Formal Methods in Software Development
Exercise 1 (November 8)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

September 9, 2010

The result is to be submitted by the deadline stated above *via the Moodle interface* of the course as a *.zip or .tgz* file which contains

1. a PDF file with
   - a cover page with the course title, your name, Matrikelnummer, and email address,
   - a section for each part of the exercise with the requested deliverables and
   - a (nicely formatted) copy of the ProofNavigator file,
   - for each proof of a formula F, a readable screenshot of the RISC ProofNavigator after executing the command `proof` F,
   - an explicit statement whether the proof succeeded,
   - optionally any explanations or comments you would like to make;
2. the RISC ProofNavigator (.pn) file(s) used in the exercise;
3. the proof directories generated by the RISC ProofNavigator.

Exercise 1a: RISC ProofNavigator

Take the file "exercise1a.pn" and use the RISC ProofNavigator to prove the formulas A, B, and C in this file. The proofs only require the commands `scatter`, `split`, and `instantiate`.

For developing the proofs, you may also try `auto`; the submitted proofs, however, must *not* make use of the `auto` command. Please also try the repeated application of the command `flatten` (rather than `scatter`) to see the gradual decomposition of the proof.

Exercise 1b: Formalization

Develop in the RISC ProofNavigator a theory that formalizes the following argument and checks its correctness:

- If Superman were able and willing to prevent evil, he would do so.
- If Superman were unable to to prevent evil, he would be impotent.
- If Superman were unwilling to prevent evil, he would be malevolent.
- Superman does not prevent evil.
- If Superman exists, then he is neither impotent or malevolent.
- Therefore Superman does not exist.

Use predicates $s(x), a(x), w(x), p(x), i(x), m(x)$ to denote the statements "x is superman/is able to prevent evil/is willing to prevent evil/prevents evil/is impotent/is malevolent". The conclusion of above derivation thus apparently is $\neg\exists x : s(x)$.

Exercise 1c: Verification Conditions

Derive the verification condition(s) for the Hoare triple

$\{x = oldx \wedge y = oldy\}$
`if (x < y) { z = x; x = y; y = z }`
`s = x-y;`
$\{s \geq 0 \wedge (oldx + s = oldy \vee oldy + s = oldx)\}$

Show each step of the derivation (not only the derived conditions).

Then formalize the conditions in the RISC ProofNavigator (declaring integer constants `x:INT`, `oldx:INT`, etc) and check their correctness.