Illinois Institute of Technology Homework 6

***Strongest Postconditions; Basic Proof Rules***

*CS 536: Science of Programming Due Wed Oct 19*

# Problems [50 points total]

For Problems 1 – 4, calculate the requested *sp*. Do only syntactic calculations, not semantic manipulations.

1. [4 points] Calculate *sp*(n ≥ 0, k := 0; r := 1)

Ans: *sp*(n ≥ 0, k := 0; r := 1)

= *sp*(*sp*(n ≥ 0, k := 0), r := 1)

= *sp*(n ≥ 0 ∧ k=0, r := 1)

= n ≥ 0 ∧ k=0 ∧ r = 1

1. [4 points] Calculate *sp*(k ≤ n ∧ r = 2^k, k := k+1; r := r+r)

Ans: *sp*(k ≤ n ∧ r = 2^k, k := k+1; r := r+r)

= *sp*(*sp*(k ≤ n ∧ r = 2^k, k := k+1), r := r+r)

= *sp*(k0 ≤ n ∧ r = 2^k0 ∧ k = k0+1, r := r+r)

= k0 ≤ n ∧ r0 = 2^k0 ∧ k = k0+1 ∧ r := r0+r0

1. [4 points] Calculate *sp*(*p*, *S*₁; *S*₂) where *p* ≡ i ≤ j ∧ j-i < n, *S*₁ ≡ i := f(i\*j), and *S*₂ ≡ j := g(i+j).

Ans: *sp*(*p*, *S*₁; *S*₂)

= *sp*(sp(*p*, *S*₁), *S*₂)

sp2= sp(*p*, *S*₁)

= i0 ≤ j ∧ j-i0 < n ∧  i := f(i0 \*j)

*sp*(sp(*p*, *S*₁), *S*₂)

=sp(i0 ≤ j ∧ j-i0 < n ∧  i := f(i0 \*j), j := g(i+j))

= i0 ≤ j0 ∧ j0-i0 < n ∧  i := f(i0 \*j0) ∧  j := g(i+j0)

1. [4 points] Calculate *sp*(Between(L, R), *S*) where

Between(L, R) ≡ 0 ≤ L < M < R < size(b) ∧ b[L] < x ≤ b[R] and

*S* ≡ **if** x ≤ b[M] **then** R := M **else** L := M **fi**

Ans: B= x ≤ b[M]

S1= R := M

S2= L := M

*sp*(Between(L, R), *S*)

= *sp*(Between(L, R) ∧ B, R := M) ∨ *sp*(Between(L, R) ∧ ⌐B, L := M)

= sp(Between(L, R) ∧ x ≤ b[M], R := M) ∨ *sp*(Between(L, R) ∧ b[M] <x, L := M)

= (0 ≤ L < M < R0 < size(b) ∧ b[L] < x ≤ b[R0] ∧ x ≤ b[M] ∧ R := M) ∨ (0 ≤ L0 < M < R < size(b) ∧ b[L0] < x ≤ b[R] ∧ b[M] <x ∧ L := M)

1. [4 points] Calculate *sp*(**T**, y := x; if y < 0 then y := -y fi) and logically simplify the result. (Show the result before and after simplification.)

Ans: *sp*(**T**, y := x; if y < 0 then y := -y fi)

S1= y := x

S2= if y < 0 then y := -y fi

B= y < 0

S3= y := -y

*sp*(**T**, S1; S2)

=sp(sp(T, S1), S2)

=sp(**T** ∧ y := x, S2)

=sp(**T** ∧ y := x ∧ B , S3) ∨ sp(**T** ∧ y := x ∧ ⌐B,Skip)

= (T ∧ x < 0 ∧ y = -x) ∨(T ∧ y = x ∧ y ≥ 0)

Logically simplifying

= (x < 0 ∧ y = -x) ∨(y = x ∧ y ≥ 0)

= (x < 0 ∧ y = -x) ∨(x ≥ 0 ∧ y ≥ x)

= y = |x|

For Problems 6 – 10, fill in the missing parts in the following rule instances.

1. [4 points] {???} x := y+1 {x < y ∨ x < 0} \_\_\_\_\_\_ ???

Ans: backward assignment rule

y +1 < y ∨ y+1 < 0

= F ∨ y+1 < 0

= y+1 < 0

=y< -1

1. [4 points] {???} k := k+1 {x = 2^k} \_\_\_\_\_\_ ???

Ans: backward assignment rule

x = 2^( k+1)

1. [6 points]

{x = 2^k} x := x\*2 {???}

??? → x = 2^(k+1)

{x = 2^k} x := x\*2 {???} # # # # # # \_\_\_\_\_\_ ???

Ans:

{x = 2^k} x := x\*2 {x0=2^k ∧ x=x0\*2 }

x0=2^k ∧ x=x0\*2 → x = 2^(k+1)

{x = 2^k} x := x\*2 { x = 2^(k+1)} weakening pre condition

1. [5 points]

{*p* ∧ b[m] < v} L := m {*p*}

{*p* ∧ b[m] ≥ v} R := m {*p*}

{*p*} ??? {???} conditional

Ans: {*p*} IF b[m] < v **then** L := m **else** R := m **fi** {p}

1. [6 points]

???

{**inv** x = 2^k ∧ k ≤ n}

**while** k < n **do** x := x\*2; k := k+1 **od** loop

{???}

Ans: {n>0}k=0;{n>0 ∧ k=0}

{**inv** x = 2^k ∧ k ≤ n}

**while** k < n **do** x := x\*2; k := k+1 **od** loop

{k=n ∧ x =x^(k-1)}