P \) is weaker than \( R \) iff \( R \implies P \).

Weakest Precondition \( \text{wp}(S, Q) \) for \( S \) with \( Q \):

for any \( \{ R \} S \{ Q \} \) we have \( R \implies \text{wp}(S, Q) \).

Note: \( \{ \text{wp}(S, Q) \} S \{ Q \} \).

Verification of \( \{ P \} S \{ Q \} \):

\( S = s_1; \ldots; s_{n−1}; s_n \)

1. Compute \( \text{wp}(S, Q) \);
2. Prove \( P \implies \text{wp}(S, Q) \)
**Weakest Precondition Strategy**

\[ P \text{ is weaker than } R \text{ iff } R \implies P. \]

**Weakest Precondition** \( \text{wp}(S, Q) \) for \( S \) with \( Q \):

for any \( \{ R \} \ S \{ Q \} \) we have \( R \implies \text{wp}(S, Q) \).

Note: \( \{ \text{wp}(S, Q) \} \ S \{ Q \} \).

**Verification of \( \{ P \} \ S \{ Q \} \):**

\( S = s_1; \ldots; s_{n-1}; s_n \)

1. Compute \( \text{wp}(S, Q) \);
2. Prove \( P \implies \text{wp}(S, Q) \)
Weakest Precondition Strategy

\( P \) is weaker than \( R \) iff \( R \implies P \).

Weakest Precondition \( \text{wp}(S, Q) \) for \( S \) with \( Q \):

for any \( \{R\} S \{Q\} \) we have \( R \implies \text{wp}(S, Q) \).

Note: \( \{\text{wp}(S, Q)\} S \{Q\} \).

Verification of \( \{P\} S \{Q\} \):

\( S = s_1; \ldots; s_{n-1}; s_n \)

1. Compute \( \text{wp}(S, Q) \);
2. Prove \( P \implies \text{wp}(S, Q) \)

\( s_1; \)
\( \vdots \)
\( s_{n-1}; \)
\( s_n \)
\( \{Q\} \)
Weakest Precondition Strategy

\[ P \text{ is weaker than } R \iff R \implies P. \]

Weakest Precondition \( \text{wp}(S, Q) \) for \( S \) with \( Q \):

for any \( \{ R \} S \{ Q \} \) we have \( R \implies \text{wp}(S, Q) \).

Note: \( \{ \text{wp}(S, Q) \} S \{ Q \} \).

Verification of \( \{ P \} S \{ Q \} \):

\( S = s_1; \ldots; s_{n-1}; s_n \)

1. Compute \( \text{wp}(S, Q) \);
2. Prove \( P \implies \text{wp}(S, Q) \)

\[ s_1; \]
\[ \vdots \]
\[ s_{n-1}; \]
\[ s_n \leftarrow \text{wp}(s_n, Q) \]
\[ \{ Q \} \]
Weakest Precondition Strategy

\( P \) is weaker than \( R \) iff \( R \implies P \).

Weakest Precondition \( \text{wp}(S, Q) \) for \( S \) with \( Q \):

for any \( \{ R \} S \{ Q \} \) we have \( R \implies \text{wp}(S, Q) \).

Note: \( \{ \text{wp}(S, Q) \} S \{ Q \} \).

Verification of \( \{ P \} S \{ Q \} \):

\( S = s_1; \ldots; s_{n-1}; s_n \)

1. Compute \( \text{wp}(S, Q) \);

2. Prove \( P \implies \text{wp}(S, Q) \)

\begin{align*}
\text{wp}(S, Q) & \quad \leftarrow \text{wp}(s_{n-1}, \text{wp}(s_n, Q)) \\
\text{wp}(s_n, Q) & \quad \leftarrow \text{wp}(s_n, Q)
\end{align*}
Weakest Precondition Strategy

*P* is weaker than *R* iff \( R \implies P \).

Weakest Precondition \( \text{wp}(S, Q) \) for *S* with *Q*:

for any \( \{ R \} \ S \{ Q \} \) we have \( R \implies \text{wp}(S, Q) \).

Note: \( \{ \text{wp}(S, Q) \} \ S \{ Q \} \).

Verification of \( \{ P \} \ S \{ Q \} \):

\( S = s_1; \ldots; s_{n-1}; s_n \)

1. Compute \( \text{wp}(S, Q) \);
2. Prove \( P \implies \text{wp}(S, Q) \)
WP Inference Rules

► Assignments:

\[ wp(x := \text{expression}, \ Q) = Q_x \leftarrow \text{expression} \]

\[ wp(x := 5, \ x + y = 6) = 5 + y = 6 \]

\[ wp(x := x + 1, \ x + y = 6) = x + 1 + y = 6 \]

► Sequencing:

\[ wp(s_1; s_2, \ Q) = wp(s_1, wp(s_2, \ Q)) \]

\[ wp(x := x + 1; y := y + x, \ 2 + y > 10) = wp(x := x + 1, wp(y := y + x, \ 2 + y > 10)) \]

\[ = wp(x := x + 1, 2 \times (y + x) > 10) \]

\[ = 2 \times (y + x + 1) > 10 \]
WP Inference Rules

- **Assignments:**

  
  \[
  \begin{align*}
  \text{wp}(x := \text{expression}, \ Q) & = Q_{x \leftarrow \text{expression}} \\
  \text{wp}(x := 5, \ x + y = 6) & = 5 + y = 6 \\
  \text{wp}(x := x + 1, \ x + y = 6) & = x + 1 + y = 6
  \end{align*}
  \]

- **Sequencing:**

  
  \[
  \begin{align*}
  \text{wp}(s_1; s_2, \ Q) & = \text{wp}(s_1, \ \text{wp}(s_2, \ Q)) \\
  \text{wp}(x := x + 1; y := y + x, \ 2 \cdot y > 10) & = \text{wp}(x := x + 1, \ \text{wp}(y := y + x, \ 2 \cdot y > 10)) \\
  & = \text{wp}(x := x + 1, \ 2 \cdot (y + x) > 10) \\
  & = 2 \cdot (y + x + 1) > 10
  \end{align*}
  \]
WP Inference Rules

- **Assignments:**
  \[
  \text{wp}(x := \text{expression}, \ Q) = Q_{x\leftarrow\text{expression}} \\
  \text{wp}(x := 5, \ x + y = 6) = 5 + y = 6 \\
  \text{wp}(x := x + 1, \ x + y = 6) = x + 1 + y = 6
  \]

- **Sequencing:**
  \[
  \text{wp}(s_1; \ s_2, \ Q) = \text{wp}(s_1, \ \text{wp}(s_2, \ Q)) \\
  \text{wp}(x := x + 1; \ y := y + x, \ 2 \times y > 10) = \text{wp}(x := x + 1, \ \text{wp}(y := y + x, \ 2 \times y > 10)) \\
  \quad = \text{wp}(x := x + 1, \ 2 \times (y + x) > 10) \\
  \quad = 2 \times (y + x + 1) > 10)
  \]
WP Inference Rules

- **Assignments:**
  \[ wp(x := \text{expression}, Q) = Q_{x \leftarrow \text{expression}} \]
  \[ wp(x := 5, x + y = 6) = 5 + y = 6 \]
  \[ wp(x := x+1, x + y = 6) = x+1 + y = 6 \]

- **Sequencing:**
  \[ wp(s_1; s_2, Q) = wp(s_1, wp(s_2, Q)) \]
  \[ wp(x := x + 1; y := y + x, 2 \times y > 10) = wp(x := x + 1, wp(y := y + x, 2 \times y > 10)) \]
  \[ = wp(x := x + 1, 2 \times (y + x) > 10) \]
  \[ = 2 \times (y + x + 1) > 10 \]
WP Inference Rules

- Assignments:

  \[ \text{wp}(x := \text{expression}, \; Q) = Q_{x \leftarrow \text{expression}} \]

  \[ \text{wp}(x := 5, \; x + y = 6) = 5 + y = 6 \]

  \[ \text{wp}(x := x + 1, \; x + y = 6) = x + 1 + y = 6 \]

- Sequencing:

  \[ \text{wp}(s_1; \; s_2, \; Q) = \text{wp}(s_1, \; \text{wp}(s_2, \; Q)) \]

  \[ \text{wp}(x := x + 1; \; y := y + x, \; 2 \cdot y > 10) = \text{wp}(x := x + 1, \; \text{wp}(y := y + x, \; 2 \cdot y > 10)) \]

  \[ = \text{wp}(x := x + 1, \; 2 \cdot (y + x) > 10) \]

  \[ = 2 \cdot (y + x + 1) > 10 \]
WP Inference Rules

- **Conditionals:**

\[
wp(\text{if } \text{cond then } s_1 \text{ else } s_2, \ Q) = (\text{cond } \implies \ wp(s_1, \ Q)) \land (\neg \text{cond } \implies \ wp(s_2, \ Q))
\]

and, if \( s_1, s_2 \) contain loops, the verification conditions:

\[
\begin{align*}
\text{cond} & \implies \text{VerifConditions}[s_1, \ Q] \\
\neg \text{cond} & \implies \text{VerifConditions}[s_2, \ Q]
\end{align*}
\]

\[
wp(\text{if } x \geq y \text{ then } m := x \text{ else } m := y, \ m = \text{Max}[x, y]) =
\begin{align*}
(x \geq y & \implies wp(m := x, \ m = \text{Max}[x, y])) \land (x < y & \implies wp(m := y, \ m = \text{Max}[x, y])) = \\
(x \geq y & \implies x = \text{Max}[x, y]) \land (x < y & \implies y = \text{Max}[x, y])
\end{align*}
\]
WP Inference Rules

- **Conditionals:**

\[
wp(\text{if } \text{cond} \text{ then } s_1 \text{ else } s_2, \ Q) = \ (\text{cond} \implies \ wp(s_1, \ Q)) \land (\neg \text{cond} \implies \ wp(s_2, \ Q))
\]

and, if \( s_1, \ s_2 \) contain loops, the verification conditions:

- \( \text{cond} \implies \ \text{VerifConditions}[s_1, \ Q] \)
- \( \neg \text{cond} \implies \ \text{VerifConditions}[s_2, \ Q] \)

\[
wp(\text{if } x \geq y \text{ then } m := x \text{ else } m := y, \ m = \text{Max}[x, y]) = \\
(x \geq y \implies wp(m := x, \ m = \text{Max}[x, y])) \land (x < y \implies wp(m := y, \ m = \text{Max}[x, y])) = \\
(x \geq y \implies x = \text{Max}[x, y]) \land (x < y \implies y = \text{Max}[x, y])
\]
WP Inference Rules

▶ Conditionals:

\[ \text{wp}(\text{if } \text{cond then } s_1 \ \text{else} \ s_2, \ Q) = (\text{cond} \Rightarrow \text{wp}(s_1, \ Q)) \land (\neg \text{cond} \Rightarrow \text{wp}(s_2, \ Q)) \]

and, if \( s_1, \ s_2 \) contain loops, the verification conditions:

\[ \text{cond} \Rightarrow \text{VerifConditions}[s_1, \ Q] \]
\[ \neg \text{cond} \Rightarrow \text{VerifConditions}[s_2, \ Q] \]

\[ \text{wp}(\text{if } x \geq y \ \text{then} \ m := x \ \text{else} \ m := y, \ m = \text{Max}[x, y]) = \]
\[ (x \geq y \Rightarrow \text{wp}(m := x, \ m = \text{Max}[x, y])) \land (x < y \Rightarrow \text{wp}(m := y, \ m = \text{Max}[x, y])) = \]
\[ (x \geq y \Rightarrow x = \text{Max}[x, y]) \land (x < y \Rightarrow y = \text{Max}[x, y]) \]
WP Inference Rules

- **Conditionals:**

\[
wp(\text{if } \text{cond then } s_1 \text{ else } s_2, \ Q) = (\text{cond } \implies \ wp(s_1, \ Q)) \land (\neg\text{cond } \implies \ wp(s_2, \ Q))
\]

and, if \(s_1, s_2\) contain loops, the verification conditions:

\[
\text{cond } \implies \ \text{VerifConditions}[s_1, \ Q]
\]
\[
\neg\text{cond } \implies \ \text{VerifConditions}[s_2, \ Q]
\]

\[
wp(\text{if } x \geq y \text{ then } m := x \text{ else } m := y, \ m = \text{Max}[x, y]) =
\]
\[
(x \geq y \implies wp(m := x, m = \text{Max}[x, y])) \land (x < y \implies wp(m := y, m = \text{Max}[x, y])) =
\]
\[
(x \geq y \implies x = \text{Max}[x, y]) \land (x < y \implies y = \text{Max}[x, y])
\]
WP Inference Rules

▷ Conditionals:

\[ wp(\text{if } \text{cond } \text{then } s_1 \text{ else } s_2, \ Q) = (\text{cond } \implies wp(s_1, \ Q)) \land (\neg\text{cond } \implies wp(s_2, \ Q)) \]

and, if \( s_1, s_2 \) contain loops, the verification conditions:

\[ \text{cond } \implies \text{VerifConditions}[s_1, Q] \]
\[ \neg\text{cond } \implies \text{VerifConditions}[s_2, Q] \]

\[
wp(\text{if } x \geq y \text{ then } m := x \text{ else } m := y, \ m = \text{Max}[x, y]) = \]
\[ (x \geq y \implies wp(m := x, \ m = \text{Max}[x, y])) \land (x < y \implies wp(m := y, \ m = \text{Max}[x, y])) = \]
\[ (x \geq y \implies x = \text{Max}[x, y]) \land (x < y \implies y = \text{Max}[x, y]) \]
WP Inference Rules

- Loops:

\[
wp(\text{while } \text{cond} \text{ do } s \text{ end while}, \ Q) = /
\]
WP Inference Rules

- Loops:

\[
wp(\text{while } \text{cond } \text{do } s \text{ end while}, \ Q) = I
\]

where \( I \) is a loop invariant

1. \( I \land \text{cond} \implies I', \) where \( I' = wp(S, I); \)

2. \( I \land \neg \text{cond} \implies Q. \)
WP Inference Rules

Loops:

\[ \text{wp}(\text{while } \text{cond} \text{ do } s \text{ end while}, \ Q) = I \]

where \( I \) is a loop invariant

1. \( I \land \text{cond} \implies I' \), where \( I' = \text{wp}(S, I) \);  
2. \( I \land \neg \text{cond} \implies Q \).

**Loop Invariants (Inductive Assertions):**

evaluate to true before and after each loop iteration

\( I \) is an invariant for \( \{P\} \text{ while } \text{cond} \text{ do } S \text{ end while} \{Q\} \) iff:

0. initial condition: \( P \implies I \);  
1. iterative (inductive) condition: \( \{I \land \text{cond}\} S \{I\} \);  
2. final condition: \( I \land \neg \text{cond} \implies Q \)
WP Inference Rules

▶ Loops:

\[
wp(\text{while } \text{cond} \text{ do } s \text{ end while}, \ Q) = I
\]

where \( I \) is a loop invariant

1. \( I \land \text{cond} \implies I' \), where \( I' = wp(S, I) \);
2. \( I \land \lnot \text{cond} \implies Q \).

**LOOP INVARIANTS (INDUCTIVE ASSERTIONS):**

evaluate to true before and after each loop iteration

\( I \) is an invariant for \( \{P\} \text{ while } \text{cond} \text{ do } S \text{ end while} \{Q\} \) iff:

0. initial condition: \( P \implies I \);
1. iterative (inductive) condition: \( \{I \land \text{cond}\} S \{I\} \);
2. final condition: \( I \land \lnot \text{cond} \implies Q \)
WP Inference Rules

Loops:

\[ \text{wp(while cond do } s \text{ end while, } Q) = I \]

and verification conditions:

1. \( I \land \text{cond} \implies I' \), where \( I' = \text{wp}(S, I) \);
2. \( I \land \neg\text{cond} \implies Q \).

**Loop Invariants (Inductive Assertions):**

evaluate to true before and after each loop iteration

\( I \) is an invariant for \( \{P\} \text{ while cond do } S \text{ end while } \{Q\} \) iff:

0. initial condition: \( P \implies I \);
1. iterative (inductive) condition: \( \{I \land \text{cond}\} S \{I\} \);
2. final condition: \( I \land \neg\text{cond} \implies Q \)
WP Inference Rules

- Loops:
  \[ \wp(\text{while } \text{cond} \text{ do } s \text{ end while}, Q) = I \]

  and verification conditions:

1. \( I \land \text{cond} \implies I' \), where \( I' = \wp(S, I) \);
2. \( I \land \neg \text{cond} \implies Q \).

Division Example (revisited):
Postcondition \( Q \): \((\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem} < y)\)

Loop \( \text{DivLoop} \): \( \text{assume} (\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \)
  \( \text{while } y \leq \text{rem} \text{ do } \)
    \( \text{rem} := \text{rem} - y; \text{quo} := \text{quo} + 1 \)
  \( \text{end while} \)

\[ \wp(\text{DivLoop}, Q) = \left( \text{quo} \times y + \text{rem} = x \right) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \]

\[ I \land (y \leq \text{rem}) \implies \left( (\text{quo} + 1) \times y + (\text{rem} - y) = x \right) \land (0 \leq \text{rem} - y) \land (0 < y) \land (x \geq 0) \]

\[ I \land (y > \text{rem}) \implies Q \]
WP Inference Rules

- Loops:

\[ \text{wp}\left( \text{while cond do } s \text{ end while}, \ Q \right) = I \]

and verification conditions:

1. \( I \land \text{cond} \implies I' \), where \( I' = \text{wp}(S, I) \);

2. \( I \land \neg\text{cond} \implies Q \).

Division Example (revisited):

Postcondition \( Q \): \((\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem} < y)\)

Loop \( \text{DivLoop} \): 

\begin{align*}
\text{assume} & \quad (\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \\
\text{while} & \quad y \leq \text{rem} \text{ do} \\
& \quad \text{rem} := \text{rem} - y; \quad \text{quo} := \text{quo} + 1 \\
\text{end while}
\end{align*}

\[ \text{wp}(\text{DivLoop}, Q) = (\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \]

\[ I \land (y \leq \text{rem}) \implies ((\text{quo} + 1) \times y + (\text{rem} - y) = x) \land (0 \leq \text{rem} - y) \land (0 < y) \land (x \geq 0) \]

\[ I \land (y > \text{rem}) \implies Q \]
WP Inference Rules

- Loops:
  \[
  \text{wp}(\text{while } \text{cond} \text{ do } s \text{ end while}, Q) = I
  \]
  and verification conditions:
  1. \( I \land \text{cond} \implies I', \) where \( I' = \text{wp}(S, I); \)
  2. \( I \land \neg\text{cond} \implies Q. \)

Division Example (revisited):

Postcondition \( Q: (\text{quo} \ast y + \text{rem} = x) \land (0 \leq \text{rem} < y) \)

Loop \( \text{DivLoop}: \text{assume} (\text{quo} \ast y + \text{rem} = x) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \)
while \( y \leq \text{rem} \) do
  \( \text{rem} := \text{rem} - y; \) \( \text{quo} := \text{quo} + 1 \)
end while

\[
\text{wp}((\text{DivLoop}, Q) = \underbrace{(\text{quo} \ast y + \text{rem} = x) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0)}_{I}
\]

\( I \land (y \leq \text{rem}) \implies ((\text{quo} + 1) \ast y + (\text{rem} - y) = x) \land (0 \leq \text{rem} - y) \land (0 < y) \land (x \geq 0) \)

\( I \land (y > \text{rem}) \implies Q \)
Weakest Precondition Strategy (revised)

Verification of \( \{P\} S \{Q\} \):

\[ S = s_1; \ldots; s_{n-1}; s_n \]

1. Compute \( \text{wp}(S, Q) \);

2. Prove \( P \implies \text{wp}(S, Q) \) and additional verification conditions
Examples of Verification by WP (1)

Example (Division.)
Verify the partial correctness of the annotated \( \{ P \} \ S \{ Q \} \), where:

\[ P: (x \geq 0) \land (y > 0) \]
\[ Q: (\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem} < y) \]

Annotated \( S \):
\[
\text{quo} := 0; \quad \text{rem} := x; \\
\text{invariant} (\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \\
\text{while } y \leq \text{rem} \text{ do} \\
\quad \text{rem} := \text{rem} - y; \quad \text{quo} := \text{quo} + 1 \\
\text{end while}
\]

Verification Conditions:
\[
(x \geq 0) \land (y > 0) \implies \\
(x = x) \land x \geq 0 \land x \geq 0 \land y > 0
\]
\[
(x = \text{rem} + y \times \text{quo}) \land x \geq 0 \land \text{rem} \geq 0 \land y > 0 \land y \leq \text{rem} \implies \\
(x = (\text{rem} - y) + y \times (\text{quo} + 1)) \land x \geq 0 \land \text{rem} - y \geq 0 \land y > 0
\]
\[
(x = \text{rem} + y \times \text{quo}) \land x \geq 0 \land \text{rem} \geq 0 \land y > 0 \land y > \text{rem} \implies \\
(x = \text{rem} + y \times \text{quo}) \land 0 \leq \text{rem} < y
\]
Example (Division.)
Verify the partial correctness of the annotated \( \{ P \} \ S \ \{ Q \} \), where:

\[ P: (x \geq 0) \land (y > 0) \]
\[ Q: (\text{quo} \times y + \text{rem} = x) \land (0 \leq \text{rem} < y) \]

Annotated \( S: \)

\text{quo} := 0; \rem := x;

\text{invariant} \left( \text{quo} \times y + \text{rem} = x \land (0 \leq \text{rem}) \land (0 < y) \land (x \geq 0) \right)

\text{while} y \leq \text{rem} \text{ do}

\text{rem} := \text{rem} - y; \text{quo} := \text{quo} + 1

\text{end while}

Verification Conditions:

\((x \geq 0) \land (y > 0) \implies (x = x) \land x \geq 0 \land x \geq 0 \land y > 0\)

\((x = \text{rem} + y \times \text{quo}) \land x \geq 0 \land \text{rem} \geq 0 \land y > 0 \land y \leq \text{rem} \implies (x = (\text{rem} - y) + y \times (\text{quo} + 1)) \land x \geq 0 \land \text{rem} - y \geq 0 \land y > 0\)

\((x = \text{rem} + y \times \text{quo}) \land x \geq 0 \land \text{rem} \geq 0 \land y > 0 \land y > \text{rem} \implies (x = \text{rem} + y \times \text{quo}) \land 0 \leq \text{rem} < y\)

\((x = \text{rem} + y \times \text{quo}) \land x \geq 0 \land \text{rem} \geq 0 \land y > 0 \land y > \text{rem} \implies (x = \text{rem} + y \times \text{quo}) \land 0 \leq \text{rem} < y\)
Examples of Verification by WP(2)

Example (Cubic Root.)
Verify the partial correctness of the annotated $\{P\} S \{Q\}$, where:

$P$: $a \geq 1$

$Q$: $(r - \frac{1}{2})^3 < a \land (r + \frac{1}{2})^3 > a$

Annotated $S$: $x := a; \ r := q; \ s := 13/4;$

\textit{invariant} $(x \geq 1) \land (s = 3r^2 + \frac{1}{4}) \land (2x = \frac{1}{2} + 2a - \frac{3}{2} r + 3r^2 - 2r^3)$

\textbf{while} $x - s > 0$ \textbf{do}

\hspace{1em} $x := x - s; \ s := s + 6 \ast r + 3; \ r := r + 1$

\textbf{end while}

\textbf{Verification Conditions:}

\[ a \geq 1 \implies \left( \frac{13}{4} = \frac{1}{4} + 3 \right) \land (2a = \frac{1}{2} + 2a - \frac{3}{2} + 3 - 2) \land a \geq 1 \]

\[(x \geq 1) \land (s = 3r^2 + \frac{1}{4}) \land (2x = \frac{1}{2} + 2a - \frac{3}{2} r + 3r^2 - 2r^3) \land x - s > 0 \implies \]

\[(x - s \geq 1) \land (s + 6r + 3 = 3(r + 1)^2 + \frac{1}{4}) \land \]

\[(2(x - s) = \frac{1}{2} + 2q - \frac{3}{2}(r + 1) + 3(r + 1)^2 - 2(r + 1)^3) \]

\[(x \geq 1) \land (s = 3r^2 + \frac{1}{4}) \land (2x = \frac{1}{2} + 2a - \frac{3}{2} r + 3r^2 - 2r^3) \land (x - s) \leq 0 \implies \]

\[(r - \frac{1}{2})^3 < a \land (r + \frac{1}{2})^3 > a \]
Examples of Verification by WP(2)

Example (Cubic Root.)
Verify the partial correctness of the annotated \{P\} S \{Q\}, where:

\( P: a \geq 1 \)
\( Q: (r - \frac{1}{2})^3 < a \land (r + \frac{1}{2})^3 > a \)

Annotated \( S: \ x := a; \ r := q; \ s := 13/4; \)
\begin{align*}
\text{\textbf{invariant}} \quad & (x \geq 1) \land (s = 3r^2 + \frac{1}{4}) \land (2x = \frac{1}{2} + 2a - \frac{3}{2}r + 3r^2 - 2r^3) \\
\text{while } & x - s > 0 \text{ do} \\
\quad & x := x - s; \ s := s + 6 \ast r + 3; \ r := r + 1 \\
\text{end while}
\end{align*}

Verification Conditions:
\( a \geq 1 \implies (\frac{13}{4} = \frac{1}{4} + 3) \land (2a = \frac{1}{2} + 2a - \frac{3}{2} + 3 - 2) \land a \geq 1 \)

\( (x \geq 1) \land (s = 3r^2 + \frac{1}{4}) \land (2x = \frac{1}{2} + 2a - \frac{3}{2}r + 3r^2 - 2r^3) \land x - s > 0 \implies \\
(x - s \geq 1) \land (s + 6r + 3 = 3(r + 1)^2 + \frac{1}{4}) \land \\
(2(x - s) = \frac{1}{2} + 2q - \frac{3}{2}(r + 1) + 3(r + 1)^2 - 2(r + 1)^3) \\

\( (x \geq 1) \land (s = 3r^2 + \frac{1}{4}) \land (2x = \frac{1}{2} + 2a - \frac{3}{2}r + 3r^2 - 2r^3) \land (x - s) \leq 0 \implies \\
(r - \frac{1}{2})^3 < a \land (r + \frac{1}{2})^3 > a \)
End of Session 1

Slides for session 1 ended here . . .