2017年11月13日 星期一

Formal Method Project

Formal verification of a sorting function 2

ISEP - XU Haiyin - 9797

ISEP - JI Xiaoyu - 9154

- Outline

The codes are separated into 4 main parts, sort, swap, affiche and main. Before we test the whole codes, we prove these four functions.

- Analyze the sort function

We choose the second function, and the following codes are main sort function

This function has the inner and outer loop. To analyze the variants of each loop, we print the results of step by step.

And an example of once inner loop is:

```
xhy0908deMacBook-Pro:~ xhy0908$ ./tests
i=0, j=0 { 3, 4, 8, 8, 1, 0, 7, 2, 9, 1}
i=0, j=1 { 3, 4, 8, 8, 1, 0, 7, 2, 9, 1}
i=0, j=2 { 3, 4, 8, 8, 1, 0, 7, 2, 9, 1}
i=0, j=3 { 3, 4, 8, 1, 8, 0, 7, 2, 9, 1}
i=0, j=4 { 3, 4, 8, 1, 0, 8, 7, 2, 9, 1}
i=0, j=5 { 3, 4, 8, 1, 0, 7, 8, 2, 9, 1}
i=0, j=6 { 3, 4, 8, 1, 0, 7, 2, 8, 9, 1}
i=0, j=7 { 3, 4, 8, 1, 0, 7, 2, 8, 9, 1}
i=0, j=8 { 3, 4, 8, 1, 0, 7, 2, 8, 1, 9}
i=1, j=0 { 3, 4, 8, 1, 0, 7, 2, 8, 1, 9}
i=1, j=1 { 3, 4, 8, 1, 0, 7, 2, 8, 1, 9}
i=1, j=2 { 3, 4, 1, 8, 0, 7, 2, 8, 1, 9}
i=1, j=3 { 3, 4, 1, 0, 8, 7, 2, 8, 1, 9}
i=1, j=4 { 3, 4, 1, 0, 7, 8, 2, 8, 1, 9}
i=1, j=5 { 3, 4, 1, 0, 7, 2, 8, 8, 1, 9}
i=1, j=6 { 3, 4, 1, 0, 7, 2, 8, 8, 1, 9}
i=1, j=7 { 3, 4, 1, 0, 7, 2, 8, 1, 8, 9}
i=1, j=8 { 3, 4, 1, 0, 7, 2, 8, 1, 8, 9}
```

Figure 1. proof codes of sort function

We can get the conclusion that when 0< j <l-i ==> $\forall_a$: 0 <= a <=j :t[a] <= t[j+1].

Similarly, for the outer loop,(This part is unknown statue)

```
i=0, j=8 { 3, 4, 8, 1, 0, 7, 2, 8, 1, 9}
i=1, j=8 { 3, 4, 1, 0, 7, 2, 8, 1, 8, 9}
i=2, j=0 { 3, 4, 1, 0, 7, 2, 8, 1, 8, 9}
i=3, j=0 { 1, 3, 0, 4, 2, 7, 1, 8, 8, 9}
i=4, j=0 { 0, 1, 3, 2, 4, 1, 7, 8, 8, 9}
i=7, j=0 { 0, 1, 1, 2, 3, 4, 7, 8, 8, 9}
```

So the conclusion is $\forall_{a,b}$: 0 <=b <= l-i <= a <= l-1, t[a]>=t[b]. The result of sort function:

```
/*@ requires l > 0;
    requires \valid(t + (0 .. l - 1));
    ensures
     ∀ int a;
        0 ≤ a < \old(l) ⇒
        (∃ int b;
           0 ≤ b < \old(l) ⇒ \old(*(t + b)) ≡ \at(*(\old(t) + a),Here));
    ensures Sorted(\old(t), 0, \old(l) - 1);
  */
void sort(int *t, int l)
{
  int i;
  int j;
  { /* sequence */
    j = 0;
    i = j;
  }
  i = 0;
  /*@ loop invariant l ≠ 0;
      loop invariant 0 ≤ i ≤ l;
      loop invariant
        0 < i ≤ l ⇒
        (∀ int a, int b;
           0 ≤ b ≤ l - i ≤ a ≤ l - 1 ⇒ *(t + a) ≥ *(t + b));
      loop assigns i, j, *(t + (0 .. l - 1));
  */
  while (i < l) {
    {
      j = 0;
      /*@ loop invariant l ≠ 0;
          loop invariant 0 ≤ j ≤ l - 1;
          loop invariant
            0 < j < l - 1 ⇒
            (∀ int a; 0 ≤ a ≤ j ⇒ *(t + a) ≤ *(t + j));
          loop assigns j, *(t + (0 .. l - 1));
          loop variant l - j;
      */
      while (1) {
        /*@ assert rte: signed_overflow: -2147483648 ≤ l - 1; */
        if (! (j < l - 1)) {
          break;
        }
        {
          /*@ assert rte: mem_access: \valid_read(t + j); */
          /*@ assert rte: signed_overflow: j + 1 ≤ 2147483647; */
          /*@ assert rte: mem_access: \valid_read(t + (int)(j + 1)); */
          if (*(t + j) > *(t + (j + 1))) {
            /*@ assert rte: signed_overflow: j + 1 ≤ 2147483647; */
            swap(t,l,j,j + 1);
          }
        }
        /*@ assert rte: signed_overflow: j + 1 ≤ 2147483647; */
        j ++;
      }
    }
    /*@ assert rte: signed_overflow: i + 1 ≤ 2147483647; */
```

- the result showing on the console:

```
[wp] Proved goals:   70 / 84
  Qed:             44  (0.48ms-15ms-51ms)
  Alt-Ergo:        26  (10ms-74ms-669ms) (876) (interrupted: 12) (unknown: 2)
```

- Reference

https://ulissesaraujo.wordpress.com/2011/02/12/correct-sorting-with-frama-c-and-some-thoughts-on-formal-methdos/

chrome-extension://ikhdkkncnoglghljlkmcimlnlhkeamad/pdf-viewer/web/viewer.html?file=https%3A%2F%2Fframa-c.com%2Fdownload%2Facsl_1.4.pdf