Available at the COMP3151 website.

Warm-Up Assignment COMP[39]151 11s2

Kai Engelhardt

Revision: 1.1 of Date: 2011/08/02 06:47:24

This assignment is worth 5 marks and due before the logical beginning of week 4, that is, **Wednesday August 17th, 12:59:59** local time Sydney.

The purpose of this assignment is to ensure that every student in this subject is familiar with some of the tools we will be using for the two major assignments.

Task

Recall Exercise 3.1 from the week 3 tutorial.

1. For each of the five parts of this exercise, use `spin` to solve it or aid in the finding of a solution as much as this is feasible.
2. For program **Zero E** from that exercise, state and prove the partial correctness part of the claim using the Owicki/Gries method.

Prepare a report that explains your findings.

Deliverables

`zeroA.pml` – `zeroE.pml` faithful implementations of all five algorithms, **Zero A** – **Zero E**, in Promela. Each of these Promela programs will begin with the line

```
#include "fdef.h"
```

to allow for an external definition of the function f which takes an argument of type `int` and returns and `int`. Do not redefine f inside your programs. Do not attempt to submit `fdef.h`.

`zero.tex` a LaTeX document with your names or student numbers mentioned in the `\author` command. The document should describe your efforts, incorporate the previous deliverables, quote some output of `spin` as evidence for your answers, and contain the Owicki/Gries-style proof.

Submission Instructions

The `give` command to be run is:

```
% 3151
% give cs3151 warmup zero[A-E].pml zero.tex
```

The command above submits the bare minimum. Should you feel the need to include more files, e.g., for vector diagrams, just list them as well.

Hints

- It is allowed to share relevant `fdef.h` files on the forum. Here is a listing of the most basic one used in testing.

```
1 #define f(x) x
```

- It is sometimes helpful not to use `int` typed variables but smaller types such as `byte` or `short` to reduce state space sizes.
- `spin`'s partial order reduction has roughly the same effect as manually enclosing all maximal code fragments that have the LCR in `atomic` blocks.