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

2 Task 2 - Derivation

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

$(1) \sqsubseteq \quad \langle \text{c-frame} \rangle$

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

$(2) \sqsubseteq \quad \langle \text{i-loc} \rangle$

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

$(3) \sqsubseteq \quad \langle \text{seq} \rangle$

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

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

$(4) \sqsubseteq \quad \langle \text{c-frame} \rangle$

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

$\sqsubseteq \quad \langle \text{ass - } \mathbf{(1)} \rangle$

$\mathbf{i := 1}$

$\mathbf{x := 13}$

$(5) \sqsubseteq \quad \langle \text{seq} \rangle$

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

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

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

$(6) \sqsubseteq \quad \langle \text{w-pre, c-frame - } \mathbf{(2)} \rangle$

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

$\sqsubseteq \quad \langle \text{ass - } \mathbf{(3)} \rangle$

$\mathbf{r := 13}$

$(7) \sqsubseteq \quad \langle \text{while} \rangle$

$\textbf{while i} \neq \textbf{n do}$

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

od;

$(8) \sqsubseteq \quad \langle \text{w-pre, c-frame - } \mathbf{(4)}\ \rangle$

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

$\sqsubseteq \quad \langle \text{skip - } \mathbf{(5)} \rangle$

skip

$(9) \sqsubseteq \quad \langle \text{seq} \rangle$

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

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

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

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

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

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

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

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

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

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

$\llcorner a, i, r, s, b, x : \left[\ pre(21), pre(21) \land b = 1\ \right]\lrcorner_{(22)};$

$\llcorner a, i, r, s, b, x : \left[\ pre(21) \land b = 1, post(21)\ \right]\lrcorner_{(23)}$

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

$\mathbf{b := 1}$

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

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

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

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

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

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

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

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

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

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

else $\llcorner i, r, a, s, b, x : [(b \neq 1 \lor s = x) \land pre(25), post(25)]\lrcorner_{(27)}$

fi;

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

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

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

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

$\mathbf{r := x}$

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

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

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

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**$\rangle$

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

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

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

j := 2

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

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

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

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

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

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

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

skip

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

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

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

skip

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

while j $\neq$ r do

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

od;

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

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

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

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

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;

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

j := j + 1

$(11) \sqsubseteq$ $\langle$**w-pre - (22)**$\rangle$

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

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

a := 0

$(12) \sqsubseteq \quad \langle$ **w-pre - (24)** $\rangle$

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

$\sqsubseteq \quad \langle$ **skip - (25)** $\rangle$

$\quad$ **skip**

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$1=1 is vacuously true$\rangle$

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

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

$1 = 1 \wedge n > 0$

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

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

$\Rightarrow$ $\langle$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:

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

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

$i = 1$

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

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

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

$Inv[^{13}/_r]$

$\Rightarrow$ $\langle$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:

$$
\begin{aligned}
& LHS = r = r_0 \wedge Inv[^{13}/_r] \\
\Rightarrow\ & \langle A \wedge B \Rightarrow A \rangle \\
& Inv[^{13}/_r] \\
\Rightarrow\ & \langle \text{clearly} \rangle \\
& RHS
\end{aligned}
$$

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:

$$
\begin{aligned}
& LHS = Emirp(r, n) \\
\Rightarrow\ & \langle \text{Since } r_0 \text{ is the value of r in the precondition, } r = r_0 \text{ in the precondition} \rangle \\
& Emirp(r_0, n) \\
\Rightarrow\ & \langle \text{clearly} \rangle \\
& RHS
\end{aligned}
$$

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

To prove: a=a$_0 \land Inv[^{r+1}/_r] \Rightarrow (a = 1 \land Inv[^{r+1}/_r])[^1/_a]$

Proof:

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

$\Rightarrow$ ⟨1=1 is vacuously true⟩

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

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

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

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

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

$\Rightarrow$ ⟨clearly⟩

RHS

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

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

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

Proof:

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

$\lor(a = 0 \land \exists k \in 2..(x-1)\,(x \textbf{ mod } k = 0)))$

$\Rightarrow$ ⟨0=0 is vacuously true⟩

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

$\lor(a = 0 \land \exists k \in 2..(x-1)\,(x \textbf{ mod } k = 0)))$

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

$0 = 0$

$\Rightarrow$

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

$\Rightarrow$ ⟨clearly⟩

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(r, &a);
        if (a == 1) {
            unsigned long s = 0;
            reversen(x, &s);
            int b = 1;
            isPrime(s, &b);
            if (b == 1 && s != r)
                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