

Software Verification

Partial and Total Correctness

Ahmed Rezine

IDA, Linköpings Universitet

Spring 2025

Hoare Triples and Deductive Reasoning

Weakest preconditions

Further readings

Hoare Triples and Deductive Reasoning

Weakest preconditions




Further readings

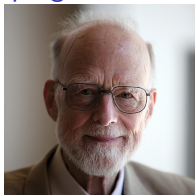
Function Specifications and Correctness

```
method add(x: int, y: int) returns (sum: int)
  requires 0 < x && 0 < y
  ensures  x < sum && y < sum
{
  sum := x + y;
}
```

- ▶ Contract between the caller and the implementation. **Total Correctness** requires that:
 - ▶ if the pre-condition ($0 < x \ \&\& \ 0 < y$) holds
 - ▶ then the implementation terminates,
 - ▶ after termination, the following post-condition holds
($x < \text{sum} \ \&\& \ y < \text{sum}$)
- ▶ **Partial Correctness** does not require termination

Already more than 50 years

-  R. W. Floyd. Assigning meanings to programs. 1968.
-  C. A. R. Hoare. An axiomatic basis for computer programming. 1969.
-  E. W. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. 1975.



Hoare Triples and Partial Correctness

- ▶ a Hoare triple $\{P\} \text{ stmt } \{R\}$ consists in:
 - ▶ a predicate pre-condition P
 - ▶ a program stmt ,
 - ▶ a predicate post-condition R
- ▶ intuitively, $\{P\} \text{ stmt } \{R\}$ holds if whenever P holds and stmt is executed and terminates (**partial correctness**), then R holds after stmt terminates.
(i.e., $(P \text{ and } \text{stmt terminates}) \text{ implies } (R \text{ after termination})$).
- ▶ All following triples hold (i.e., are valid):
 - ▶ $\{true\} x := y \{x = y\}$
 - ▶ $\{x = 1 \wedge y = 2\} x := y \{x = 2\}$
 - ▶ $\{x \geq 1\} y := 2 \{x = 0 \vee y \leq 10\}$
 - ▶ $\{x \geq 1\} \text{ if}(y = 2) \text{ then } x := 0 \{x \geq 0\}$
 - ▶ $\{false\} x := 1 \{x = 2\}$
 - ▶ $\{true\} \text{ abort } \{false\}$
 - ▶ $\{x = 10\} \text{ skip } \{x = 10\}$
 - ▶ $\{i = 0\} \text{ while}(i \neq 10) i := i + 1; \{i = 10\}$
 - ▶ $\{i \geq 0\} \text{ while}(i \neq 10) i := i + 1; \{i = 10\}$

Hoare Triples and Partial Correctness

$$\frac{P' \implies P \quad \{P\} \text{ stmt } \{Q\} \quad Q \implies Q'}{\{P'\} \text{ stmt } \{Q'\}}$$

$$\frac{\{P\} \text{ stmt } \{Q\} \quad \{Q\} \text{ stmt}' \{R\}}{\{P\} \text{ stmt; stmt}' \{R\}}$$

$$\frac{\{P \wedge B\} \text{ stmt } \{Q\} \quad \{P \wedge \neg B\} \text{ stmt}' \{Q\}}{\{P\} \text{ if } B \text{ then stmt else stmt}' \{Q\}}$$

$$\frac{}{\{P[e/x]\} x := e \{P\}}$$

$$\frac{\{P \wedge B\} \text{ stmt } \{P\}}{\{P\} \text{ (while } (B) \{ \text{stmt} \}) \{P \wedge \neg B\}}$$

Hoare Triples and Deductive Reasoning

Weakest preconditions

Further readings

Weakest precondition

- ▶ The **weakest precondition** of a predicate R wrt. a program $stmt$, written $\mathbf{wp}(stmt, R)$, is the union of all preconditions that guarantee termination of $stmt$ and that ensure R holds after its execution.
- ▶ Observe $\{\mathbf{wp}(stmt, R)\} stmt \{R\}$ and $\mathbf{wp}(stmt, R)$ is unique.
- ▶ $\mathbf{wp}(stmt, R)$ transforms predicate R wrt. $stmt$. It is said to be a **predicate transformer**.
- ▶ $\mathbf{wp}(x := x + 1, x \geq 1) = (x \geq 0)$. Observe $(x \geq 5)$, $(x = 6)$, $(x \geq 0 \wedge y = 8)$ are all valid preconditions, but they are not weaker than $x \geq 0$.

Weakest precondition of assignments

- ▶ **wp** ($x := e, R$) = $R[x/e]$ replaces occurrences of x in R by e .
- ▶ examples:
 - ▶ **wp** ($x := 3, x = 5$) = $(x = 5)[x/3] = (3 = 5) = \text{false}$
 - ▶ **wp** ($x := 3, x \geq 0$) = $(x \geq 0)[x/3] = (3 \geq 0) = \text{true}$
 - ▶ **wp** ($x := y + 5, x \geq 0$) = $(x \geq 0)[x/y + 5] = (y + 5 \geq 0)$
 - ▶ **wp** ($x := 5 * y + 2 * z, x + y \geq 0$) = $(x + y \geq 0)[x/5 * y + 2 * z] = (6 * y + 2 * z \geq 0)$

Weakest precondition of sequences

- ▶ Assume a sequence of two instructions $stmt; stmt'$, for example $x := 2 * y; y := x + 3 * y$;
- ▶ the weakest precondition is given by:

$$\begin{aligned} \mathbf{wp}(stmt; stmt', R) &= \mathbf{wp}(stmt, \mathbf{wp}(stmt', R)), \\ &= \mathbf{wp}(x := 2 * y; y := x + 3 * y, y > 10) \\ &= \mathbf{wp}(x := 2 * y, \mathbf{wp}(y := x + 3 * y, y > 10)) \\ &= \mathbf{wp}(x := 2 * y, (y > 10)[y/x + 3 * y]) \\ &= \mathbf{wp}(x := 2 * y, x + 3 * y > 10) \\ &= (x + 3 * y > 10)[x/2 * y] \\ &= (2 * y + 3 * y > 10) \\ &= y > 2 \end{aligned}$$

Weakest precondition of conditionals

- Assume a conditional ($\text{if}(B) \text{ then } stmt \text{ else } stmt'$), for example ($\text{if}(x > y) \text{ then } z := x \text{ else } z := y$)

- The weakest precondition is given by:

$$\left(\begin{array}{l} \mathbf{wp}((\text{if}(B) \text{ then } stmt \text{ else } stmt'), R) \\ = (B \Rightarrow \mathbf{wp}(stmt, R)) \wedge (\neg B \Rightarrow \mathbf{wp}(stmt', R)) \end{array} \right)$$

- For example,

$$\begin{aligned} & \mathbf{wp}((\text{if}(x > y) \text{ then } z := x \text{ else } z := y), z \leq 10) \\ &= (x > y \Rightarrow \mathbf{wp}(z := x, z \leq 10)) \wedge (x \leq y \Rightarrow \mathbf{wp}(z := y, z \leq 10)) \\ &= (x > y \Rightarrow x \leq 10) \wedge (x \leq y \Rightarrow y \leq 10) \end{aligned}$$

- More general:

$$\mathbf{wp} \left(\left(\begin{array}{l} \mathbf{if} \quad B_1 \rightarrow stmt_1 \\ \quad \square \quad B_2 \rightarrow stmt_2 \\ \mathbf{fi} \end{array} \right), R \right)$$

=

$$(B_1 \vee B_2) \wedge (B_1 \Rightarrow \mathbf{wp}(stmt_1, R)) \wedge (B_2 \Rightarrow \mathbf{wp}(stmt_2, R))$$

Examples: weakest preconditions

Show:

1. $\mathbf{wp}(stmt; \text{skip}, P) = \mathbf{wp}(stmt, P)$
2. $stmt_a; stmt_b$ is equivalent to $stmt_c$ where:

$$stmt_a = \begin{array}{l} \mathbf{if} \quad B_1 \rightarrow stmt_1 \\ \quad \square \quad B_2 \rightarrow stmt_2 \\ \mathbf{fi} \end{array}$$
$$stmt_c = \begin{array}{l} \mathbf{if} \quad B_1 \rightarrow stmt_1; stmt_b \\ \quad \square \quad B_2 \rightarrow stmt_2; stmt_b \\ \mathbf{fi} \end{array}$$

3. find the weakest P s.t. $\{P\} stmt \{x = 1\}$ where:

$$stmt = \begin{array}{l} \mathbf{if} \quad T \rightarrow x := 1 \\ \quad \square \quad T \rightarrow x := -1 \\ \mathbf{fi} \end{array}$$

Hoare Triples and Deductive Reasoning

Weakest preconditions

Further readings

Further readings



A. R. Bradley and Z. Manna.

(chap 5-6) *The calculus of computation: decision procedures with applications to verification.*

Springer Science & Business Media, 2007.



C. A. R. Hoare.

An axiomatic basis for computer programming.

Communications of the ACM, 12(10):576–580, 1969.



K. R. M. Leino.

Dafny: An automatic program verifier for functional correctness.

In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 348–370. Springer, 2010.