Critical Systems (CSCM13)
Coursework

Anton Setzer

27 October 2016

General Remark:

- **Deadline: Wednesday, 14 December 2016, 11:00 AM**.
- There are **different coursework** for CSCM13 and CSC313. Please make sure that you attempt the **appropriate coursework.**
- Note you need to submit the coursework **both as a paper copy and electronically** as outlined next.
- As a paper copy you need to submit
  - a printout of your SPARK Ada code,
  - documentation as required by Question 3,
  - in case of Question 1 and 2 you need only submit the code of the last subquestion, i.e. of the final program, unless there are parts of earlier subquestions which are not visible in the final program,
  - all the above you should submit via the **paper courswork submission dropbox**,
  - all the above should be with printed in readable form with a easy to read font size.
- As electronic submission you need to submit
  - the SPARK Ada source code and an electronic pdf version of the paper submission,
    - as one zip archive (**not rar**),
    - subdirectories should not be in zipped form, the only zipped part should be the file you submit,
    - with a clear directory/folder structure (e.g. pdf, question-1, question-2, question-3, with possible subdirectories such as question-1-a etc)
    - with meaningful filenames (not such e.g. example)
    - each subdirectory/folder of the source code should be directly checkable as it is, include the main.gpr file and contain only the .ads .adb files needed for this particular question,
  - all submitted **via Blackboard**.
- Failure to follow the instructions above will result in marks being deducted.
- Subquestions which you have been completed during the lab sessions will be signed off during the lab session.

  - But any marks will be **subject to submission of printouts and the source code** as mentioned before.

- All conditions added (data flow, information flow, verification conditions) are only accepted, if they **express the correctness of the intended program**, as far as it can be expressed by the conditions.

Question 1 Consider the following fragment of code of a function with input X and Y:

```
Aux := X + Y;
if Aux < 360 then return Aux;
else return Aux - 360;
end if;
```

This code accepts two angles which are supposed to be integers between 0 and 359, adds them and computes the resulting angle, which is obtained by possibly deducting 360 in order to be in the range of angles.

(a) Write in Ada a package containing two versions of this program.

- In the first version this code is written as a function which has as input X and Y and returns the value of the result of the above function.
- The second version would be a procedure with 3 parameters, namely X, Y, and the value returned (you might call it Z). You will need to replace the return statements by assignment statements, which set Z to the result returned.
- Add both versions into one package.

The type of integers should be used for all variables and as return value.

[3 marks]

(b) Add files main.adb and main.ads defining a procedure which asks the user for two integer values, and returns the values returned by the function and by the procedure. [3 marks]

(c) Replace the type of integers by the type of angles, which is the range of 0 to 359 of integers.

Note that this requires to modify the code so that the conversion to angles is carried out explicitly when needed.

[3 marks]

(d) Add to your procedure suitable depends clauses so that it passes the information flow analysis by SPARK Ada.

[3 marks]

(e) Add suitable pre and post conditions to your procedure and function, and verify using SPARK Ada that your program is correct w.r.t. these conditions, including range checks.

[4 marks]

Question 2
Consider the following fragment of code:

```
I:= 0;
Res := 0;
loop
  I := I + 1;
  Res := Res + 2;
  exit when Res >= N;
end loop;
```

At the end of this code Res is between N and N+1 and I is half of this value. This means I contains the rounded up value of $\frac{\text{Res}}{2}$. One can achive this result more directly. But this little piece of code is well suitable to explore basic features of SPARK Ada.

(a) Write in Ada one procedure which has parameters N, I, Res and executes this little piece of code.

[3 marks]

(b) Add files main.adb and main.ads defining a procedure which asks the user for two integer values, and returns the values returned by the function and by the procedure.

[3 marks]

(c) Add depends clauses so that your program passes SPARK Ada's information flow analysis. When checking your program, you will get error messages such as

```
loopexample.adb:11:20: medium: overflow check might fail
```

You can ignore such error messages at this stage.

[3 marks]

(d) Add suitable pre and post conditions and intermediate conditions to your procedure and verify using SPARK Ada that your program is correct w.r.t. these conditions. In this subquestion you can still ignore failed overflow checks. Your pre- and post-condition should express that your program is correct, i.t. that at end of this code Res is between N and N + 1 and I is half of this value.

[7 marks]

(e) Add additional verification conditions so that there are no longer any errors regarding overflows. This can be obtained by adding conditions that the variables are in suitable ranges, e.g. that N is in the range 0 .. 1000. In order to pass the verification conditions, you will need to add similar conditions at assert statements, loop invariants and possibly other places in your code.

[5 marks]

Question 3

The goal of this question is to develop a small example of a safety critical system.
Choose a simplified example of a critical system (e.g. a simple control of a rocket or of a nuclear power station, the control of an extremely simple railway system, very simple control of air traffic). You are not expected to understand the technical details of such system apart from what is common sense, and you can make some reasonable assumptions about its behaviour. For instance, when

controlling a rocket, you might demand that the temperature and pressure inside must be between certain values (which you can choose using common sense), that the propulsion is sufficiently strong, etc.

The interface should be simple console input/output. For instance the program might ask for the current temperature, pressure, and then determine an action (like opening of some valve, self-destruction of the rocket etc.). Your program should consist of one loop in which the user is asked for suitable parameters and the results calculated by the system are presented.

The emphasis in this project is not on writing a complicated program with an interesting user interface (restrict yourself to console input and output and a rather simple behaviour, but in demonstrate that you understand how to verify such a program using SPARK Ada.

However some marks are reserved in the last subquestion for really clever solutions. Such solutions need not be long, but have some more interesting aspects to it.

(a) Specify your program taking into account that it is a safety critical system. About 1 - 2 pages are sufficient. [12 marks]

(b) Carry out a hazard analysis of your system using one of the techniques taught in the lecture [10 marks]

(c) Write your program so that it compiles with the compiler supplied as part of the SPARK Ada package. In your printouts include examples of runs of your system, which demonstrates the main features specified in your system. [9 marks]

(d) Add depends clauses so that your program passes SPARK Ada's data and information flow analysis. [7 marks]

(e) Add suitable pre and post conditions and verify using SPARK Ada that your program is correct w.r.t. these conditions. [10 marks]

(f) The following marks are reserved for really good and deep solutions which go beyond. Such solutions need not be very long, but they should have some interesting aspects in it. [15 marks]