Assignment 2 - Emirps

Rahil Agrawal z5165505
Aditya Karia zXXXXXXX

COMP2111 18s1

1 Task 1 - Specification Statement

Notes,x:
-Write neatly
-make sure grammar is correct
-look at examples for default spec structure.

Define an Emirp using 2 functions - reverse and prime. Make these functions match with their given specs in order to help prove implications.

Pre condition,x: n is a positive number - $n > 0$

Post condition $EMIRP(r, n)$ where r is the n^{th} emirp(where emirp is as defined above). Therefore our program can be specified by,x:
proc $\textsc{Emirp}(\textbf{value } n, \textbf{result } r) \cdot$
$\llcorner n, r, x : \left[\ n > 0, Emirp(r, n)\ \right]\lrcorner_{(1)}$

2 Task 2 - Derivation

proc EMIRP(**value** n, **result** r) $\cdot \llcorner n, r, x : \left[\ n > 0, Emirp(r, n)\ \right]\lrcorner_{(1)}$

(1) $\sqsubseteq$ ⟨**c-frame**⟩

$\llcorner r, x : \left[\ n > 0, Emirp(r, n)\ \right]\lrcorner_{(2)}$

(2) $\sqsubseteq$ ⟨**i-loc**⟩

$\llcorner i, r, x : \left[\ n > 0, Emirp(r, n)\ \right]\lrcorner_{(3)}$

(3) $\sqsubseteq$ ⟨**seq**⟩

$\llcorner i, x, r : \left[\ n > 0, i = 1 \wedge x = 13 \wedge n > 0\ \right]\lrcorner_{(4)};$

$\llcorner i, x : \left[\ i = 1 \wedge x = 13 \wedge n > 0, Emirp(r, n)\ \right]\lrcorner_{(5)}$

(4) $\sqsubseteq$ ⟨**c-frame**⟩

$i, x : \left[\ n > 0, i = 1 \wedge x = 13 \wedge n > 0\ \right]$

$\sqsubseteq$ ⟨**ass - (1)**⟩

i := 1

x := 13

(5) $\sqsubseteq$ ⟨**seq**⟩

$\llcorner i, r, x : \left[\ i = 1 \wedge x = 13 \wedge n > 0, Inv\ \right]\lrcorner_{(6)};$

$\llcorner i, r, x : \left[\ Inv, Inv \wedge i = n\ \right]\lrcorner_{(7)};$

$\llcorner i, r, x : \left[\ Inv \wedge i = n, Emirp(r, n)\ \right]\lrcorner_{(8)}$

(6) $\sqsubseteq$ ⟨**w-pre, c-frame - (2)**⟩

$r, x : \left[\ Inv[^{13}/_{r}], Inv\ \right]$

$\sqsubseteq$ ⟨**ass - (3)**⟩

r := 13

(7) $\sqsubseteq$ ⟨**while**⟩

while i $\neq$ n do

$\llcorner i, r, x : \left[\ Inv \wedge i \neq n, Inv\ \right]\lrcorner_{(9)}$

od;

(8) $\sqsubseteq$ ⟨**w-pre, c-frame - (4)** ⟩

$\llcorner r, x : \left[\ \text{EMIRP}(r, n), \text{EMIRP}(r, n)\ \right]\lrcorner_{(10)}$

$\sqsubseteq$ ⟨**skip - (5)**⟩

skip

(9) $\sqsubseteq$ ⟨**seq**⟩

$\llcorner r, x : \left[\ Inv \wedge i \neq n, Inv[^{x+1}/_{x}]\ \right]\lrcorner_{(10)};$

$\llcorner r, x : \left[\ Inv[^{x+1}/_{x}], Inv\ \right]\lrcorner_{(11)}$

$(10) \sqsubseteq \quad \langle \textbf{ass - (6)} \rangle$

$\mathbf{x := x + 1}$

$(11) \sqsubseteq \quad \langle \textbf{i-loc, seq} \rangle$

$\llcorner a, i, r, x : [\ Inv[^{x+1}/_x], Inv[^{x+1}/_x] \wedge a = 1\ ]\lrcorner_{(12)};$

$\llcorner a, i, r, x : [\ Inv[^{x+1}/_x] \wedge a = 1, Inv\ ]\lrcorner_{(13)}$

$(12) \sqsubseteq \quad \langle \textbf{c-frame} \rangle$

$a, x : [\ Inv[^{x+1}/_x], Inv[^{x+1}/_x] \wedge a = 1\ ]$

$\sqsubseteq \quad \langle \textbf{ass - (7)} \rangle$

$\mathbf{a := 1}$

$(13) \sqsubseteq \quad \langle \textbf{seq} \rangle$

$$\llcorner a, i, r, x : \left[\begin{array}{l} Inv[^{x+1}/_x] \wedge a = 1, (a = 1\ \wedge \neg \exists k \in 2..(x-1)\,(x \textbf{ mod } k = 0)) \\ \vee (a = 0 \wedge \exists k \in 2..(x-1)\,(x \textbf{ mod } k = 0))) \end{array} \right]\lrcorner_{(14)};$$

$$\llcorner a, i, r, x : \left[\begin{array}{l} (a = 1\ \wedge \neg \exists k \in 2..(x-1)\,(x \textbf{ mod } k = 0)) \\ \vee (a = 0 \wedge \exists k \in 2..(x-1)\,(x \textbf{ mod } k = 0))), Inv \end{array} \right]\lrcorner_{(15)}$$

$(14) \sqsubseteq \quad \langle \textbf{w-pre - (8)} \rangle$

$a, x : [\ a > 1 \wedge x > 0, post(14)\ ]$

$\sqsubseteq \quad \langle \textbf{proc} \rangle$

$\mathbf{isPrime(x, a)}$

$(15) \sqsubseteq \quad \langle \textbf{if} \rangle$

$\mathbf{if\ a = 1}$

$\mathbf{then}\ \llcorner a, i, r, x : [a = 1 \wedge pre(15), post(15)]\lrcorner_{(16)}$

$\mathbf{else}\ \llcorner p, x : [a \neq 1 \wedge pre(15), post(15)]\lrcorner_{(17)}$

$\mathbf{fi}$

$(16) \sqsubseteq \quad \langle \textbf{i-loc} \rangle$

$a, i, r, s, x : [\ pre(16), post(16)\ ]$

$\sqsubseteq \quad \langle \textbf{seq} \rangle$

$\llcorner s, x : [\ pre(16), s = 0\ ]\lrcorner_{(18)};$

$\llcorner a, i, r, s, x : [\ s = 0, post(16)\ ]\lrcorner_{(19)}$

$(17) \sqsubseteq \quad \langle \textbf{c-frame, w-pre- (9)} \rangle$

$i, r : [\ Inv, Inv\ ]$

$\sqsubseteq \quad \langle \textbf{skip - (10)} \rangle$

$\mathbf{skip}$

$(18) \sqsubseteq \quad \langle \textbf{ass - (11)} \rangle$

$\mathbf{s := 0}$

$(19) \sqsubseteq \quad \langle \textbf{seq} \rangle$

$\llcorner s, x : [\ pre(19),$ reversen function post condition $]\lrcorner_{(20)};$

$\llcorner a, i, r, s, x : [$ reversen function post condition, $post(19)\ ]\lrcorner_{(21)}$

$(20) \sqsubseteq \quad \langle \textbf{w-pre - (12)} \rangle$

$s, x : [\text{ reverse function pre condition}, \text{reversen function post condition }]$

$\sqsubseteq \quad \langle \textbf{proc} \rangle$

$\mathbf{reversen(x, s)}$

$(21) \sqsubseteq \quad \langle \textbf{i-loc, seq} \rangle$

$\llcorner a, i, r, s, b, x : [\ pre(21), pre(21) \wedge b = 1\ ]_{\lrcorner(22)};$

$\llcorner a, i, r, s, b, x : [\ pre(21) \wedge b = 1, post(21)\ ]_{\lrcorner(23)}$

$(22) \sqsubseteq \quad \langle \textbf{c-frame, ass - (13)} \rangle$

$\mathbf{b := 1}$

$(23) \sqsubseteq \quad \langle \textbf{seq} \rangle$

$$\llcorner a, i, r, s, b, x : \left[\begin{array}{l} pre(21) \wedge b = 1, (b = 1\ \wedge \neg \exists k \in 2..(s-1)\,(s \textbf{ mod } k = 0)) \\ \vee (b = 0 \wedge \exists k \in 2..(s-1)\,(s \textbf{ mod } k = 0)) \end{array} \right]_{\lrcorner(24)};$$

$$\llcorner a, i, r, s, b, x : \left[\begin{array}{l} (b = 1\ \wedge \neg \exists k \in 2..(s-1)\,(s \textbf{ mod } k = 0)) \\ \vee (b = 0 \wedge \exists k \in 2..(s-1)\,(s \textbf{ mod } k = 0)), post(21) \end{array} \right]_{\lrcorner(25)}$$

$(24) \sqsubseteq \quad \langle \textbf{w-pre - (14)} \rangle$

$$a, i, r, s, b, x : \left[\begin{array}{l} s > 0 \wedge b = 1, (b = 1\ \wedge \neg \exists k \in 2..(s-1)\,(s \textbf{ mod } k = 0)) \\ \vee (b = 0 \wedge \exists k \in 2..(s-1)\,(s \textbf{ mod } k = 0)) \end{array} \right]$$

$\sqsubseteq \quad \langle \textbf{proc} \rangle$

$\mathbf{isPrime(s, b)}$

$(25) \sqsubseteq \quad \langle \textbf{if} \rangle$

$\mathbf{if\ b = 1 \wedge s \neq x}$

$\mathbf{then}\ \llcorner i, x : [b = 1 \wedge s \neq x \wedge pre(25), post(25)]_{\lrcorner(26)}$

$\mathbf{else}\ \llcorner i, r, a, s, b, x : [(b \neq 1 \vee s = x) \wedge pre(25), post(25)]_{\lrcorner(27)}$

$\mathbf{fi};$

$(26) \sqsubseteq \quad \langle \textbf{c-frame, w-pre- (15)} \rangle$

$a, i, r, s, b, x : [\ Inv[^{i+1}/_i][^x/_r], Inv\ ]$

$\sqsubseteq \quad \langle \textbf{ass - (16)} \rangle$

$\mathbf{i := i + 1}$

$\mathbf{r := x}$

$(27) \sqsubseteq \quad \langle \textbf{c-frame, w-pre- (17)} \rangle$

$a, i, r, s, b, x : [\ Inv, Inv\ ]$

$\sqsubseteq \quad \langle \textbf{skip - (18)} \rangle$

$\mathbf{skip}$

proc $\textsc{isPrime}$(**value** r, **result** a) $\cdot$

$\llcorner r, a : \left[\begin{array}{l} a = 1 \wedge r > 0, (a = 1\ \wedge \neg\exists k \in 2..(x-1)\,(x \ \mathbf{mod}\ k = 0)) \\ \vee(a = 0 \wedge \exists k \in 2..(x-1)\,(x\ \mathbf{mod}\ k = 0) \end{array}\right]\lrcorner_{(1)}$

(1) $\sqsubseteq$ $\langle$**seq, i-loc**$\rangle$

$\llcorner r, a, j : [\ a = 1 \wedge r > 0, a = 1 \wedge r > 0 \wedge j = 2\ ]\lrcorner_{(2)};$

$\llcorner r, a, j : \left[\begin{array}{l} a = 1 \wedge r > 0 \wedge j = 2, (a = 1\ \wedge \neg\exists k \in 2..(x-1)\,(x\ \mathbf{mod}\ k = 0)) \\ \vee(a = 0 \wedge \exists k \in 2..(x-1)\,(x\ \mathbf{mod}\ k = 0) \end{array}\right]\lrcorner_{(3)}$

(2) $\sqsubseteq$ $\langle$**ass - (19)**$\rangle$

j := 2

(3) $\sqsubseteq$ $\langle$**seq**$\rangle$

$\llcorner r, a, j : [\ a = 1 \wedge r > 0 \wedge j = 2, Inv_2\ ]\lrcorner_{(4)};$

$\llcorner r, a, j : [\ Inv_2, Inv_2 \wedge j = r\ ]\lrcorner_{(5)};$

$\llcorner r, a, j : [\ Inv_2 \wedge j = r, post(3)\ ]\lrcorner_{(6)}$

(4) $\sqsubseteq$ $\langle$**w-pre - (20)**$\rangle$

$r, a, j : [\ Inv_2, Inv_2\ ]$

$\sqsubseteq$ $\langle$**skip - (21)**$\rangle$

skip

(6) $\sqsubseteq$ $\langle$**w-pre - (22)**$\rangle$

$\llcorner r, a, j : [\ post(3), post(3)\ ]\lrcorner_{(7)}$

$\sqsubseteq$ $\langle$**skip - (23)**$\rangle$

skip

(5) $\sqsubseteq$ $\langle$**while**$\rangle$

while j $\neq$ r do

$\llcorner r, j : [\ Inv_2 \wedge j \neq r, Inv_2\ ]\lrcorner_{(8)}$

od;

(8) $\sqsubseteq$ $\langle$**seq**$\rangle$

$\llcorner r, j : [\ pre(8), Inv_2[^{j+1}/_j]\ ]\lrcorner_{(9)};$

$\llcorner r, j : [\ Inv_2[^{j+1}/_j], Inv_2\ ]\lrcorner_{(10)}$

(9) $\sqsubseteq$ $\langle$**if**$\rangle$

if r mod j = 0

then $\llcorner a : [r\ \mathbf{mod}\ j = 0 \wedge pre(9), post(9)]\lrcorner_{(11)}$

else $\llcorner a : [r\ \mathbf{mod}\ j \neq 0 \wedge pre(9), post(9)]\lrcorner_{(12)}$

fi;

(10) $\sqsubseteq$ $\langle$**ass - (24)**$\rangle$

j := j + 1

(11) $\sqsubseteq$ $\langle$**w-pre - (25)**$\rangle$

$r, a, j : [\ Inv_2[^0/_a][^{j+1}/_j], post(11)\ ]$

$\sqsubseteq$ $\langle$**ass - (26)**$\rangle$

a := 0

$(12) \sqsubseteq \quad \langle$**w-pre - (27)**$\rangle$

$r, a, j : [\ Inv_2, post(11)\ ]$

$\sqsubseteq \quad \langle$**skip - (28)**$\rangle$

skip

We gather the code for the procedure body of EMIRP:

$\mathbf{EMIRP}(\mathbf{r}, \mathbf{n}) :$

```
    var i := 1;
    var x := 13;
    r := 13;
    while j ≠ r do
        x := x + 1;
        var a := 1;
        isPrime(x, a);
        if a = 1 then
            var s := 0;
            reversen(x, s);
            var b := 1;
            isPrime(s, b);
            if b = 1 ∧ s ≠ x then
                i := i + 1;
                r := x;
    od;
```

Also, we gather the code for the procedure body of ISPRIME:

$\mathbf{isPrime}(\mathbf{r}, \mathbf{j}) :$

```
    var j := 2;
    while j ≠ r do
        if (r mod j) = 0 then
            a := 0;
        j := j + 1;
    od;
```

We have derived our code. However we need to prove **some** refinements.

2.1 Implication 1: $(4) \sqsubseteq i := 1$

To prove: $i = i_0 \wedge n > 0 \Rightarrow (i = 1 \wedge x = 13 \wedge n > 0)[^1/_i][^{13}/_x]$

Proof:

$LHS = i = i_0 \wedge n > 0$

$\Rightarrow$ ⟨1=1 ∧ 13=13 is vacuously true⟩

$1 = 1 \wedge 13 = 13 \wedge i = i_0 \wedge n > 0$

$\Rightarrow \langle A \wedge B \wedge C \wedge D \Rightarrow A \wedge B \wedge C \rangle$

$1 = 1 \wedge 13 = 13 \wedge n > 0$

$\Rightarrow \langle 1 = 1 \Rightarrow (i = 1)[^1/_i], 13 = 13 \Rightarrow (x = 13)[^{13}/_x] \rangle$

$(i = 1 \wedge x = 13 \wedge n > 0)[^1/_i][^{13}/_x]$

$\Rightarrow$ ⟨clearly⟩

RHS

2.2 Implication 2: $(6) \sqsubseteq r, x : [Inv[^{13}/_x], Inv]$

To prove w-pre we need to prove: $pre \Rightarrow pre'$

To prove: $i = 1 \wedge n > 0 \wedge x = 13 \Rightarrow Inv[^{13}/_r]$

Proof:

$LHS = i = 1 \wedge n > 0 \wedge x = 13$

$\Rightarrow \langle A \wedge B \wedge C \Rightarrow A \wedge B \rangle$

$i = 1 \wedge x = 13$

$\Rightarrow$ ⟨We know that 13 is the 1st emirp from our definition of emirp, also 13≥r in this case⟩

$i = 1 \wedge Emirp(13, 1) \wedge x = 13 \wedge 13 \geq r$

$\Rightarrow$ ⟨This is our Inv with 13 substituted for x⟩

$Inv[^{13}/_x]$

$\Rightarrow$ ⟨clearly⟩

RHS

2.3 Implication 3: $r, x : [Inv[^{13}/_{r}], Inv] \sqsubseteq r := 13$

To prove: $r = r_0 \wedge Inv[^{13}/_{r}] \Rightarrow Inv[^{13}/_{r}]$

Proof:

$LHS = r = r_0 \wedge Inv[^{13}/_{r}]$

$\Rightarrow \langle A \wedge B \Rightarrow A \rangle$

$Inv[^{13}/_{r}]$

$\Rightarrow$ ⟨clearly⟩

RHS

2.4 Implication 4: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.5 Implication 5: (10) $\sqsubseteq$ *skip*

To prove skip, we need to prove $pre \Rightarrow post[^{r_0}/_r]$

To prove: $Emirp(r, n) \Rightarrow Emirp(r, n)[^{r_0}/_r]$

Proof:

$LHS = Emirp(r, n)$

$\Rightarrow$ ⟨Since r_0 is the value of r in the precondition, $r = r_0$ in the precondition⟩

$Emirp(r_0, n)$

$\Rightarrow$ ⟨clearly⟩

RHS

2.6 Implication 6: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.7 Implication 7: $Inv[^{x+1}/_x], Inv[^{x+1}/_x] \wedge a = 1 \sqsubseteq a := 1$

To prove: $a = a_0 \wedge Inv[^{x+1}/_x] \Rightarrow (a = 1 \wedge Inv[^{x+1}/_x])[^1/_a]$

Proof:

$LHS = a = a_0 \wedge Inv[^{x+1}/_x]$

$\Rightarrow$ $\langle$1=1 is vacuously true$\rangle$

$1 = 1 \wedge a = a_0 \wedge Inv[^{x+1}/_x]$

$\Rightarrow$ $\langle A \wedge B \wedge C \Rightarrow A \wedge B \rangle$

$1 = 1 \wedge Inv[^{x+1}/_x]$

$\Rightarrow$ $\langle 1 = 1 \Rightarrow (a = 1 \wedge Inv[^x/_{x+1}])[^1/_i]$ (Since, Inv does not involve a)$\rangle$

$(a = 1 \wedge Inv[^{x+1}/_x])[^1/_a]$

$\Rightarrow$ $\langle$clearly$\rangle$

RHS

2.8 Implication 8: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

RHS

2.9 Implication 9: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.10 Implication 10: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.11 Implication 11: (18) $\sqsubseteq s := 0$

To prove: $s = s_0 \land (a = 1 \ \land \neg \exists k \in 2..(x-1)\,(x \ \mathbf{mod} \ k = 0)) \lor$
$(a = 0 \land \exists k \in 2..(x-1)\,(x \ \mathbf{mod} \ k = 0))) \Rightarrow s = 0[^0/_s]$

Proof:
$LHS = s = s_0 \land (a = 1 \ \land \neg \exists k \in 2..(x-1)\,(x \ \mathbf{mod} \ k = 0)) \lor$
$(a = 0 \land \exists k \in 2..(x-1)\,(x \ \mathbf{mod} \ k = 0)))$
$\Rightarrow$ ⟨0=0 is vacuously true⟩
$0 = 0 \land s = s_0 \land (a = 1 \ \land \neg \exists k \in 2..(x-1)\,(x \ \mathbf{mod} \ k = 0)) \lor$
$(a = 0 \land \exists k \in 2..(x-1)\,(x \ \mathbf{mod} \ k = 0)))$
$\Rightarrow \langle A \land B \land C \Rightarrow A \rangle$
$0 = 0$
$\Rightarrow \langle 0 = 0 \Rightarrow (s = 0)[^1/_s] \rangle$
$(s = 0)[^0/_s]$
$\Rightarrow$ ⟨clearly⟩
RHS

2.12 Implication 12: $\sqsubseteq$

To prove:$BLAH$

Proof:
$LHS = BLAH$
$\Rightarrow$ ⟨BLAH⟩
$BLAH$
$\Rightarrow$ ⟨BLAH⟩
$BLAH$
$\Rightarrow$ ⟨BLAH⟩
$BLAH$
$\Rightarrow$ ⟨BLAH⟩
RHS

2.13 Implication 13: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

RHS

2.14 Implication 14: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

RHS

2.15 Implication 15: $\sqsubseteq$

To prove: $BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.16 Implication 16: $\sqsubseteq$

To prove: $BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.17 Implication 17: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.18 Implication 18: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.19 Implication 19: $\sqsubseteq$

To prove: $BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

RHS

2.20 Implication 20: $\sqsubseteq$

To prove: $BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

$BLAH$

$\Rightarrow$ $\langle$BLAH$\rangle$

RHS

2.21 Implication 21: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

RHS

2.22 Implication 22: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

RHS

2.23 Implication 23: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.24 Implication 24: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.25 Implication 25: $\sqsubseteq$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.26 Implication 26: $[Inv_2[^0/_a][^{j+1}/_{j]}], post(11) \sqsubseteq a := 0$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

$BLAH$

$\Rightarrow$ ⟨BLAH⟩

RHS

2.27 Implication 27: $(12) \sqsubseteq [Inv_2, post(11)]$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

RHS

2.28 Implication 28: $[Inv_2, post(11)] \sqsubseteq skip$

To prove:$BLAH$

Proof:

$LHS = BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

$BLAH$

$\Rightarrow \langle \text{BLAH} \rangle$

RHS

3 Task 3 - C Code

```
#include <stdio.h>
#include ”reverse.h”

```

```
unsigned long emirp(unsigned long n);
void isPrime(unsigned long r, int *a);

int main (int argc, char* argv[]){
        unsigned long n;
        if(scanf("%lu", &n)==1)
           printf("%lu\n",emirp(n));
}

unsigned long emirp(unsigned long n) {
    int i = 1;
    unsigned long x = 13;
    unsigned long r = 13;
    while (i != n) {
        x = x + 1;
        int a = 1;
        isPrime(x, &a);
        if (a == 1) {
            unsigned long s = 0;
            reversen(x, &s);
            int b = 1;
            isPrime(s, &b);
            if (b == 1 && s != x)
                i = i + 1;
                r = x;
        }
    }
        return r;
}

void isPrime(unsigned long r, int *a) {
    unsigned long j = 2;
    while (j != r) {
        if (r % j == 0)
            *a = 0;
        j = j + 1;
    }
}
```

- Write something about how the C code relates.
- Compare with examples