Shobhit - 19CS06008 - M.tech - 1st year

method M (x_0 : int) returns (x : int)

R: ensures ($x_0 < 3 ==> x == 1$) && ($x_0 >= 3 ==> x < x_0$);

{

S1: $x := x_0 - 3$;

if ($\underbrace{x < 0}_{B1}$) {

S2: $x := 1$;

}

else {

if ($\underbrace{true}_{B2}$) {

S3: $x := x + 1$;

}

else {

S4: $x := 10$;

}

}

}

wp(S1; if B1 then S2 else if B2 then S3 else S4 , R)

Seq. Rule

= wp (S1; wp (if B1 then S2 else if B2 then S3 else S4, R)) --- (1)

Compute wp(if B1 then S2 else if B2 then S3 else S4, R)

if Rule

= B1 ==> wp(S2, R) ∧ ¬B1 → wp(if B_2 then S3 else S4, R) --- (2)

Compute wp (if B2 then S3 else S4, R)

if Rule

= B2 → wp(S3, R) ∧ ¬B2 → wp(S4, R)

= true → wp($x := x + 1$, ($x_0 < 3 ==> x == 1$) && ($x_0 >= 3 ==> x < x_0$))

∧ false → wp($x := 10$, ($x_0 < 3 ==> x == 1$) && ($x_0 >= 3 ==> x < x_0$))

assignment rule (2x)

Scanned by CamScanner

$= true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&\; (x_0 >= 3 \implies x+1 < x_0)) \wedge$
$false \to ((x_0 < 3 \implies 10 == 1) \;\&\&\; (x_0 >= 3 \implies 10 < x_0))$

Simplify: $false \to ((x_0 < 3 \implies 10 == 1) \;\&\&\; (x_0 >= 3 \implies 10 < x_0))$

$= True$ [Rule : false $\to$ True/false = True]

$= true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&\; (x_0 >= 3 \implies x+1 < x_0)) \wedge True$

$= true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&\; (x_0 >= 3 \implies x+1 < x_0))$ ---- (3)

Replace B1, S2, & R in (2) &

Put (3) in (2)

$x < 0 \to WP(x := 1, (x_0 < 3 \implies x == 1) \;\&\&\; (x_0 >= 3 \implies x < x_0))$

$\wedge\; x >= 0 \to (true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&\; (x_0 >= 3 \implies x+1 < x_0)))$ ---- (4)

Assignment rule

$x < 0 \to (x_0 < 3 \implies 1 == 1) \;\&\&\; (x_0 >= 3 \implies 1 < x_0))$

$\wedge\; x >= 0 \to (true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&\; (x_0 >= 3 \implies x+1 < x_0))$ --- (5)

Simplify $\underbrace{(\underbrace{x_0 < 3}_{False/true} \implies \underbrace{1 == 1}_{True})}_{True} \;\&\&\; (x_0 >= 3 \implies 1 < x_0)$

True [if LHS is true RHS have to true it never be false &&
if LHS is false then RHS may be true or false from above two scenario result will be true]

Simplify:
$x_0 < 0 \to True \;\&\&\; True$
$= True$ [Rule True/false $\to$ True = True]
--- (6)

Put (6) in (5)

$True \wedge x >= 0 \to (true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&$
$(x_0 >= 3 \implies x+1 < x_0)))$

$x >= 0 \to (true \to ((x_0 < 3 \implies x+1 == 1) \;\&\&$
$(x_0 >= 3 \implies x+1 < x_0)))$ ---- (7)

Put (7) in (1)

P.T.O.

$wp(x := x_0 - 3,\ x >= 0 \rightarrow (true \rightarrow ((x_0 < 3 \implies x+1 == 1)\ \&\&$
$(x_0 >= 3 \implies x+1 < x_0))))$

Assignment rule :

$x_0 - 3 >= 0 \rightarrow (true \rightarrow ((x_0 < 3 \implies x_0 - 3 + 1 == 1)\ \&\&$
$(x_0 >= 3 \implies x_0 - 3 + 1 < x_0)))$

$x_0 >= 3 \rightarrow (true \rightarrow ((x_0 < 3 \implies x_0 == 3)\ \&\&\ \underbrace{(x_0 >= 3 \implies x_0 - 2 < x_0)}_{True}))$

$x_0 >= 3 \rightarrow (true \rightarrow (x_0 < 3 \implies x_0 == 3))$ ---- (8)

Simplify

Case1: $x_0 >= 3 = True$

$$\underbrace{true \rightarrow \underbrace{(true \rightarrow \underbrace{(false \implies x_0 == 3)}_{True})}_{True}}_{True}$$

Case2: $x_0 >= 3 = False$ i.e. $x_0 < 3$

$$\underbrace{false \rightarrow \underbrace{(true \rightarrow \underbrace{(true \implies \underbrace{x_0 == 3}_{false})}_{false})}_{false}}_{True}$$

By case 1 & case 2 formula (8) is tautology

∴ Program is correct

Q2:

method m1(n : nat) returns (i : nat)

Q requires n>=0

R ensures i == 2*n

{

S1: i := 0;

while ($\underbrace{i<n}_{B}$)

invariant $\underbrace{i<=n}_{I}$

variant $\underbrace{n-i}_{D}$

{ $\underbrace{i:=i+1;}_{S2}$ }

$\underbrace{i := 2*i;}_{S3}$

}

wp(i:=0; while(i<n) { i:=i+1 } ; i := 2*i, i == 2*n)

Sequential rule

wp(i:=0; while(i<n) { i:=i+1 } ; wp(i:=2*i, i == 2*n))

Compute wp(i:=2*i, i==2*n)

Assignment rule

2*i == 2*n

i == n

wp(i:=0; while(i<n) { i:=i+1 }, i == n)

sequential rule

wp(i:=0 ; wp(while B I D S2 , i==n)) --- 6

Compute wp(while B I D S2 , i==n)

while loop rule

1. i<=n &&
2. (i<n && i<=n) ==> wp(i:=i+1 , i<=n) &&
3. (i>=n && i<=n ==> i == n) &&
4. (i<=n ==> n-i >= 0) &&
5. (i<n && i<=n ==> wp(tmp := n-i ; i:=i+1, tmp>n-i))

compute (2)

$(i<n \ \&\& \ i<=n) ==> wP(i:=i+1, \ i<=n)$

Assignment rule

$(i<n \ \&\& \ i<=n) ==> i+1 <= n$

$(i<n) ==> i<n$

trivially true

compute (3)

$i>=n \ \&\& \ i<=n ==> i==n$

simplify

$i==n ==> i==n$

trivially true

compute (4)

$(i<=n ==> n-i>=0)$

simplify

$i<=n ==> i<=n$

trivially true

compute (5)

sequential rule

$(i<n \ \&\& \ i<=n ==> wP(tmp:=n-i; \ wP(i:=i+1, \ tmp>n-i))$

assignment rule

$(i<n \ \&\& \ i<=n ==> wP(tmp:=n-i; \ tmp> n-i-1))$

assignment rule

$i<n \ \&\& \ i<=n ==> \underbrace{n-i > n-i-1}_{true}$

simplify true/false -> true = true

$\therefore$ wP(while B I D S2, $i==n$) = $i<=n$ && true && true && true

= $i<=n$ --- (7)

put (7) in (6)

$wP(i:=0, \ i<=n)$

Assignment rule

$0<=n$

$Q -> wP(S, R)$

$n>=0 -> n>=0$

trivially true

Method M1 is correct

Q3:

function fib(n : nat) : nat

{ if n<=1 then n else fib(n-1) + fib(n-2) }

method fibfast (n: nat) returns (c: nat)

Q requires n >=1

R ensures c == fib(n)

{

S1: var p := 0;

S2: c := 1;

S3: var i := 1;

while $\underbrace{i < n}_{B}$

invariant $\underbrace{1 <= i <= n}_{I1}$

invariant $\underbrace{p == fib(i-1) \&\& c == fib(i)}_{I2}$

decreases$\underbrace{(n - i)}_{D}$

{S4: var new := p+c;

S5: p := c;

S6: c := new;

S7: i := i+1;

}

}

$I = \{I_1, I_2\}$, $S = \{S_4, S_6, S_7\}$

Q → wp(S1; S2; S3; while i<n {S4, S6, S7}, R) --- (6)

Sequential rule

Q → wp (S1; S2; S3; wp (while B I D S, R))

compute wp(while B I D S, R))

wp(while i<n 1<=i<=n && p==fib(i-1) && c==fib(i) n-i

new:=p+c && p:=c && c:=new && i:=i+1 , c==fib(n))

while loop rule

wp(while B I D S , R) =

1. I &&
2. (B && I ==> wp (S, I)) &&
3. (!B && I ==> R) &&
4. (I ==> D >= 0) &&
5. (B && I ==> wp (tmp := D; S, tmp > D))

compute 2

i< n && 1 <= i <= n && P == fib(i-1) && C == fib(i)

==> wp(new := P+C ; P:=C ; C:=new ; i:=i+1 ,

1<=i<=n && P == fib(i-1) && C == fib(i))

~~[illegible]~~

compute wP(new :=P+C; P := C ; C :=new ; i := i + 1 , 1 <= i <= n &&

P = = fib(i-1) && C = = fib(i))

Sequential rule

wP(new:=P+C ; P:=C ; C :=new ; wP(i: i+1 , 1<=i<=n

&& P == fib(i-1) && C == fib(i)))

Assignment rule

WP(new:=P+C ; P:=C ; C:=new , 1 <= i+1 <= n && P == fib(i) &&

C == fib(i+1))

Sequential rule

wP(new:=P+C ; P: =C ; wP(C:=new , 1< = i+1 <= n && P == fib(i)

&& C == fib(i+1))

Assignment rule then Sequential rule

wP (new: =P+C ; wP(P :=C , 1 <= i+1 <= n && P == fib(i) &&

new == fib(i+1))

Assignment rule then Sequential rule

wP(new:=P+C , 1 <= i+1 <= n && C == fib(i) && new == fib(i+1))

Assignment rule

1 <= i+1 <= n && C == fib(i) && P+C == fib(i+1)

compute ③

$[i >= n \ \&\& \ 1 <= i <= n \ \&\& \ p == fib(i-1) \ \&\& \ c == fib(i)]$

$\implies c == fib(n)$

Simplify

$(i == n \ \&\& \ p == fib(i-1) \ \&\& \ c == fib(i)) \implies c == fib(n)$

trivially true

compute ④

$(1 <= i <= n \ \&\& \ p == fib(i-1) \ \&\& \ c == fib(i)) \implies n-i >= 0$

simplify

$(1 <= i <= n \ \&\& \ p == fib(i-1) \ \&\& \ c == fib(i)) \implies i <= n$

true

[to make L.H.S true $1 <= i <= n$ have to be true
if this true then R.H.S. Have to be true]

compute ⑤

$i < n \ \&\& \ 1 <= i <= n \ \&\& \ p == fib(i-1) \ \&\& \ c == fib(i)$

$\implies wp($tmp := n-i ; new := p+c ; p := c ; c := new ; i := i+1, tmp > n-i)

Sequential rule ~~[illegible]~~

wp(tmp := n-i ; new := p+c ; p := c ; c := new ; wp(i := i+1, tmp > n-i))

Assignment rule then Sequential rule

wp(tmp := n-i ; new := p+c ; p := c ; wp(c := new, tmp > n-i-1))

Assignment rule then Sequential rule

wp(tmp := n-i ; new := p+c ; wp(p := c, tmp > n-i-1))

Assignment rule then Sequential rule

wp(tmp := n-i ; wp(new := p+c , tmp > n-i-1))

Assignment rule then Sequential rule

$wp(tmp := n-i,\ tmp > n-i-1)$

Assignment rule

$n-i > n-i-1$

true

$i < n \ \&\& \ 1 <= i <= n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i)$
$==> true$

simplify $\quad$ true/false $==> true = true$

$\boxed{true}$

we can further simplify ②

$i < n \ \&\& \ 1 <= i <= n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i)$
$==> \ 1 <= i+1 <= n \ \&\& \ C == fib(i) \ \&\& \ P+C == fib(i+1)$

simplify

$1 <= i < n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i)$
$==> 1 <= i+1 <= n \ \&\& \ C == fib(i) \ \&\& \ P+C == fib(i+1)$

By using def. $fib(i+1) = fib(i) + fib(i-1)$
the above formula is true

$\boxed{true}$

Now,

$wp(\text{while } B \ I \ D \ S, R) = 1 <= i <= n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i)$
$\&\& \ true \ \&\& \ true \ \&\& \ true \ \&\& \ true$
$= 1 <= i <= n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i)$

Put above result in ⑥

$Q \to wp(P := 0;\ C := 1;\ i := 1,\ 1 <= i <= n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i))$

Sequential rule

$Q \to wp(P := 0;\ C := 1;\ wp(i := 1,\ 1 <= i <= n \ \&\& \ P == fib(i-1) \ \&\& \ C == fib(i)))$

Assignment rule & then sequential rule

~~$1 <= 1 <= n \ \&\& \ P == fib(0) \ \&\& \ C == fib(1)$~~

~~$Q \to wp(P := 0; wp(C := 1, 1 <= 1 <= n$~~

$Q \to wp(P := 0;\ wp(C := 1,\ 1 <= 1 <= n \ \&\& \ P == fib(0) \ \&\& \ C == fib(1)))$

Assignment rule & then Sequential rule

$Q \rightarrow wp(\ p := 0,\ 1 <= 1 <= n\ \&\&\ p == fib(0)\ \&\&\ 1 == fib(1))$

Assignment rule

$Q \rightarrow 1 <= 1 <= n\ \&\&\ 0 == fib(0)\ \&\&\ 1 == fib(1)$

$n >= 1 \rightarrow n >= 1\ \&\&\ 0 == fib(0)\ \&\&\ 1 == fib(1)$

Simplify

By the def. fib function $fib(0) = 0$ & $fib(1) = 1$

$n >= 1 \rightarrow n >= 1$ && true && true

$n >= 1 \rightarrow n >= 1$

trivially true

fib method is true