

SE 504 (Formal Methods and Models)
Spring 2026
HW #2: Proofs of Simple Programs
Due: 6:00pm, Thursday, Feb 19

In each of problems 1 and 2 (which are very easy), prove the given Hoare Triple. Recall that the Hoare Triple Law for the `skip` command is $\{P\} \text{ skip } \{Q\} \equiv [P \Rightarrow Q]$.

1. $\{k \neq 8\} \text{ skip } \{k \neq 8\}$.
2. $\{k > 6 \wedge r = 4\} \text{ skip } \{k \neq 2 \vee k < 4\}$.

In each of Problems 3 and 4, compute the weakest precondition. Recall that

$$[\text{wp.}(x := E).Q \equiv Q(x := E)]$$

3. $\text{wp.}(x := x + 2).((x + 4) \cdot (x - 2) \leq 0)$

(Note that $a \cdot b \leq 0 \equiv (a \leq 0 \wedge b \geq 0) \vee (a \geq 0 \wedge b \leq 0)$)

4. $\text{wp.}(x, y := y - x, x - y).(x \leq y)$

In Problems 5-7, prove the given Hoare Triple. Keep in mind the Hoare Triple Assignment Law:

$$\{P\} x := E \{Q\} \equiv [P \Rightarrow Q(x := E)]$$

In most cases, you will probably want to use the *Assume the Antecedant* approach to prove such an implication.

5. $\{y \leq 4\} x, y := y + 1, (2 * y) - 6 \{x > y\}$
6. $\{\neg z\} x, y := x \wedge z, x \vee y \{z \equiv x \wedge y\}$
7. $\{P \wedge 0 \leq i < n\} i, x := i - 1, x \text{ min } f.i \{P\}$, where $P : x = (\text{min } j \mid i < j < n : f.j)$

Here, `min` is the operator that yields the smaller of its two operands. We use it as a binary infix operator (by placing it between its two operands, just like other arithmetic operators.) Note that `min` lacks an identity element, but it is associative and commutative, so it serves well as a quantifier as long as the quantification's range is not empty. Also, it may help to recall the **Split off term** (8.23) rule from the text by Gries and Schneider. A slightly more general way to state that rule is

Split off term: Provided $a < b$,

$$(\star i \mid a \leq i < b : P) = (\star i \mid a \leq i < b - 1 : P) \star P(i := b - 1)$$

For that matter, we can split off the “first” term rather than the “last”; doing so, we get another version:

$$(\star i \mid a \leq i < b : P) = P(i := a) \star (\star i \mid a + 1 \leq i < b : P)$$

In each of Problems 8-9, calculate an expression E that makes the given Hoare Triple valid. In Problem 9, each occurrence of an upper case C denotes a *ghost variable* (or *rigid variable*, to use the Gries and Schneider terminology (see page 181)), not a program variable. Thus, the expression you give as your answer for E should not include any occurrences of C .

As in some earlier problems, you will want to rely upon the Hoare Triple Assignment Law and the *Assume the Antecedant* approach to proving an implication. But here you also want to “solve for E ”. Take advantage of opportunities to make use of the assumption for the purpose of replacing one expression by another assumed to be equal to it.

8. $\{y = x^2\} x, y := x + 1, y + E \{y = x^2\}$

9. $\{C = m + j\} j, m := j - m, E \{C = m - j\}$

Recall that if **IF** is the program

if B **then** S_1 **else** S_2 **fi**

then $\{P\} \text{IF} \{Q\} \equiv \{P \wedge B\} S_1 \{Q\} \wedge \{P \wedge \neg B\} S_2 \{Q\}$

10. Prove this Hoare Triple:

```
{P : x = X}
if x >= 0 then skip
else x := -x
fi
{Q : x = |X|}
```

The absolute value function is defined to satisfy this condition:

$$(|z| = z \equiv z \geq 0) \wedge (|z| = -z \equiv z \leq 0)$$

11. Prove this Hoare Triple:

```
{P ∧ 0 ≤ k < b.Length}
if b[k] > 0 then k, sum := k + 1, sum + b[k]
else k := k + 1
fi
{P}
```

where $P : \text{sum} = (+i \mid 0 \leq i < k : \text{if}(b[i] > 0, b[i], 0))$

Recall that $\text{if}(p, x, y)$ has value x if p is true and value y if p is false.