Frankfurt University of Applied Sciences
High Integrity Systems

Prof. Dr. Ruth Schorr

4th November 2015

Formal Specification and Verification
Winter Term 2015/16
Exercise Sheet 1

Question 1:

Consider the following Promela model:

```
byte flag = 0;
byte count = 0;
bool end = false

active proctype P() {
  if
      :: flag = 1
      :: flag = 2
  fi;
  if
      :: flag == 1  ->
         do
                        :: (count == 0) -> break
                        :: count++
                        :: count--
         od
      :: else ->
         do
                  :: (count == 0) -> break
                  :: (count != 0) ->
                        if
                              :: count++
                              :: count--
                         fi
         od
  fi;
  end = true
}
```

Consider the following properties:

1. Whenever `flag` is 1, `end` will become **true** at some point thereafter.
2. Whenever `flag` is 2, `end` will become **true** at some point thereafter.
3. If `count` is infinitely often 5, then it is infinitely often 4 or infinitely often 6.
4. `count` can only finitely often be 0.

For each of the properties provide an answer whether or not the property is valid in the transition system described by the Promela program. You don't need to justify your answer.

Question 2:

a) Write your first PROMELA program containing just an active process **`P`**. **`P`** should contain a never ending repetition. In each repetition step there should be a selection between non-deterministic choices. **`P`** should either print the string "**`First\n`**" or the string "**`Second\n`**".

Please check your program using random and *interactive simulation*.

With *interactive simulation* a specific computation can be constructed. Before each step that has a choice point – either because of nondeterminism within a single process or when a choice of the next statement to be executed can be made from several processes – Spin presents the various choices and the user can interactively choose which one to execute. To run an interactive simulation Spin needs to be called with the option –i. Set of choices will be presented as follows:

```
Select a statement
  choice 1: proc 1 (pr) file.pml: 3 (state 1) [specific stmt]
  choice 2: proc 0 (:init:) file.pml: 8 (state 2) [specific stmt]
Select [1-2]
```

b) Add any of the guard values "**`true`**", "**`false`**" and "**`else`**" as guard before the choices of your solution of question 1a. Figure out the behaviour for the different possibilities.

c) Use again your solution of question 1a and terminate the repetition after 10 steps. Verify your solution via running a random simulation.

Question 3:

Consider the following PROMELA program fragment:

```
#define N 10

active proctype P() {
  int array[N];
  int product = 1;

  /* to be added */

  printf("The product is: %d\n", product)
}
```

a) Use the program fragment and initialize all array elements non-deterministically with values between 1 and 10.

b) Use your solutions from question 3a and compute the product of all array elements and store the result in the variable product.

Note: Non-determinism in the initialization may result in different outputs for different runs.

Appendix: Some Linux commands

Files

- **ls** --- lists the files in your current directory
  ls -l --- lists your files in 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.
  ls -a --- lists all files, including the ones whose filenames begin in a dot, which you do not always want to see.
  There are many more options, for example to list files by size, by date, recursively etc.

- **mv** ***filename1 filename2*** --- moves a file, i.e. gives it a different name, or moves it into a different directory (see below

- **cp** ***filename1 filename2*** --- copies a file

- **rm** ***filename*** --- removes a file.

Directories

- **mkdir** ***dirname*** --- make a new directory

- **cd** ***dirname*** --- change directory. You basically 'go' to another directory, and you will see the files in that directory when you do 'ls'. You always start out in your 'home directory', and you can get back there by typing 'cd' without arguments. 'cd ..' will get you one level up from your current position. You don't have to walk along step by step - you can make big leaps or avoid walking around by specifying pathnames.

- **pwd** --- tells you where you currently are.