Bhavesh Rajesh Talreja (A20516822)
CS 536 HW4

1. Weakest Preconditions

Task 1.1

a) wlp(n:=sqrt(y)+1;n:=x∗n;skip,n=0)

Ans. Our task here is to find the weakest liberal precondition:

By definition,

wlp(S, Q) is called the weakest liberal precondition on the initial state such that S starting from the initial state either terminates in a final state satisfying the postcondition Q, diverges or returns a run-time error.

For a program S and postcondition Q we define wlp(S, Q) = P as
1. For any state $\sigma \models P$ either
   (a) M (S, σ) = σ’ and $\sigma' \models Q$, or
   (b) M (S, σ) =⊥, or
   (c) M (S, σ) = {}

2. For any state $\sigma \not\models P$, we have M (S, σ) = σ’ and $\sigma' \not\models Q$.

Using the algorithms for wlp:

wlp(n:=sqrt(y)+1;n:=x∗n;skip,n=0)
:=wlp(n:=sqrt(y) + 1; n:=x*n, wlp(skip, n=0)
:=wlp(n:=sqrt(y) + 1; n:=x*n, n=0)
:=wlp(n:=sqrt(y) + 1, wlp(n:=x*n, n=0))
:=wlp(n:=sqrt(y) + 1, [x*n/n](n=0))
:=wlp(n:=sqrt(y) + 1, x*n=0)
:=[sqrt(y) + 1/n](x*n=0)
:=x*(sqrt(y) + 1) = 0

b) wp(n:=sqrt(y) + 1; n:=x ∗ n; skip, n = 0)

Ans. Our task here is to find the weakest precondition:

By definition,

For a program S and postcondition Q we define wp(S, Q) = P as:
1. For any state $\sigma \models P$, we have M (S, σ) = σ’ and $\sigma' \models Q$.
2. For any state $\sigma \not\models P$, either
(a) M (S, σ) = σ’ such that $\sigma' \not\models Q$, or

(b) M (S, σ) =⊥, or (〈S, σ〉 results in an error)
(c) M (S, σ) = {}. (〈S, σ〉 diverges)

Using the algorithms for wp:

wp(S, Q) := D(S) ∧ wlp(S, Q)

wp(n:=sqrt(y)+1;n:=x∗n;skip,n=0)
:=D(n:=sqrt(y)+1;n:=x∗n;skip) ∧ wlp(n:=sqrt(y)+1;n:=x∗n;skip,n=0)

Solving first D(n:=sqrt(y)+1;n:=x∗n;skip)
:=D(n:=sqrt(y)+1) ∧ wlp(n:=sqrt(y)+1,D(n:=x*n;skip))
:=D(n:=sqrt(y)+1) ∧ wlp(n:=sqrt(y)+1,(D(n:=x*n) ∧ wlp(n:=x*n,D(skip))))
:=D(n:=sqrt(y)+1) ∧ wlp(n:=sqrt(y)+1,(D(n:=x*n) ∧ wlp(n:=x*n,T)))
:=D(n:=sqrt(y)+1) ∧ wlp(n:=sqrt(y)+1,(D(x) ∧ D(n) ∧ [x*n/x]T))
:=D(n:=sqrt(y)+1) ∧ wlp(n:=sqrt(y)+1,(T ∧ T ∧ T))
:=D(n:=sqrt(y)+1) ∧ wlp(n:=sqrt(y)+1, T)
:=D(n:=sqrt(y)+1) ∧ [sqrt(y)+1/n](T)
:=D(y) ∧ y≥ 0 ∧ D(1) ∧ T
:=T ∧ y≥ 0 ∧ T ∧ T
:=y≥ 0

Solving for wlp(n:=sqrt(y)+1;n:=x∗n;skip,n=0)
:=wlp(n:=sqrt(y) + 1; n:=x*n, wlp(skip, n=0)
:=wlp(n:=sqrt(y) + 1; n:=x*n, n=0)
:=wlp(n:=sqrt(y) + 1, wlp(n:=x*n, n=0))
:=wlp(n:=sqrt(y) + 1, [x*n/n](n=0))
:=wlp(n:=sqrt(y) + 1, x*n=0)
:=[sqrt(y) + 1/n](x*n=0)
:=x*(sqrt(y) + 1) = 0

wp(n:=sqrt(y)+1;n:=x∗n;skip,n=0)
:= (y≥ 0) ∧ (x*(sqrt(y) + 1) = 0)

c) wp(y:=−1; if y>0 then z:=1 else z:=x/y fi, z=1)

Ans. Our task here is to find the weakest precondition:

By definition,

For a program S and postcondition Q we define wp(S, Q) = P as:
1. For any state σ ⊨ P, we have M (S, σ) = σ’ and σ’ ⊨ Q.
2. For any state σ ⊭ P, either

Bhavesh Rajesh Talreja (A20516822)
CS 536 HW4

(a) M (S, σ) = σ' such that σ' ⊭ Q, or
(b) M (S, σ) =⊥, or (⟨S, σ⟩ results in an error)
(c) M (S, σ) = {}. (⟨S, σ⟩ diverges)

Using the algorithms for wp:

wp(S, Q) := D(S) ∧ wlp(S, Q)

wp(y:=−1; if y>0 then z:=1 else z:=x/y fi, z=1)
:=D(y:=−1; if y>0 then z:=1 else z:=x/y fi) ∧ wlp(y:=−1; if y>0 then z:=1 else z:=x/y fi, z=1)

Solving first D(y:=−1; if y>0 then z:=1 else z:=x/y fi)
:=D(y:=−1) ∧ wlp(y:=−1, D(if y>0 then z:=1 else z:=x/y fi))
:=D(y:=−1) ∧ wlp(y:=−1, (D(y>0) ∧ (y>0 → D(z=1)) ∧ (y≤0 → D(z=x/y))))
:=D(y:=−1) ∧ wlp(y:=−1, (D(y) ∧ D(0) ∧ (y>0 → D(1)) ∧ (y≤0 → D(x) ∧ D(y) ∧ y≠0))))
:=D(y:=−1) ∧ wlp(y:=−1, (T ∧ T ∧ (y>0 → T) ∧ (y≤0 → T ∧ T ∧ y≠0)))
:=D(y:=−1) ∧ wlp(y:=−1, ((y>0 → T) ∧ (y≤0 → y≠0)))
:=D(y:=−1) ∧ [−1/y]((y>0 → T) ∧ (y≤0 → y≠0))
:=D(y:=−1) ∧ ((-1>0 → T) ∧ (-1≤0 → -1≠0))
:=D(y:=−1) ∧ ((F → T) ∧ (T → T))
:=D(y:=−1) ∧ ((T) ∧ (T))
:=D(y:=−1) ∧ (T)
:=D(−1) ∧ (T)
:=T ∧ T
:=T

Solving for wlp(y:=−1; if y>0 then z:=1 else z:=x/y fi, z=1)
:=wlp(y:=−1; wlp(if y>0 then z:=1 else z:=x/y fi, z=1))
:=wlp(y:=−1; (y>0 → wlp(z:=1, z=1)) ∧ (y≤0 → wlp(z:=x/y, z=1)))
:=wlp(y:=−1; (y>0 → [1/z](z=1)) ∧ (y≤0 → [(x/y)/z](z=1)))
:=wlp(y:=−1; (y>0 → (1=1)) ∧ (y≤0 → (x/y=1)))
:=wlp(y:=−1; (y>0 → T) ∧ (y≤0 → x=y))
:=[-1/y](y>0 → T) ∧ (y≤0 → x=y)
:=(-1>0 → T) ∧ (-1≤0 → x=-1)
:=(F → T) ∧ (T → x=-1)
:=T ∧ (T → x=-1)
:=(T → x=-1)
:=¬(T) ∨ (x=-1)
:=F ∨ (x=-1)
:=(x=-1)

wp(y:=−1; if y>0 then z:=1 else z:=x/y fi, z=1)
:=T ∧ (x=-1)

Bhavesh Rajesh Talreja (A20516822)
CS 536 HW4

:=(x=-1)

Task 1.2

Consider a program S and a condition P. What is the relation between P and wlp(S, sp(S, P))? In particular, do we have wlp(S, sp(S, P)) ⇒ P? If yes, provide a proof, if not a counterexample.

Ans. Given a program S and a condition P, we want to find the relation between P and wlp(S, sp(S, P)).

In particular, we want to check if it is true that wlp(S, sp(S, P)) ⇒ P

The definition of sp(S, P) that we learnt in class:
We define the set SP − state, as all states that satisfy the strongest postcondition, i.e., SP − state(S, P) = {σ' ∈ states | σ' ⊨ sp(S, P)}. The strongest postcondition sp(S, P) holds in precisely those final states for which there exists an execution of S that starts from an initial state satisfying P. So, we can build an equivalent definition for the set SP − state:
SP − state(S, P) = {σ' ∈ states | σ' ⊨ sp(S, P)}
= {σ' ∈ states | ∃σ. σ ⊨ P ∧ M (S, σ) = σ'}

As per my understanding, from the above definition, it is pretty clear that strongest postcondition exists for a state where S executes when starting from an initial state satisfying P, i.e., sp(S, P) exists when S terminates. After the execution of S, we are in state σ' ⊨ sp(S, P) and ∃σ. σ ⊨ P ∧ M (S, σ) = σ'.

Now, using the definition of weakest liberal precondition:

wlp(S, Q) is called the weakest liberal precondition on the initial state such that S starting from the initial state either terminates in a final state satisfying the postcondition Q, diverges or returns a run-time error.

For a program S and postcondition Q we define wlp(S, Q) = P as
1. For any state σ ⊨ P either
   (a) M (S, σ) = σ' and σ' ⊨ Q, or
   (b) M (S, σ) =⊥, or
   (c) M (S, σ) = {}

2. For any state σ ⊭ P, we have M (S, σ) = σ' and σ' ⊭ Q.

Here, we are talking about the case where S terminates. So, we are talking about a particular case of wlp(S, Q),

Bhavesh Rajesh Talreja (A20516822)
CS 536 HW4

wlp(S, Q) = P as for any state σ ⊨ P, M (S, σ) = σ' and σ' ⊨ Q.

where our Q is sp(S, P)

wlp(S, sp(S, P)) = P as for any state σ ⊨ P, M (S, σ) = σ' and σ' ⊨ sp(S, P).

In order to prove, wlp(S, sp(S, P)) ⇒ P, we can consider an example as follows:

Proof by example:

Consider, S ≜ x:=0;y:=1 and any P, say P = {T}

We will find wlp(S, sp(S, P)) using the algorithms for wlp and sp:

wlp(x:=0;y:=1,sp(x:=0;y:=1,T))
:=wlp(x:=0;y:=1,sp(y:=1,sp(x:=0,T)))
:=wlp(x:=0;y:=1,sp(y:=1,($\exists x_0.\,[x_0/x]$(T) ∧ x = $[x_0/x]$(0))))
:=wlp(x:=0;y:=1,sp(y:=1,(T ∧ x = 0)))
:=wlp(x:=0;y:=1,sp(y:=1, x = 0))
:=wlp(x:=0;y:=1,($\exists y_0.\,[y_0/y]$(x=0) ∧ y = $[y_0/y]$(1)))
:=wlp(x:=0;y:=1,(x=0 ∧ y = 1))
:=wlp(x:=0,wlp(y:=1,x=0 ∧ y = 1))
:=wlp(x:=0, [1/y](x=0 ∧ y = 1))
:=wlp(x:=0, (x=0 ∧ 1 = 1))
:=wlp(x:=0, (x=0 ∧ T))
:=wlp(x:=0, x=0)
:=[0/x](x=0)
:=(0=0)
:=T

This is our P that we started with.

From all the above discussion and an example, we can say that wlp(S, sp(S, P)) ⇒ P

[NOTE TO TA: There might be other ways to prove this as well, but this is what I could come up with.]

2. Strongest Postconditions

   Task 2.1

   a) sp(x:=−1; if y>0 then x:=1 else z:=x/y fi, y≥0)

Bhavesh Rajesh Talreja (A20516822)
CS 536 HW4

Ans. Our task here is to find the strongest postcondition:

By definition,

sp(S, P) is called the strongest postcondition on the final state such that an execution of S exists starting from the initial satisfying the precondition P.

Using the algorithms for sp:

sp(x:=−1; if y>0 then x:=1 else z:=x/y fi, y≥0)
:=sp(if y>0 then x:=1 else z:=x/y fi, sp(x:=-1, y≥0))
:=sp(if y>0 then x:=1 else z:=x/y fi, ($\exists x_0$. $[x_0/x]$(y≥0) ∧ x=[x_0/x](-1)))
:=sp(if y>0 then x:=1 else z:=x/y fi, (y≥0 ∧ x=-1))
:=sp(x:=1, y≥0 ∧ x:=-1 ∧ y>0) ∨ sp(z:=x/y, y≥0 ∧ x:=-1 ∧ y≤0)
:=($\exists x_0$. $[x_0/x]$(y≥0 ∧ x:=-1 ∧ y>0) ∧ x=$[x_0/x]$(-1))
∨ ($\exists z_0$. $[z_0/z]$(y≥0 ∧ x:=-1 ∧ y≤0) ∧ z=$[z_0/z]$(x/y))
:=($\exists x_0$.(y≥0 ∧ x_0:=-1 ∧ y>0) ∧ x=-1) ∨ ($\exists z_0$.(y≥0 ∧ x:=-1 ∧ y≤0) ∧ z=(x/y))

This can be further simplified. But as the question mentions that there is no need to simplify the conditions, I am leaving it here.

b) sp(if y=0 then x:=x*5 else skip fi, x=10)

Ans. Our task here is to find the strongest postcondition:

By definition,

sp(S, P) is called the strongest postcondition on the final state such that an execution of S exists starting from the initial satisfying the precondition P.

Using the algorithms for sp:

sp(if y=0 then x:=x*5 else skip fi, x=10)
:=sp(x:=x*5, x=10 ∧ y=0) ∨ sp(skip, x=10 ∧ y≠0)
:=($\exists x_0$. $[x_0/x]$(x=10 ∧ y=0) ∧ x=$[x_0/x]$(x*5)) ∨ (x=10 ∧ y≠0)
:=($\exists x_0$. (x_0=10 ∧ y=0) ∧ x=(x_0*5)) ∨ (x=10 ∧ y≠0)

This can be further simplified. But as the question mentions that there is no need to simplify the conditions, I am leaving it here.

Task 2.1

Bhavesh Rajesh Talreja (A20516822)
CS 536 HW4

a) Consider a program S and a condition Q. What is the relation between the postcondition Q and sp(S, wlp(S, Q))? In particular, do we have sp(S, wlp(S, Q)) ⇒ Q? If yes, provide a proof, if not a counterexample.

Ans. Given a program S and a condition Q, we want to find the relation between Q and sp(S, wlp(S, Q)).

In particular, we want to check if it is true that sp(S, wlp(S, Q)) ⇒ Q

The definition of sp(S, P) that we learnt in class:
We define the set SP − state, as all states that satisfy the strongest postcondition, i.e., SP − state(S, P) = {σ' ∈ states | σ' ⊨ sp(S, P)}. The strongest postcondition sp(S, P) holds in precisely those final states for which there exists an execution of S that starts from an initial state satisfying P. So, we can build an equivalent definition for the set SP − state:
SP − state(S, P) = {σ' ∈ states | σ' ⊨ sp(S, P)}
= {σ' ∈ states | ∃σ. σ ⊨ P ∧ M (S, σ) = σ'}

As per my understanding, from the above definition, it is pretty clear that strongest postcondition exists for a state where S executes when starting from an initial state satisfying P, i.e., sp(S, P) exists when S terminates. After the execution of S, we are in state σ' ⊨ sp(S, P) and ∃σ. σ ⊨ P ∧ M (S, σ) = σ'.

Now, using the definition of weakest liberal precondition:

wlp(S, Q) is called the weakest liberal precondition on the initial state such that S starting from the initial state either terminates in a final state satisfying the postcondition Q, diverges or returns a run-time error.

For a program S and postcondition Q we define wlp(S, Q) = P as
1. For any state σ ⊨ P either
   (a) M (S, σ) = σ' and σ' ⊨ Q, or
   (b) M (S, σ) =⊥, or
   (c) M (S, σ) = {}

2. For any state σ ⊭ P, we have M (S, σ) = σ' and σ' ⊭ Q.

Here, we are talking about the case where S terminates. So, we are talking about a particular case of wlp(S, Q),

From above, sp(S, P) is
σ' ⊨ sp(S, P) and ∃σ. σ ⊨ P ∧ M (S, σ) = σ'

where our P is wlp(S, Q)

sp(S, wlp(S, Q)) = Q as for a state σ', where σ' ⊨ sp(S, P) and ∃σ. σ ⊨ P ∧ M (S, σ) = σ'

In order to prove, sp(S, wlp(S, Q)) ⇒ Q, we can consider an example as follows:

Proof by example:

Consider, S ≜ x:=0;y:=1, when we execute this statement, we can say the value of x will become 0 and y will become 1. So, logically, Q = {x=0 ∧ y=1}

We will find sp(S, wlp(S, Q)) using the algorithms for wlp and sp:

sp(x:=0;y:=1,wlp(x:=0;y:=1, (x=0 ∧ y = 1)))
:=sp(x:=0;y:=1, wlp(x:=0;y:=1,(x=0 ∧ y = 1)))
:=sp(x:=0;y:=1, wlp(x:=0,wlp(y:=1,x=0 ∧ y = 1)))
:=sp(x:=0;y:=1, wlp(x:=0, [1/y](x=0 ∧ y = 1)))
:=sp(x:=0;y:=1, wlp(x:=0, (x=0 ∧ 1 = 1)))
:=sp(x:=0;y:=1, wlp(x:=0, (x=0 ∧ T)))
:=sp(x:=0;y:=1, wlp(x:=0, x=0))
:=sp(x:=0;y:=1, [0/x](x=0))
:=sp(x:=0;y:=1, (0=0))
:=sp(x:=0;y:=1, T)
:=sp(y:=1,sp(x:=0, T))
:=sp(y:=1, ($\exists x_0$. $[x_0/x]$(T) ∧ x = $[x_0/x]$(0)))
:=sp(y:=1, (T ∧ x = 0))
:=sp(y:=1, x = 0)
:=($\exists y_0$. $[y_0/y]$(x=0) ∧ y = $[y_0/y]$(1))
:=(x=0 ∧ y = 1)

This is our Q that we logically thought at the start of this proof.

From all the above discussion and an example, we can say that sp(S, wlp(S, Q)) ⇒ Q

[NOTE TO TA: There might be other ways to prove this as well, but this is what I could come up with.]

3. One more wrap-up question

Task 3.1

How long (approximately) did you spend on this homework, in total hours of actual working time?

Ans. Totally I spent 6.5 hours on this assignment.