CS 536 - Science of Programming

Assignment - 4

1) $(w * x \neq 0 \wedge z \le 2 \rightarrow f(w) > 0$

a) $p[y+z/x] \equiv (w * x \neq 0 \wedge z \le 2 \rightarrow f(w) > 0$
$\wedge \forall x. \exists y. 0 \le y \le x \wedge f(w \div x) + y > f(z)$ $[y+z/x]$

$\equiv w * (y+z) \neq 0 \wedge z \le 2 \rightarrow f(w) > 0 \wedge \forall x.$
$\exists y. 0 \le y \le x \wedge f(w \div x) + y > f(z)$

b) $p[x+z/w] \equiv (w * x \neq 0 \wedge z \le 2 \rightarrow f(w) > 0$
$\wedge \forall x. \exists y. 0 \le y \le x \wedge f(w \div x) + y > f(z)$ $[x+z/w]$

$\equiv (x+z) * x \neq 0 \wedge z \le 2 \rightarrow f(w) > 0 \wedge \forall x.$
$\exists y. 0 \le y \le x \wedge f(w \div x) + y > f(z)$ $[x+z/w]$

$\equiv (x+z) * x \neq 0 \wedge z \le 2 \rightarrow f(x+z) > 0 \wedge$
$\forall x. \exists y. 0 \le y \le x \wedge f((x+z) \div x) + y > f(z)$

c) $p[x+y/z] \equiv (w * x \neq 0 \wedge z \le 2 \rightarrow f(w) > 0 \wedge$
$\forall x. \exists y. 0 \le y \le x \wedge f(w \div x) + y > f(z)$ $[x+y/z]$

$\equiv w * x \neq 0 \wedge (x+y) \le 2 \rightarrow f(w) > 0 \wedge \forall x. \exists y.$
$0 \le y \le x \wedge f(w \div x) + y > f(z)$ $[x+y/z]$

$\equiv w * x \neq 0 \wedge (x+y) \le 2 \rightarrow f(w) > 0 \wedge \forall x. \exists y.$
$0 \le y \le x \wedge f(w \div x) + y > f(x+y)$

2) Given conjecture,

$$(x * y)\{e/x\}\{e'/y\} \equiv (x * y)\{e'/y\}\{e/x\}$$

a) Let us take $e = p$ and $e' = q$

LHS of conjecture :-

$$(x * y)\{e/x\}\{e'/y\} = (x * y)\{p/x\}\{q/y\}$$
$$= (p * q)$$

RHS of conjecture :-

$$(x * y)\{e/x\}\{e'/y\} = (x * y)\{q/y\}\{p/x\}$$

$$= (p * q)$$

so, $(p * q) \equiv (p * q)$

b) Let us take $e = y+1$ and $e' = x+1$

LHS of conjecture :-

$$(x * y)\{e/x\}\{e'/y\} = (x * y)\{y+1/x\}\{x+1/y\}$$
$$= ((y+1) * y) = (((x+1)+1) * x+1)$$
$$= (x+2) * (x+1)$$

RHS of conjecture :-

$$(x * y)\{e'/y\}\{e/x\} = (x * y)\{x+1/y\}\{y+1/x\}$$
$$= (x * (x+1)) = (y+1 * ((y+1)+1))$$

$= (y+1) * (y+2)$

So, $(x+2) * (x+1) \not\equiv (y+1) * (y+2)$

a) $(x * y)\{e/x\}\{e'/y\} \equiv (x * y)\{e'/y\}\{e/x\}$

b) $(x * y)\{e/x\}\{e'/y\} \not\equiv (x * y)\{e'/y\}\{e/x\}$

3)

a) ~~By~~ the definition of wlp,

$\models \{wlp(S, q)\} S \{q\}$

It is given that $P \iff wlp(S, q)$

so, $\models \{P\} S \{q\}$

By the definition of SP,

$\models \{P\} S \{sp(P, S)\}$

so, strengthening a post-condition doesn't affect validity.

so, $sp(P, S) \Rightarrow q$

b) Here, let us take an example to disprove if $P \iff wlp(S, q)$ then $q \Rightarrow sp(P, S)$

Let,

$S \Rightarrow x := x+1$

$P \Rightarrow x = 0$

$q \Rightarrow x \geq 1$

Now, let us calculate $wlp(S,q)$ and $sp(p,S)$

$wlp(S,q)$ is $x \geq 0$

we need to check $p \Leftarrow\Rightarrow wlp(S,q)$

$wlp(S,q) = (x \geq 0)$ when $x = 0$

Now, let us calculate $sp(p,S)$

since, p is $x=0$, $sp(p,S)$ after executing S would be $x=1$.

we need to check $q \Rightarrow sp(p,S)$

Here, q is $x \geq 1$ so while $x=1$ satisfies q it does not imply $q \Rightarrow sp(p,S)$ as q also can include values where $x>1$

$sp(p,S)$ is $x=1$

So, this example can show that $q \Rightarrow sp(p,S)$ does not hold in general. so, disproving the statement.

4)

a) False, Because it also possible that if precondition p satisfies, the program can diverge or create error.

b) False, Beacause if σ satisfies p then $M(S,\sigma) - \perp \models S$.

c) False, Because if $\sigma \models p$ for some σ S may not terminate in $M(S, \sigma)$.

d) True, Because $\sigma \models p$ if the execution reaches S without error.

e) False, Because $\sigma \models \neg p$ this tells σ does not satisfy p initially but this does not imply that after executing S will satisfy $\neg S$.

5) Given,

$IF \equiv$ if $x \geq 0 \to x := y+1; z := x \;\square\; x \leq 0 \to y := x-1; z := y$ fi

calculate $sp(x = y, IF)$

$lhs(S) \equiv \{x, y, z\}$

$rhs(S) \cup free(p) \equiv \{x, y\}$

$aged(p, S) \equiv \{x, y\}$

for arm 1:-

$sp(x = y \wedge x = x_0 \wedge y = y_0 \wedge x \geq 0, x := y+1; z := x)$

$\equiv x_0 = y \wedge x_0 = x_0 \wedge y = y_0 \wedge x_0 \geq 0 \wedge x = y+1 \wedge z = x$

For arm 2 :-

$sp(x=y \wedge x=x_0 \wedge y=y_0 \wedge x \le 0,\ y:=x-1;\ z:=y)$

$\equiv x=y_0 \wedge x=x_0 \wedge y_0=y_0=y_0 \wedge x \le 0 \wedge$
$y=x-1 \wedge z=y$

$sp(p,S) \equiv (x_0=y \wedge x_0=x_0 \wedge y=y_0 \wedge x_0 \ge 0 \wedge$
$x=y+1 \wedge z=x) \vee (x=y_0 \wedge x=x_0 \wedge$
$y_0=y_0 \wedge x \le 0 \wedge y=x-1 \wedge z=y)$

6) Given,

$p \equiv y=x+1$

$S \equiv y:=y+1;$ if $x<0$ then $y:=-y$ fi

$lhs(S) \equiv \{y\}$

$rhs(S) \cup free(p) = \{x, y\}$

$aged(p, S) = \{y\}$

sp for $y:=y+1$:

$sp(y=x+1 \wedge y=y_0,\ y:=y+1)$

$\equiv y_0=x+1 \wedge y=y_0+1 \wedge x \ge 0$

sp for if :

$sp(y=x+1 \wedge y=y_0, y!=y+1; y:=-y)$

$\equiv y_0=x+1 \wedge y=y_0+1 \wedge x<0, y:=-y$

$\equiv y_0=x+1 \wedge y=-(y_0+1) \wedge x<0$

$sp(p, S) \equiv (y_0 = x+1 \wedge y = y_0 + 1 \wedge x \geq 0) \vee$
$(y_0 = x+1 \wedge y = -(y_0+1) \wedge x < 0)$

7) we can show the formal proof as following

1. $\{p \wedge B\}\ S_1\ \{q_1\}$ premise
2. $\{p \wedge \neg B\}\ S_2\ \{q_2\}$ premise
3. $\{(B \to p \wedge B) \wedge (\neg B \to p \wedge \neg B)\}$ if B then S_1 else S_2 fi $\{q_1 \vee q_2\}$ if-else 1,2
4. $(B \to p \wedge B) \wedge (\neg B \to p \wedge \neg B) \Leftrightarrow (B \wedge (p \wedge B)) \vee (\neg B \wedge (p \wedge \neg B))$ predicate logic
5. $(B \wedge (p \wedge B)) \vee (\neg B \wedge (p \wedge \neg B)) \Leftrightarrow p$ predicate logic
   # $LHS \Leftrightarrow p \wedge B \vee p \wedge \neg B \Leftrightarrow p \wedge (B \vee \neg B) \Leftrightarrow p$
6. $\{p\}$ if B then S_1 else S_2 fi $\{q_1 \vee q_2\}$ strengthen precondition 5,3

8) Given,

$S \equiv x := x * x;\ y := 2 * y$

$wlp(S, x = y) \equiv wlp(x := x * x;\ wlp(y := 2 * y,\ x = y))$

$\equiv wlp(x := x * x,\ x = 2 * y)$

$\equiv (x * x = 2 * y) \equiv p$

so, we need to show a formal proof for

$\vdash \{P\}\ S\ \{x = y\}$

1. $\{x = 2 * y\}\ y := 2 * y\ \{x = y\}$ Backward Assignment

2. $\{x * x = 2 * y\}\ x := x * x\ \{x = 2 * y\}$ Backward Assignment

3. $\{x * x = 2 * y\}\ x := x * x;\ y := 2 * y\ \{x = y\}$ sequence 2, 1

so, Hence Proved $\vdash \{P\}\ S\ \{x = y\}$.

9)

1. $\{x = 2^K \wedge K < n\}\ x := x * 2\ \{x_0 = 2^K \wedge K < n \wedge x = x_0 * 2\}$

2. $\{x_0 = 2^K \wedge K < n \wedge x = x_0 * 2\}\ K := K + 1\ \{x_0 = 2^{K_0} \wedge K_0 < n \wedge x = x_0 * 2 \wedge K = K_0 + 1\}$

3. $\{x = 2^K \wedge K < n\}\ x := x * 2;\ K := K + 1\ \{x_0 = 2^{K_0} \wedge K_0 < n \wedge x = x_0 * 2 \wedge K = K_0 + 1\}$

4. $x_0 = 2^{K_0} \wedge K_0 < n \wedge x = x_0 * 2 \wedge K = K_0 + 1 \rightarrow x = 2^K \wedge K \leq n.$

5. $\{x = 2^K \wedge K < n\}\ x := x * 2;\ K := K + 1\ \{x = 2^K \wedge K \leq n\}$

6. $\{inv\ x = 2^K \wedge K \leq n\}$ while $K < n$ do $x := x * 2;\ K := K + 1$ od $\{x = 2^K \wedge K \leq n \wedge K \geq n\}$

10)

1. R_1 - forward assignment
2. R_2 - forward assignment.
3. predicate logic
4. R_3 - weaken postcondition 2,3.
5. R_4 - sequence 1,4.
6. R_5 - backward assignment
7. R_6 - backward assignment
8. predicate logic
9. R_7 - strengthen precondition 8,7
10. R_8 - sequence 9,6.

11. R_9 - loop 10
12. R_{10} - sequence 5, 11
13. predicate logic
14. R_{11} - weaken postcondition 12, 13.