CS569 HW1

Xin Liu (liux4@onid.oregonstate.edu)

April 20, 2015

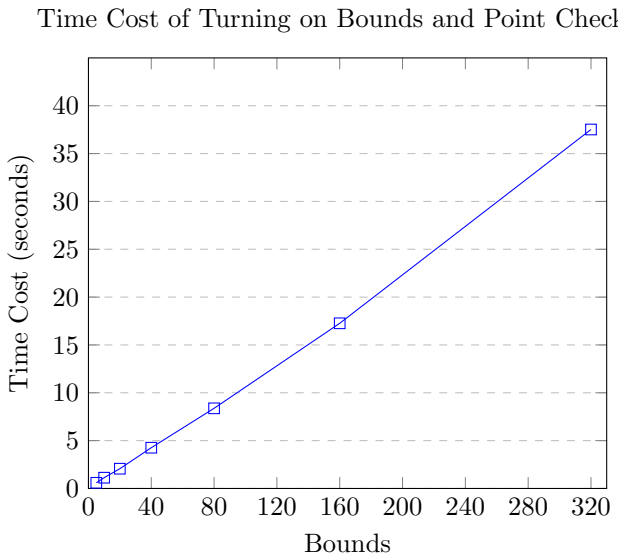
1. Was the program correct or buggy? Does your harness find these bugs?
seqlist.c contains three functions under test, InsertList, DeleteList, and InitList. InsertList and DeleteList functions have bugs related to the index out of the upper bound of array L, respectively. For the details, see the following code snippets. The harness designed for the program verification can find such IndexOutOfBound bugs, when run it independently. However, the two bugs in the following code snippets were not be detected when use CBMC to check harness.c and seqlist.c together with –bounds-check option. Instead, when run CBMC only against seqlist.c directly to check both InsertList and DeleteList functions, the bugs can be dug out.

```
void InsertList(struct Seqlist *L, int x, int i)
{
    ...

    for (j = L->length - 1; j >= i; j--)
        // Bug: when size == ListSize, j+1 will be
        //      greater than the upper bound.
        L->data[j+1] = L->data[j];
    ...
}

void DeleteList ( struct Seqlist *L, int i )
{
    ...

    // Injected Bug: when key == size holds, here is a position error.
```

```
    // Correct: j < L->length
    // Found by --memory-leak-check option. --bounds-check option
    // can also detect here is a bug, but points that the bug is
    // in the body of for-loop.
    for (j = i+1; j <= L->length; j++)
        L->data[j-1] = L->data[j];
    ...
}
```

2. What was hard to specify?
When I used CBMC to check both harness.c and seqlist.c together, it just reported "unwinding assertion loop 0". I don't know how to suppress such a check rule, so that it can reports some other things.

3. How did turning on/off bounds and pointer checker affect cbmc runtime?
When turning on both bounds and pointer check options, the time cost of CBMC has a linear distribution.



4. The details of test result are not listed here, please reference the attached files (click to see).