Proofs for TDA567 Lab 3

Farzad Besharati
Gustaf Järgren

1 Q1 Proof

We have the conditional statement:

```
wp(if B then S1 else S2) =
    (B => wp(S1, R)) && (!B => wp(S2, R))
```

Let S be

```
if (x < 0) then
    y := -x else y := x
```

Abbreviate:
$S1 : y := -x$
$S2 : y := x$

Let R be $0 \leq y \bigwedge 0 \leq x \rightarrow y = x \bigwedge 0 > x \rightarrow y = -x$

By conditional rule:
$x < 0 \rightarrow wp(y := -x, 0 \leq y \bigwedge 0 \leq x \rightarrow y = x \bigwedge 0 > x \rightarrow y = -x)$
$\bigwedge$
$\neg(x < 0) \rightarrow wp(y := x, 0 \leq y \bigwedge 0 \leq x \rightarrow y = x \bigwedge 0 > x \rightarrow y = -x)$

By assignment rule:
$x < 0 \rightarrow (0 \leq -x \bigwedge 0 \leq x \rightarrow -x = x \bigwedge 0 > x \rightarrow -x = -x)$
$\bigwedge$
$\neg(x < 0) \rightarrow (0 \leq x \bigwedge 0 \leq x \rightarrow x = x \bigwedge 0 > x \rightarrow x = -x)$
$=$
$true$

This program satisfies its postcondition in any initial state.

2 Q2 Proof

We have the conditional statement:

```
wp(if B then S1 else S2) =
    (B => wp(S1, R)) && (!B => wp(S2, R))
```

Let S be:

```
if (x > y) then
    big,small := x,y else big,small := y,x
```

Abbreviate:
$S1 : big, small := x, y$
$S2 : big, small := y, x$

Let R be $(big > small)$

By conditional rule:
$((x > y) \to wp(big := x; small := y, big > small))$
$\bigwedge$
$(\neg(x > y) \to wp(big := y; small := x, big > small))$

By assignment rule:
$((x > y) \to (x > y)) \bigwedge ((x <= y) \to (y > x))$

Simplify (by $p \to p == true$):
$true \bigwedge (x <= y \to y > x)$

Simplify (by $(true \bigwedge a = a)$):
$x <= y \to y > x$
$false \to x = y$

This program don't hold when $x = y$

3 Q3 Proof

We have the program:

```
wp(S1, wp(while B I D S2, R))
```

Let $S1$ be:

```
res := 0;
if (n0 >= 0) then (n,m := n0, m0) else (n,m := −n0, −m0)
```

Let B be:

```
0 < n
```

Let I be:

```
n0 >= 0 ==> (n0 − n) ∗ m0 == res
n0 < 0 ==> (n0 + n) ∗ m0 == res
```

Let D be:

```
decreases n
```

Let $S2$ be:

```
res := res + m
n := n − 1
```

Let R be:

```
n ∗ m = res
```

1. **Prove wp(S1, I), need to hold before loop:**

   By conditional rule:
   $wp(res := 0,$
   $((n0 >= 0) \rightarrow wp((n, m := n0, m0), I))$
   $\bigwedge$
   $(\neg(n0 >= 0) \rightarrow ((n, m := -n0, -m0), I)))$

   By assignment rule (n,m := n0,m0 ∧ n,m := -n0,-m0):
   $wp(res := 0,$
   $((n0 >= 0) \rightarrow$
   $(n0 >= 0 \rightarrow (n0 - n0) * m0 == res) \bigwedge n0 < 0 \rightarrow (n0 + n0) * m0 == res)$
   $\bigwedge$
   $(\neg(n0 >= 0) \rightarrow$
   $(n0 >= 0 \rightarrow (n0 + n0) * m0 == res) \bigwedge n0 < 0 \rightarrow (n0 - n0) * m0 == res))$

By assignment rule (res := 0):
$((n0 >= 0) \rightarrow$
$(n0 >= 0 \rightarrow (n0 - n0) * m0 == 0) \bigwedge n0 < 0 \rightarrow (n0 + n0) * m0 == 0)$
$\bigwedge$
$(\neg(n0 >= 0) \rightarrow$
$(n0 >= 0 \rightarrow (n0 + n0) * m0 == 0) \bigwedge n0 < 0 \rightarrow (n0 - n0) * m0 == 0)$

By simplify rule (a - a = 0 $\wedge$ 0 * a = 0):
$((n0 >= 0) \rightarrow$
$(n0 >= 0 \rightarrow 0 == 0) \bigwedge n0 < 0 \rightarrow 2 * n0 * m0 == 0)$
$\bigwedge$
((n0 < 0) $\rightarrow$
$(n0 >= 0 \rightarrow 2 * n0 * m0 == 0) \bigwedge n0 < 0 \rightarrow 0 == 0)$

True (inner implications true when their premises are false)

2. **Prove B $\wedge$ I $\rightarrow$ wp(S2, I), check invaraint for each interation:**

$B \wedge I \rightarrow$
$wp(res := res + m, wp(n := n - 1, (0 <= n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n < 0 \rightarrow (-n0 + n) * m == res)))$

By assignment:
$B \wedge I \rightarrow$
$wp(res := res + m, (0 <= n - 1 \wedge (n0 >= 0 \rightarrow (n0 - (n - 1)) * m == res) \wedge (n < 0 \rightarrow (-n0 - (n - 1)) * m == res))$

By assignment:
$B \wedge I \rightarrow$
$0 <= n - 1 \wedge (n0 >= 0 \rightarrow (n0 - (n - 1)) * m == res + m) \wedge (n < 0 \rightarrow (n0 + (n - 1)) * m0 == res + m)$

By unfolding (B $\wedge$ I):
$0 < n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n < 0 \rightarrow (-n0 - n) * m == res) \rightarrow$
$0 <= n - 1 \wedge (n0 >= 0 \rightarrow (n0 - (n - 1)) * m == res + m) \wedge (n < 0 \rightarrow (-n0 - (n - 1)) * m == res + m)$

Simplify:
$0 < n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n < 0 \rightarrow (-n0 - n) * m == res) \rightarrow$
$1 <= n \wedge (n0 >= 0 \rightarrow (n0 - n + 1) * m == res + m) \wedge (n < 0 \rightarrow (-n0 - n + 1) * m == res + m)$

Simplify:
$0 < n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n < 0 \rightarrow (-n0 - n) * m == res) \rightarrow$
$0 < n \wedge (n0 >= 0 \rightarrow (n0 - n) * m + m == res + m) \wedge (n < 0 \rightarrow (-n0 - n) * m + m == res + m)$

Simplify:
$0 < n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n < 0 \rightarrow (-n0 - n) * m == res) \rightarrow$
$0 < n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n < 0 \rightarrow (-n0 - n) * m == res)$

$I \wedge B \rightarrow wp(S, I)$ is True because this is proven above, $I \wedge B \leftrightarrow wp(S, I)$

3. **Prove $\neg$B $\wedge$ I $\rightarrow$ R, postcondition holds after loop:**

$\Longleftrightarrow$ { declaring B and I }
$\neg(0 < n) \wedge (0 <= n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n0 < 0 \rightarrow (-n0 - n) * m == res)) \rightarrow n0 * m0 == res$

$\Longleftrightarrow$ { arithmetic }
0 >= n $\wedge (0 <= n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n0 < 0 \rightarrow (-n0 - n) * m == res)) \rightarrow n0 * m0 == res$

$\Longleftrightarrow$ { 0 >= n and 0 <= 0 is equal to 0 == n }
0 == n $\wedge (0 <= n \wedge (n0 >= 0 \rightarrow (n0 - n) * m == res) \wedge (n0 < 0 \rightarrow (-n0 - n) * m == res)) \rightarrow n0 * m0 == res$

$\Longleftrightarrow$ { rewrite LHS based on above}
0 == n $\wedge (n0 >= 0 \rightarrow (n0 - 0) * m == res) \wedge (n0 < 0 \rightarrow (-n0 - 0) * m == res) \rightarrow n0 * m0 == res$

$\Longleftrightarrow$ { rewrite LHS based on above}
0 == n $\wedge (n0 >= 0 \rightarrow n0 * m == res) \wedge (n0 < 0 \rightarrow -n0 * m == res) \rightarrow n0 * m0 == res$

Based on $S1$ if n0 >= 0 then m = m0 because it's unchanged and the following is trivially True
$(n0 >= 0 \rightarrow n0 * m0 == res) \rightarrow n0 * m0 == res$

Based on $S1$ if n0 < 0 then m = -m0 then -a*-a = a*a and the following is trivially True
$(n0 < 0 \rightarrow -n0 * -m0 == res) \rightarrow n0 * m0 == res$

4. **Prove decreases expression is always** $>= 0$

   $I \wedge B \rightarrow n >= 0$

   $\Longleftrightarrow$ { declaring B }
   I $\wedge (0 < n) \rightarrow 0 >= n$

   $\Longleftrightarrow$ { arithmetic }
   I $\wedge (0 < n) \rightarrow 0 < n$

   True by trivial implication

5. **Prove I $\wedge$ B $\rightarrow$ wp(tmp := V; S, V $<$ tmp), decreasing n each iteration**

   B $\wedge$ I $\rightarrow wp(tmp := n; res := res + m; n := n - 1, n < tmp)$

   By sequential (x2):
   B $\wedge$ I $\rightarrow wp(tmp := n; wp(res := res + m, wp(n := n - 1, n < tmp)))$

   By assignment (x2):
   B $\wedge$ I $\rightarrow wp(tmp := n; n - 1 < tmp)$

   By assignment:
   B $\wedge$ I $\rightarrow n - 1 < n$

   By aritmetic:
   B $\wedge$ I $\rightarrow$ true

   It's always trivially True so it holds

4 Q4 Proof

We have the program:

```
n > 0 ==> wp(res :=1, i:=2, wp(while B I D S, R)) =
     I
     && (B && I ==> wp(S, I))
     && (!B && I ==> R)

     && (I ==> D >= 0)
     && (B && I ==>
          wp(tmp := D ; S, tmp > D))
```

Let B be:

```
i <= n
```

Let I be:

```
res == fact(i-1) && (i <= n+1)
```

Let D be:

```
n - i
```

Let S be:

```
res := res * i;
i := i + 1;
```

Let R be:

```
res == fact(n)
```

Finally let the function $fact(x)$ be a function that calculates the factorial for a number x.

The correctness of our method $ComputeFact$ can be expressed by the formula:

$$n > 0 ==> wp(res := 1, wp(i := 2, wp(whileBIDS, R)))$$

The first step is solving the loop, let's begin by proving that the invariant is preserved $B \wedge I ==> wp(S, I)$

$B \wedge I \rightarrow wp(res := res * i; i := i + 1, I)$

$\iff$ {sequential composition and expansion}
$B \wedge I \rightarrow wp(res := res * i, wp(i := i + 1, (res == fact(i - 1) \wedge (i <= n + 1))))$

$\iff$ {assignment}
$B \wedge I \rightarrow wp(res := res * i, (res == fact(i + 1 - 1) \wedge (i + 1 <= n + 1)))$

$\iff$ {assignment and simplification}
$B \wedge I \rightarrow (res * i == fact(i) \wedge (i + 1 <= n + 1)))$

$\iff$ {expansion}
$(i <= n) \wedge res == fact(i - 1) \wedge (i <= n + 1) \rightarrow (res * i == fact(i) \wedge (i + 1 <= n + 1)))$

$\iff$ {expand fact(i)}
$(i <= n) \wedge res == fact(i - 1) \wedge (i <= n + 1) \rightarrow$
$(res * i ==$ if$(n == 1)$ then 1 else $n * fact(n - 1) \wedge (i + 1 <= n + 1)))$

$\iff$ {use $res = fact(i - 1)$ from the LHS to simplify the RHS}
$(i <= n) \wedge res == fact(i - 1) \wedge (i <= n + 1) \rightarrow$
$(res * i ==$ if$(n == 1)$ then 1 else $n * res \wedge (i + 1 <= n + 1)))$

$\iff$ {$i := 2$ and from the LHS we get that $n >= i$, so we can simplify the if-case}
$(i <= n) \wedge res == fact(i - 1) \wedge (i <= n + 1) \rightarrow$
$(res * i == n * res \wedge (i + 1 <= n + 1)))$

unsure of how to continue, can we somehow prove that i == n?

We continue with proving that the failure of the loop guard and invariant implies the post-condition $!B \wedge I \rightarrow R$:

$!B \wedge I \rightarrow res = fact(n)$

$\iff$ {expand B and I}
$!(i \leq n) \wedge res == fact(i-1) \wedge (i \leq n+1) \rightarrow res = fact(n)$

$\iff$ {arithmetic}
$(i > n) \wedge res == fact(i-1) \wedge (i \leq n+1) \rightarrow res = fact(n)$

$\iff$ {$i > n$ and $i \leq n+1$ implies that $n = i-1$}
$(i > n) \wedge res == fact(i-1) \wedge (i \leq n+1) \rightarrow res = fact(i-1)$

$\iff$ {substitute $fact(i-1)$ on the RHS by using the LHS}
$(i > n) \wedge res == fact(i-1) \wedge (i \leq n+1) \rightarrow res = res$
$(i > n) \wedge res == fact(i-1) \wedge (i \leq n+1) \rightarrow true$
$true$

Next let's prove $I \to D \geq 0$

$I \to D \geq 0$

$\iff$ {expand}
$res == fact(i-1) \wedge (i \leq n+1) \to (n-1 \geq 0)$

$\iff$ {arithmetic}
$res == fact(i-1) \wedge (i \leq n+1) \to (n \geq 1)$

$\iff$ $\{i := 2$ and $i \leq n+1$ so it follows that $n \geq 1\}$
$res == fact(i-1) \wedge (i \leq n+1) \to true$
$true$