DD2452 – Lab 1
Mert Alp Taytak – taytak@kth.se

REPORT

Question 1

I tried to follow the given steps to the best of my ability. Unfortunately, some very simple things were the hardest to verify. So, my work contains a lot of trial and error.

Question 2

Amount of time was enough for the task. However, it was a difficult lab. Because sources for FRAMA-C is very lacking. Official documentation does not lend itself to subject look-ups well and the Internet contains surprisingly few questions asked or answered.

Therefore, it was very difficult to try to prove some of the most obvious things. Combined with the fact that FRAMA-C is still a bit of a blackbox to me made it so that I could not understand what exactly is wrong when I could not verify a step.

Question 3

For *tabs*, 200 new lines with some whitespace and 386 goals; for *tabs_loop*, 200 new lines with some whitespace and 178 goals.

Question 4

Keeping it short to an overview, and using the terminology established in the assignment text. *e* and *e'* are restricted to be less than -500, and every membership function other than the ones for *NB* and *NM* return 0 when given such an input. *NB* and *NM* return values such that their sums result in 1000, as long as the input is less than or equal to -500. Therefore, most of the terms disappear when calculating the *numerator* and the *denominator* and whatever is left ends up to be nice numbers. Eventually we get 1000, which is what is wanted by the given property.

Question 5

For *tabs*, 19.3 seconds; for *tabs_loop* 4.3 seconds.

Question 6

I would consolidate summations into loops. While it is generally harder to come up with loop invariants I think working with fewer statements is worth the tradeoff.

DD2452 – Lab 1
Mert Alp Taytak – taytak@kth.se

Question 7

If the question is if I am going to change the way I write code specifically because of this lab, then the answer is no. Because the exercise did not have enough of an impact on me to change mt ways. However, as a general result of Formal Methods course I will be more conscious of writing code that is more easily provable.

Question 8

A majority of input combinations are left unverified. Beyond that, the only inputs that were verified in this lab were not verified in the place they are used, in the driver function. Also, given the nature of embedded hardware and its use case here, verifying against overflows and underflows is important.

Question 9

Yes. In the more broad sense of the question, this analysis makes many assumptions about the hardware. Correctness of the hardware, or even the units of input/output values should be verified before such a system can be considered correct as a whole. Any conclusion can follow from a false assumption, therefore assumptions made in the software must be checked.

Question 10

Yes. It is much slower to verify a program than to write the program and a test suite for it. Meaning, finding a proof takes a lot of time. Also, verification is very interactive and even for small programs runtimes for verification software can grow considerably. Finally, writing programs that are feasible to prove forces the programmer to work with a simpler subset of the language which can be a detriment in many cases.

Question 11

Yes. In if clauses there is no construct that refers to the else clause. Which makes it tedious to write contracts for functions with long if-else chains.

Question 12

Yes. In a simple example, FRAMA-C could not verify something was true when $x < 0$ even though it could verify it when $x < 100$. It is surprising being unable to solve something so simple, and of course nothing remotely close exists on any kind of documentation. In the reverse, I did not expect FRAMA-C to prove long summations for numerator and denominator but it did.

Question 13

Yes. When using the GUI, initial calculations show proofs correctly. However, when refreshed every statement turns to unassigned. You have to restart the GUI to get it working again. Which immediately gets broken after the first refresh. It is a matter of definition if GUI counts as FRAMA-C though. I also suspect that first example to the previous question is a bug.