Correctness ("Hoare") Triples, pt. 1

CS 536: Science of Programming, Fall 2021

A. Why

• To specify a program's correctness, we need to know its precondition (what must be true before executing it) and its postcondition (what should be true after it).

B. Objectives

At the end of this practice you should be able to

• Recognize syntactically correct correctness triples.
• Say whether a correctness triple is satisfied, given information about whether the current state satisfies the precondition, whether the statement terminates, and if it does, whether the terminating state satisfies the postcondition.

C. Questions

For all the questions below, you can assume (unless otherwise said) that $\sigma \in \Sigma$, not $\Sigma_\bot$. (I.e., we're not trying to start run a program after an infinite loop or runtime failure.)

1. For a loop-free program without runtime errors, is there any difference between partial and total correctness?

2. Say we're given $\sigma \models \{x > 0\} S \{y > x\}$ for all $\sigma$ and we're given a state $\tau$ where $\tau(x) = -3$. Do we know what $S$ will do if we run in $\tau$? Must it terminate? (With or without a runtime error?) Diverge? Must $y > x$ afterwards? How about $y \leq x$?

3. For which $\sigma$ does $\sigma \models \{x > 1\} y := x \cdot x \{y > x\}$ hold? Is this triple valid?

4. For which $\sigma$ does $\sigma \models \{x > 0\} y := x \cdot x \{y > x\}$ hold? Is this triple valid?

5. Under partial correctness, does $\sigma \models \{F\} S \{q\}$ hold for all $\sigma$, $q$, and $S$? What about $\sigma \models \{p\} S \{T\}$? Do these triples say anything interesting about $S$?

6. Repeat the previous question under total correctness: Does $\sigma \models_{\text{tot}} \{F\} S \{q\}$ always hold? Does $\sigma \models_{\text{tot}} \{p\} S \{T\}$? Do these triples say anything interesting about $S$?

For Problems 7 – 14, say for each statement whether it's true or false and give a brief explanation. (Just a sentence or two is fine.) Assume $\sigma \in \Sigma$. (Remember, if $\sigma \models$ any predicate or triple, then $\sigma \neq \bot$.)

7. If $\sigma \models \{p\} S \{q\}$, then $\sigma \models p$.

8. If $\sigma \not\models \{p\} S \{q\}$, then $\sigma \not\models p$.

9. If $M(S, \sigma) \subset \{\bot_d, \bot_e\}$, then $\sigma \models \{p\} S \{q\}$.
10. If \( \sigma \models p \) and \( M(S, \sigma) \cap \{ \bot_d, \bot_e \} \neq \emptyset \), then \( \sigma \not\models_{\text{tot}} \{p\} S \{q\} \).

11. If \( \sigma \models \{p\} S \{q\} \) and \( \sigma \models p \), then every state in \( M(S, \sigma) \) either \( \in \{ \bot_d, \bot_e \} \) or satisfies \( q \).

12. If \( \sigma \models \{p\} S \{q\} \) and \( \sigma \not\models p \), then every state in \( M(S, \sigma) \) is either \( \in \{ \bot_d, \bot_e \} \) or satisfies \( \neg q \).

13. For nondeterministic \( S \), if \( \sigma \not\models \{p\} S \{q\} \), then \( \tau \models \neg q \) for some \( \tau \in M(S, \sigma) \) but it's possible for \( \xi \models q \) for some \( \xi \in M(S, \sigma) \).

14. For nondeterministic \( S \), if \( \sigma \not\models_{\text{tot}} \{p\} S \{q\} \), if \( \bot \not\in M(S, \sigma) \), then \( \tau \models \neg q \) for some \( \tau \in M(S, \sigma) \) but it's possible for \( \xi \models q \) for some \( \xi \in M(S, \sigma) \).

15. Let \( S = x := x \ast x; y := y \ast y \) and let \( \sigma(x) = \alpha \) and \( \sigma(\xi) = \beta \). Verify that \( \sigma \models \{x > y > 0\} S \{x > y > 0\} \).
   I.e., assume \( \sigma \) satisfies the precondition, calculate \( M(S, \sigma) \), and verify that \( M(S, \sigma) \setminus \bot \) satisfies the postcondition.