Assignment 2 - Emirps

Rahil Agrawal z5165505
Aditya Karia zXXXXXXX

COMP2111 18s1

1 Task 1 - Specification Statement

The spec

2 Task 2 - Derivation

proc BS(**value** n, **result** r) ·

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

(1) $\sqsubseteq$ ⟨**c-frame**⟩

$\llcorner p : \left[\begin{array}{l} l \in 0..N \wedge r \in -1..N-1 \wedge l \leq r+1 \wedge s'ed(a[0..N-1]) \\ (a[p] = v \wedge p \in l..r) \vee (p = -1 \wedge \forall x \in l..r\,(a[x] \neq v)) \end{array}\right] \lrcorner_{(2)}$

$\sqsubseteq$ ⟨**if**⟩

if $l = r + 1$

then $\llcorner p : [l = r + 1 \wedge pre(2), post(2)] \lrcorner_{(3)}$

else $\llcorner p : [l \neq r + 1 \wedge pre(2), post(2)] \lrcorner_{(4)}$

fi

(3) $\sqsubseteq$ ⟨**ass**, justified below in Sect. **??**⟩

$p := -1$

(4) $\sqsubseteq$ ⟨**if**⟩

if $a[{}^{(r+l)}/_{2}] = v$

then $\llcorner p : [a[{}^{(r+l)}/_{2}] = v \wedge pre(4), post(4)] \lrcorner_{(5)}$

else $\llcorner p : [a[{}^{(r+l)}/_{2}] \neq v \wedge pre(4), post(4)] \lrcorner_{(6)}$

fi

(5) $\sqsubseteq$ ⟨**ass**, justified below in Sect. **??**⟩

$p := {}^{(r+l)}/_{2}$

(6) $\sqsubseteq$ ⟨**if**⟩

if $a[{}^{(r+l)}/_{2}] < v$

then $\llcorner p : [a[{}^{(r+l)}/_{2}] < v \wedge pre(6), post(6)] \lrcorner_{(7)}$

else $\llcorner p : [a[{}^{(r+l)}/_{2}] \not< v \wedge pre(6), post(6)] \lrcorner_{(8)}$

fi

(7) $\sqsubseteq$ ⟨**s-post**, justified below in Sect. **??**⟩

$p : [pre(7), (post(2))[{}^{(r+l)/2+1}/_{l}]]$

$\sqsubseteq$ ⟨**w-pre**, justified below in Sect. **??**⟩

$p : [(pre(1))[{}^{(r+l)/2+1}/_{l}], (post(2))[{}^{(r+l)/2+1}/_{l}]$

$\sqsubseteq$ ⟨**proc**⟩

$\text{BS}(a, N, v, {}^{(r+l)}/_{2} + 1, r, p)$

(8) $\sqsubseteq$ ⟨**s-post**, justified below in Sect. **??**⟩

$p : [pre(8), (post(2))[{}^{(r+l)/2-1}/_{r}]]$

$\sqsubseteq$ ⟨**w-pre**, justified below in Sect. **??**⟩

$p : [(pre(1))[{}^{(r+l)/2-1}/_{r}], (post(2))[{}^{(r+l)/2-1}/_{r}]$

$\sqsubseteq$ ⟨**proc**⟩

$\text{BS}(a, N, v, l, {}^{(r+l)}/_{2} - 1, p)$

We gather the code for the procedure body of BS:

if $l = r + 1$
then $p := -1$
else if $a[{}^{(r+l)}/_2] = v$
  then $p := {}^{(r+l)}/_2$
  else if $a[{}^{(r+l)}/_2] < v$
    then BS$(a, N, v, {}^{(r+l)}/_2 + 1, r, p)$
    else BS$(a, N, v, l, {}^{(r+l)}/_2 - 1, p)$
    fi
  fi
fi

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 /*
14 var i := 1
15 r := 13
16 while i != n do
17     r := r + 1
18     var a := 1
19     isPrime(r,a)
20     if a = 1 then
21         var s := 0
22         reversen(r,s)
23         var b := 1
24         isPrime(s, b)
25         if b = 1 && s != r then
26             i = i + 1
27 od
28
```

```
29 */
30 unsigned long emirp(unsigned long n) {
31         int i = 1;
32         unsigned long r = 13;
33     while (i != n) {
34         r = r + 1;
35         int a = 1;
36         isPrime(r, &a);
37         if (a == 1) {
38             unsigned long s = 0;
39             reversen(r, &s);
40             int b = 1;
41             isPrime(s, &b);
42             if (b == 1 && s != r) {
43                 i = i + 1;
44             }
45         }
46     }
47         return r;
48 }
49
50 /*
51 var j := 0
52 while j != r do
53     if r mod j = 0 then
54         a = 0
55     j := j + 1
56 od
57 */
58 void isPrime(unsigned long r, int *a) {
59     unsigned long j = 2;
60     while (j != r) {
61         if (r % j == 0) {
62             *a = 0;
63         }
64         j = j + 1;
65     }
66 }
```