CS536 HW6

Luping Xue A20453695
Jiaxin He A20450323
Zhenyu Meng A20380646

1. [20 = 10 * 2 points] Consider the program below, which calculates the sum of the first n squares.

$\{n \geqslant 0\}\ S_0$; {inv p} while k < n do S_1 od {s = sum(n)}

where

- $S_0 \equiv$ k := 0; s := k; r := s
- $S_1 \equiv$ r := r+2*k+1; k := k+1; s := s+r
- $p \equiv 0 \leqslant k \leqslant n \wedge s = sum(k^2) \wedge r = k^2$
- $sum(k^2)$ = the sum of $0^2 \cdots k^2$. (Let $sum(k^2) = 0$ if k < 0.)

The formal proof of partial correctness for the program below is incomplete. For parts (a) – (f), give
definitions for p_1 – p_6 . Use substitution notation and (separately) list the results of carrying out the
substitutions. For parts (g) – (j), give definitions for the rule references r_1 – r_4 . Include the line numbers (e.g.
Sequence 1, 2, not just Sequence).

(a) p_1 (b) p_2 (c) p_3 (d) p_4 (e) p_5 (f) p_6
(g) r_1 (h) r_2 (i) r_3 (j) r_4

1. $\{n \geqslant 0\}$ k := 0 $\{p_1\}$ Assignment
2. $\{p_1\}$ s := 0 $\{p_2\}$ Assignment
3. $\{n \geqslant 0\}$ k := 0; s := 0 $\{p_2\}$ Sequence 1, 2
4. $\{p_2\}$ r := 0 $\{p_3\}$ Assignment
5. $p_3 \rightarrow p$ Predicate logic
6. $\{n \geqslant 0\}\ S_0\ \{p_3\}\ r_1$
7. $\{n \geqslant 0\}\ S_0$ {p} r_2
8. $\{p_4\}$ s := s+r {p} Assignment
9. $\{p_5\}$ k := k+1 $\{p_4\}$ Assignment
10. $\{p_5\}$ k := k+1; s := s+r {p} Sequence 9, 8
11. $\{p_6\}$ r := r+2*k+1 $\{p_5\}$ Assignment
12. $\{p_6\}\ S_1$ {p} Sequence 11, 10
13. $p \wedge k < n \rightarrow p_6 \quad r_3$
14. $\{p \wedge k < n\}\ S_1$ {p} r_4
15. {inv p} W $\{p \wedge k \geqslant n\}$ while, 14
    where W $\equiv$ while k < n do S_1 od
16. $p \wedge k \geqslant n \rightarrow s = sum(n)$ Predicate logic
17. {inv p} W {s = sum(n)} Postcondition weakening, 14, 15
18. $\{n \geqslant 0\}\ S_0$; W {s = sum(n)} Sequence, 7, 17

$p_1 \equiv n \geqslant 0 \wedge k = 0$

$p_2 \equiv n \geqslant 0 \wedge k = 0 \wedge s = 0$

$p_3 \equiv n \geqslant 0 \wedge k = 0 \wedge s = 0 \wedge r = 0$

$p_4 \equiv p[s = s+r/s] \equiv 0 \leqslant k \leqslant n \wedge s+r = sum(k^2) \wedge r = k^2$

$p_5 \equiv p_4 [k = k+1/k] \equiv 0 \leqslant k+1 \leqslant n \wedge s+r = sum((k+1)^2) \wedge r = (k+1)^2$

$p_6 \equiv p_5 [r+2*k+1/r] \equiv 0 \leqslant k+1 \leqslant n \wedge s+ r+2*k+1 = sum((k+1)^2) \wedge r+2*k+1 = (k+1)^2$

$r_1 \equiv$ Sequence 3,4

$r_2 \equiv$ Postcondition weakening6,5

$r_3 \equiv$ Predicate logic

$r_4 \equiv$ Precondition strengthening 13,12

2. [12 = 6 * 2 points] Give the full proof outline that corresponds to the proof in problem 1: Insert conditions p, p_1 , p_2 , etc., as necessary.

{n ⩾ 0} k := 0; {⋯} s := k; {⋯} r := s; {⋯}
{inv p} while k < n do
 {⋯}
 {⋯} r := r+2*k+1;
 {⋯} k := k+1;
 {⋯} s := s+r {⋯}
od
{⋯} {s = sum(n)}

{n ⩾ 0} k := 0; {p_1 } s := k; {p_2} r := s; {p_3}
{inv p} while k < n do
  {p ∧ k < n }
  {p_6 } r := r+2*k+1;
  {p_5 } k := k+1;
  {p_4 } s := s+r {p}
  od
  {p ∧ k ⩾ n} {s = sum(n)}

3. [16 points] We will perform different expansions of the minimal outline below into full proof outlines for
partial correctness.

{T} if y ⩾ 0 then x := sqrt(y) fi {y ⩾ 0 → x = sqrt(y)}

a. [10 = 5 * 2 points] Complete the full outline below, which uses wp everywhere to add internal conditions. Feel free to add or remove whitespace.

{T} {⋯}
if y ⩾ 0 then
{⋯} x := sqrt(y) {⋯}
else
{⋯} skip {⋯}

fi {q ≡ y ⩾ 0 → x = sqrt(y)}

{T} {y ⩾ 0 → y ⩾ 0 → sqrt(y) = sqrt(y) ∧ y < 0 → y ⩾ 0 → x = sqrt(y)}
if y ⩾ 0 then
{y ⩾ 0 → sqrt(y) = sqrt(y)} x := sqrt(y) {y ⩾ 0 → x = sqrt(y)}
else
{y ⩾ 0 → x = sqrt(y)} skip {y ⩾ 0 → x = sqrt(y)}
fi {q ≡ y ⩾ 0 → x = sqrt(y)}

b [6 = 3 * 2 points] Complete the full outline below, which uses a mix of wp and sp.
{T} if y ⩾ 0 then
{⋯} {⋯} x := sqrt(y) {q }
else
{⋯} {q } skip {q }
fi
{q ≡ y ⩾ 0 → x = sqrt(y)}

{T} if y ⩾ 0 then
{y ⩾ 0} {y ⩾ 0 ∧ x = sqrt(y) = sqrt(y)} x := sqrt(y) {y ⩾ 0 ∧ x = sqrt(y) }
else
{y < 0} {y ⩾ 0 → x = sqrt(y) } skip {y ⩾ 0 → x = sqrt(y) }
fi
{q ≡ y ⩾ 0 → x = sqrt(y)}

4. [12 = 6 * 2 points] Expand the minimal outline below into a full proof outline for full correctness by giving definitions for p_1 – p_6 . Also list the three predicate logic obligations. List the results of carrying out the substitutions. Hint: It's not always the case that p j+1 is a function of pj
.
{b[j] ⩾ 1} x := 1 {p_1 }; k := 0; {p_2 }
{inv p ≡ 1 ⩽ x = 2^k ⩽ b[j]} {bd b[j] - x}
while 2*x ⩽ b[j] do
{p ∧ p_3 }
{p_4 } k := k+1
{p_5 } x := 2*x
{p ∧ p_6 }
od {p ∧ 2*x ⩽ b[j]}
{x = 2^k ⩽ b[j] < 2^(k+1)}

p_1 ≡ b[j] ⩾ 1 ∧ x = 1
p_2 ≡ b[j] ⩾ 1 ∧ x = 1 ∧ k = 0

$p_3 \equiv 2*x \leqslant b[j]$

$p_6 \equiv p \equiv 1 \leqslant x = 2\text{^}k \leqslant b[j]\}$

$p_5 \equiv p\ [x := 2*x] \equiv 1 \leqslant 2*x = 2\text{^}k \leqslant b[j]\}$

$p_4 \equiv p_5\ [k := k+1] \equiv 1 \leqslant 2*x = 2\text{^}(k+1) \leqslant b[j]\}$

predicate logic obligations:

$p_2 \rightarrow p$

$p_3 \rightarrow p_4$

$\{p \wedge 2*x \leqslant b[j]\} \rightarrow \{x = 2\text{^}k \leqslant b[j] < 2\text{^}(k+1)\}$