

Program Syntax & Semantics; Simple Correctness Triples

CS 536 Homework 2, Due Wed Feb 22, 2012

A. Why

- Our simple programming language is a model for the kind of constructs seen in actual languages.
- Our programs stand for state transformers.
- Correctness triples are how we characterize a program specification.

B. Objectives

- At the end of this homework you should
- Be able to translate simple programs into our language
- Be able to calculate the meaning of small programs
- Be able to explain under what conditions a correctness triple is satisfied or valid, for partial or total correctness.
- Be able to modify a correctness triple to make it valid.

C. Questions [100 points total]

1. [7 points] Translate the following C++/Java-like program into our programming language:

```
for (int i = 0, j = n; --j >= 0; ) x[i++] = y[j];
```

2. [7 points] Calculate $M(t := b[i]; b[i] := b[j]; b[j] := t, \sigma)$. Assume $\sigma(b) = \alpha$ where α maps indexes to values and assume the values of i and j under σ are legal indexes for b . Show each M step of the calculation.
3. [7 points] Calculate $M(\mathbf{if } x < 0 \mathbf{ then } y := -x \mathbf{ else } y := x \mathbf{ fi}, \sigma)$
4. [7 points] Calculate $M(y := x; \mathbf{if } x < 0 \mathbf{ then } y := -x \mathbf{ fi}, \sigma)$
5. [6 points] Let S be the statement below. For which σ is $M(S, \sigma)$ defined, and what is it?

```
s := 0; while n >= 0 do s := s+n; n := n-1 od
```

6. [18 = 6 * 3 points] Which of the following statements are false? Given a brief explanation of why you think it's false. (Just a sentence or two is fine.)

- (a) If a program is totally correct (in some given state), then it is partially correct (in that state).
 - (b) If a loop-free program is partially correct, then it is totally correct.
 - (c) If a program is partially correct but not totally correct, it diverges.
 - (d) If $\sigma \models \{p\} S \{q\}$, then $\sigma \models p$.
 - (e) If $\sigma \models \{p\} S \{q\}$, then $M(S, \sigma)$ is either undefined or satisfies q .
 - (f) If $\sigma \models \{p\} S \{q\}$ and $\sigma \not\models p$, then $M(S, \sigma)$ is either undefined or satisfies $\neg q$.
7. [7 points] Let $\sigma(x) = 3$ and $\sigma(y) = 2$. Verify that $\sigma \models \{x > y > 0\} x := x * x; y := y * y \{x > y > 0\}$, by calculating the meaning of the statement and showing it satisfies the postcondition.
 8. [5 points] Say that both $\{x = 1\} S \{q\}$ and $\{x \geq 1\} S \{q\}$ are valid. From the point of view of a user of the program S , which of these is better? Explain briefly.
 9. [5 points] Say that both $\{p\} S \{q_1\}$ and $\{p\} S \{q_1 \vee q_2\}$ are valid. From the point of view of a user of the program S , which of these is better? Explain briefly.

For problems 10 – 13, consider the triple **{true} while $i \neq n$ do $i := i + 1$ od { $i = n$ }**. There are multiple right answers to these problems.

10. [5 points] Find a state that shows that the triple is invalid under partial correctness.
 11. [5 points] Suggest a change to the precondition that will make the triple valid for total correctness (but don't change the program or postcondition).
 12. [5 points] Suggest a change to the program that will make the triple valid for total correctness (but don't change the precondition or postcondition).
 13. [5 points] If we keep the precondition and program fixed but allow changes to the postcondition, the triple will still be invalid for total correctness. Explain why, briefly.
14. [11 points] Give a recursive definition for a predicate function *ReversedSegment*(b, i, j, m) that is true iff the segment $b[i..i+m-1]$ is the reverse of the segment $b[j..j+m-1]$. E.g., if $b = (1, 2, 3, 4, 5, 0, 5, 4, 3, 2, 6)$, then *ReversedSegment*($b, 1, 6, 4$) is true. Include clauses that ensure that all the indexes involved are legal for b . If $m \leq 0$, then *ReversedSegment* should be true. (If you write your definition carefully, then this doesn't need to be a special case.)