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

2 Task 2 - Derivation

$$\textbf{proc } \textsc{emirp}(\textbf{value } n, \textbf{result } r) \cdot \llcorner n, r, x : [\ n > 0, Emirp(r, n)\ ] \lrcorner_{(1)}$$

$(1) \sqsubseteq$ $\langle$**c-frame**$\rangle$

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

$(2) \sqsubseteq$ $\langle$**i-loc**$\rangle$

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

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

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

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

$(4) \sqsubseteq$ $\langle$**c-frame**$\rangle$

$i, x : [\ n > 0, i = 1 \land x = 13 \land n > 0\ ]$

$\sqsubseteq$ $\langle$**ass - (1)**$\rangle$

i $:=$ **1**

x $:=$ **13**

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

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

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

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

$(6) \sqsubseteq$ $\langle$**w-pre, c-frame - (2)**$\rangle$

$r, x : [\ Inv[^{13}/_x], Inv\ ]$

$\sqsubseteq$ $\langle$**ass - (3)**$\rangle$

r $:=$ **13**

$(7) \sqsubseteq$ $\langle$**while**$\rangle$

while i $\neq$ **n do**

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

od;

$(8) \sqsubseteq$ $\langle$**w-pre, c-frame - (4)** $\rangle$

$\llcorner r, x : [\ \textsc{emirp}(r, n), \textsc{emirp}(r, n)\ ] \lrcorner_{(10)}$

$\sqsubseteq$ $\langle$**skip - (5)**$\rangle$

skip

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

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

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

(10) $\sqsubseteq$ $\langle$**ass - (6)**$\rangle$

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

(11) $\sqsubseteq$ $\langle$**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) $\sqsubseteq$ $\langle$**c-frame**$\rangle$

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

$\sqsubseteq$ $\langle$**ass - (7)**$\rangle$

$\mathbf{a := 1}$

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

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

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

(14) $\sqsubseteq$ $\langle$**w-pre - (8)**$\rangle$

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

$\sqsubseteq$ $\langle$**proc**$\rangle$

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

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

if $\mathbf{a = 1}$

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

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

fi

(16) $\sqsubseteq$ $\langle$**i-loc**$\rangle$

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

$\sqsubseteq$ $\langle$**seq**$\rangle$

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

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

(17) $\sqsubseteq$ $\langle$**c-frame, w-pre- (9)**$\rangle$

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

$\sqsubseteq$ $\langle$**skip - (10)**$\rangle$

skip

(18) $\sqsubseteq$ $\langle$**ass - (11)**$\rangle$

$\mathbf{s := 0}$

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

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

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

$(20) \sqsubseteq$ $\langle$**w-pre - (12)**$\rangle$

$s, x : [$ reverse function pre condition, reversen function post condition $]$

$\sqsubseteq$ $\langle$**proc**$\rangle$

reversen(x, s)

$(21) \sqsubseteq$ $\langle$**i-loc, seq**$\rangle$

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

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

$(22) \sqsubseteq$ $\langle$**c-frame, ass - (13)**$\rangle$

b := 1

$(23) \sqsubseteq$ $\langle$**seq**$\rangle$

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

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

$(24) \sqsubseteq$ $\langle$**w-pre - (14)**$\rangle$

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

$\sqsubseteq$ $\langle$**proc**$\rangle$

isPrime(s, b)

$(25) \sqsubseteq$ $\langle$**if**$\rangle$

if b = 1 ∧ s ≠ x

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

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

fi;

$(26) \sqsubseteq$ $\langle$**c-frame, w-pre- (15)**$\rangle$

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

$\sqsubseteq$ $\langle$**ass - (16)**$\rangle$

i := i + 1

r := x

$(27) \sqsubseteq$ $\langle$**c-frame, w-pre- (17)**$\rangle$

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

$\sqsubseteq$ $\langle$**skip - (18)**$\rangle$

skip

proc ISPRIME(**value** r, **result** a) ·

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

(1) $\sqsubseteq$ ⟨**seq, i-loc**⟩

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

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

(2) $\sqsubseteq$ ⟨**ass - (19)**⟩

j := **2**

(3) $\sqsubseteq$ ⟨**seq**⟩

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

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

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

(4) $\sqsubseteq$ ⟨**w-pre - (20)**⟩

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

$\sqsubseteq$ ⟨**skip - (21)**⟩

skip

(6) $\sqsubseteq$ ⟨**w-pre - (22)**⟩

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

$\sqsubseteq$ ⟨**skip - (23)**⟩

skip

(5) $\sqsubseteq$ ⟨**while**⟩

while j $\neq$ **r do**

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

od;

(8) $\sqsubseteq$ ⟨**seq**⟩

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

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

(9) $\sqsubseteq$ ⟨**if**⟩

if r mod j $= 0$

then $\llcorner a : [r \ \mathbf{mod}\ j = 0 \wedge pre(10), post(10)] \lrcorner_{(11)}$

else $\llcorner a : [r \ \mathbf{mod}\ j \neq 0 \wedge pre(18), post(10)] \lrcorner_{(12)}$

fi;

(10) $\sqsubseteq$ ⟨**ass - (24)**⟩

j := **j** $+ 1$

(11) $\sqsubseteq$ ⟨**w-pre - (22)**⟩

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

$\sqsubseteq$ ⟨**ass - (23)**⟩

a := **0**

$$
\begin{array}{ll}
(12) \sqsubseteq & \langle \textbf{w-pre - (24)} \rangle \\
& r, a, j : [\ Inv_2, post(11)\ ] \\
\sqsubseteq & \langle \textbf{skip - (25)} \rangle \\
& \textbf{skip}
\end{array}
$$

We gather the code for the procedure body of EMIRP:

```
EMIRP(r, 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:

```
isPrime(r, 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: $i = i_0 \wedge n > 0 \Rightarrow (i = 1 \wedge n = 0)[^1/_i]$

Proof:

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

$\Rightarrow \langle 1{=}1 \text{ is vacuously true} \rangle$

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

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

$1 = 1 \wedge n > 0$

$\Rightarrow \langle 1 = 1 \wedge i = 1 \Rightarrow [^1/_i] \rangle$

$(i = 1 \wedge n > 0)[^1/_i]$

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

RHS

2.2 Implication 2: $(8) \sqsubseteq (9)$

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

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

Proof:

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

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

$i = 1$

$\Rightarrow \langle \text{We know that 13 is the 1st emirp and from our definition of emirp} \rangle$

$i = 1 \wedge Emirp(13, 1)$

$\Rightarrow \langle \text{This is our Inv with 13 substituted for r} \rangle$

$Inv[^{13}/_r]$

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

RHS

2.3 Implication 3: $(9) \sqsubseteq r := 13$

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

Proof:
$LHS = r = r_0 \wedge Inv[^{13}/_r]$
$\Rightarrow \langle A \wedge B \Rightarrow A \rangle$
$Inv[^{13}/_r]$
$\Rightarrow \langle$clearly$\rangle$
RHS

2.4 Implication 4: $(11) \sqsubseteq skip$

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:
$LHS = Emirp(r, n)$
$\Rightarrow \langle$Since r_0 is the value of r in the precondition, $r = r_0$ in the precondition$\rangle$
$Emirp(r_0, n)$
$\Rightarrow \langle$clearly$\rangle$
RHS

2.5 Implication 6: $Inv[^{x+1}/_x], Inv[^{x+1}/_x] \wedge a = 1 \sqsubseteq a := 1$

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

Proof:

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

$\Rightarrow$ ⟨1=1 is vacuously true⟩

$1 = 1 \wedge a = a_0 \wedge Inv[^{r+1}/_r]$

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

$1 = 1 \wedge Inv[^{r+1}/_r]$

$\Rightarrow$ $\langle 1 = 1 \wedge a = 1 \Rightarrow [^1/_i] \rangle$

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

$\Rightarrow$ ⟨clearly⟩

RHS

2.6 Implication 7: $(20) \sqsubseteq s := 0$

To prove: $s = s_0 \wedge (a = 1 \ \wedge \neg \exists k \in 2..(x -$

$(a = 0 \wedge \exists k \in 2..(x-1)\,(x \ \mathbf{mod}\ k = 0))) \Rightarrow s = 0[^0/_s]$

Proof:

$LHS = s = s_0 \wedge (a = 1 \ \wedge \neg \exists k \in 2..(x -$

$\vee (a = 0 \wedge \exists k \in 2..(x-1)\,(x \ \mathbf{mod}\ k = 0)))$

$\Rightarrow$ ⟨0=0 is vacuously true⟩

$0 = 0 \wedge s = s_0 \wedge (a = 1 \ \wedge \neg \exists k \in 2..(x -$

$\vee (a = 0 \wedge \exists k \in 2..(x-1)\,(x \ \mathbf{mod}\ k = 0)))$

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

$0 = 0$

$\Rightarrow$

$(s = 0)[^0/_s]$

$\Rightarrow$ ⟨clearly⟩

RHS

3 Task 3 - C Code

```
1  #include <stdio.h>
2  #include "reverse.h"
3
4  unsigned long emirp(unsigned long n);
5  void isPrime(unsigned long r, int *a);
6
7  int main (int argc, char* argv[]){
8          unsigned long n;
9          if(scanf("%lu", &n)==1)
10             printf("%lu\n",emirp(n));
11 }
12
13 unsigned long emirp(unsigned long n) {
14     int i = 1;
15     unsigned long x = 13;
16     unsigned long r = 13;
17     while (i != n) {
18         x = x + 1;
19         int a = 1;
20         isPrime(x, &a);
21         if (a == 1) {
22             unsigned long s = 0;
23             reversen(x, &s);
24             int b = 1;
25             isPrime(s, &b);
26             if (b == 1 && s != r)
27                 i = i + 1;
28                 r = x;
29         }
30     }
31         return r;
32 }
33
34 void isPrime(unsigned long r, int *a) {
35     unsigned long j = 2;
36     while (j != r) {
37         if (r % j == 0)
38             *a = 0;
39         j = j + 1;
40     }
41 }
```

- Write something about how the C code relates.

- Compare with examples