Illinois Institute of Technology Homework 4

# Basic Correctness Triples

*CS 536: Science of Programming; Due Mon Sep 26*

9/21: p.2

***A. Why?***

• Correctness triples are how we write a program with its specification.

## B. Outcomes

After this homework, you should be able to

• Identify the properties that imply (and the relationships between) ⊨, ⊭, ⊨*tot* and ⊭*tot* for correctness triples.

## C. Problems [50 points total]

For Problems 1 – 10, write a short answer (a short paragraph at most). These problems are worth 4 points each.

1. If *S* is loop-free, how are  ⊨ {*p*} *S* {*q*} and ⊨*tot* {*p*} *S* {*q*} related?

**Ans:** If S is loop free then  ⊨*tot* {*p*} *S* {*q*} Iff ⊨ {*p*} *S* {*q*}

1. Suppose σ ⊨*tot* {*p*} *S* {T}. Under what conditions can *M*(*S*, σ) = ⊥?

**Ans:** If σ ⊨*tot* {*p*} *S* {T} Then iff *M*(*S*, σ) = ⊥ then it means σ ⊭  p

1. If σ ⊨  {*p*} *S* {*q*}, do we know whether σ ⊨ or ⊭ *p*?

**Ans:** No

1. If σ ≠ ⊥ and σ ⊭  {*p*} *S* {*q*}, do we know whether σ ⊨ or ⊭ *p*? Whether *M*(*S*, σ) = or ≠ ⊥? Whether *M*(*S*, σ) ⊨ or ⊭ *q*?

**Ans:** If σ ⊭  {*p*} *S* {*q*} then σ ⊨ *p* and *M*(*S*, σ) ≠ ⊥ and *M*(*S*, σ) ⊭ *q*

1. If σ ⊨ *p* and σ ⊨  {*p*} *S* {*q*}, do we know whether *M*(*S*, σ) = or ≠ ⊥? Whether *M*(*S*, σ) ⊨ or ⊭ *q*?

**Ans:** If σ ⊨ *p* and σ ⊨  {*p*} *S* {*q*} then either *M*(*S*, σ) =⊥ or *M*(*S*, σ) ≠ ⊥ and *p*⊨ q

1. If σ ⊨ *p* and σ ⊨*tot* {*p*} *S* {*q*}, do we know whether *M*(*S*, σ) = or ≠ ⊥? Whether *M*(*S*, σ) ⊨ or ⊭ *q*?

**Ans:** σ ⊨ *p* and σ ⊨*tot* {*p*} *S* {*q*} then *M*(*S*, σ) ≠ ⊥ and *M*(*S*, σ) ⊨ q

1. If σ ⊨ *p* and σ ⊭*tot* {*p*} *S* {*q*}, do we know whether *M*(*S*, σ) = or ≠ ⊥? Whether *M*(*S*, σ) ⊨ or ⊭ *q*?

**Ans:** If σ ⊨ *p* and σ ⊭*tot* {*p*} *S* {*q*} then either *M*(*S*, σ) = ⊥ or *M*(*S*, σ) ≠ ⊥ and *M*(*S*, σ) ⊭ *q*

1. If σ ≠ ⊥ and σ ⊭tot {p} S {q}, do we know whether M(S, σ) = or ≠ ⊥? Whether M(S, σ) ⊨ or ⊭ q? Whether σ ⊨ or ⊭ {p} S {¬q}?

**Ans:** Ifσ ⊭*tot* {*p*} *S* {*q*} the σ ⊨ *p then* M(S, σ) = ⊥ or M(S, σ) ⊭ q

σ ⊨{p} S {¬q} iff M(S, σ) = ⊥ or M(S, σ) ⊨ ¬q and σ ⊭ {p} S {¬q} Iff M(S, σ) ≠ ⊥ and M(S, σ) ⊭ ¬q

1. If σ ⊨  {*p*} *S* {*q*} and σ ⊭*tot* {*p*} *S* {*q*}, do we know whether *M*(*S*, σ) = or ≠ ⊥? Whether *M*(*S*, σ) ⊨ or ⊭ *q*?

**Ans:** If σ ⊨  {*p*} *S* {*q*} and σ ⊭*tot* {*p*} *S* {*q*} then *M*(*S*, σ) =⊥ but we cant say whether *M*(*S*, σ) ⊨ or ⊭ *q*

1. If σ ≠ ⊥ and σ ⊭  {p} S {q}, do we know whether σ ⊨tot  or ⊭tot {p} S {¬q}?

**Ans:** σ ⊭  {*p*} *S* {*q*} implies that σ ⊨ *p and* M(S, σ) ≠ ⊥ and *M*(*S*, σ) ⊭ *q*

*If M*(*S*, σ) ⊨ ¬*q Then* σ ⊨*tot* {*p*} *S* {¬*q*} *else* σ ⊭tot {p} S {¬q}

1. For Problems 11 – 15, determine the precondition using the (goal-directed, “reverse”) assignment rule. Logical simplification is optional, but if you do it, be sure to maintain logical equivalence. These problems are worth 2 points each.

[Added 9/21]: For each problem, you get the precondition by taking the postcondition and replacing the assigned variable *v* with the expression *e* we'e assigning to it. The general rule is {*P*(*e*)} *v* := *e* {*P*(*v*)}. Also see the section added to Lecture 7.

1. {*p*₁} m := (a+b)/2 {a < m < b}

**Ans:** *p*₁ ≡ a < (a+b)/2  < b

1. {*p*₂} r := 0 {x = 2 ^ r ∧ r ≤ n} (^ means exponentiation)

**Ans:** *p*₂≡ x = 2 ^ 0 ∧ 0 ≤ n

1. {*p*₃} x := 1 {*p*₂} where *p*₂ is from Problem 12.

**Ans:** *p*₃≡ 1 = 2 ^ 0 ∧ 0 ≤ n

1. {*p*₄} r := r+1 {x = 2^r ∧ r ≤ n}

**Ans:** *p*₄≡ x = 2^(r+1)  ∧  r+1  ≤ n

1. *{p₅} x := 2\*x {p₄} where p₄ is from Problem 14.*

**Ans:** *p₅* ≡ *2\*x*  = 2^(r+1) ∧ r+1 ≤ n