Formal Methods in Software Development
Exercise 1 (October 31)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

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.

Email submissions are *not* accepted.

Exercise 1a: Computer-Supported Predicate Logic Proofs

(30P) 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`. The proof of formula C can be considered a "proof by contradiction": after scattering, the command `flip` may be used to introduce the negation of the goal as an assumption.

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: Formalizing and Proving

(35P) Develop in the RISC ProofNavigator a theory that formalizes each of the following statements $F_1, \ldots, F_6$ as a boolean constant

1. If Superman were able and willing to prevent evil, he would do so.
2. If Superman were unable to to prevent evil, he would be impotent.
3. If Superman were unwilling to prevent evil, he would be malevolent.
4. Superman does not prevent evil.
5. If Superman exists, then he is neither impotent nor malevolent.
6. Superman does not exist.

For instance, F_1 requires a definition

```
F1: BOOLEAN = ...;
```

To write the corresponding definitions, first introduce an undefined type T of objects

```
T: TYPE;
```

and, for each required property, an atomic predicate on T, e.g.,

```
superman: T->BOOLEAN;
```

You can then denote by the atomic formula `superman(x)` the statement "x is Superman".

Finally, define a formula

```
F: FORMULA F1 AND ... AND F5 => F6;
```

and prove it with the same restrictions as in Exercise 1a (the submitted result must not contain applications of command `auto`).

In addition to the commands needed for Exercise 1a, you also need the command `expand`; for easier intuition, you may apply the command `goal` to exchange the goal formula.

Exercise 1c: Deriving and Proving Verification Conditions

(35P) Derive verification conditions for the Hoare triple

$\{n = oldn\}$

```
s = 0; i = n
if (i > 0) { s = s+i; i = i-1; }
if (i > 0) { s = s+i; i = i-1; }
```

$$\left\{\begin{array}{l} n = oldn \wedge \\ (n > 1 \Rightarrow s = 2n - 1 \wedge i = n - 2) \wedge (n = 1 \Rightarrow s = 1 \wedge i = 0) \wedge \\ (n < 1 \Rightarrow s = 0 \wedge i = n) \end{array}\right\}$$

i.e. one or more plain logic formulas whose validity implies the correctness of the Hoare triple.

Show each step of the derivation (not only the derived conditions). *Do not try to "guess" the condition(s) but derive them by application of the Hoare axioms respectively of the predicate transformer calculus. Only such solutions will be rewarded!*

Formalize the conditions in the RISC ProofNavigator (declaring constants `n:INT`, `oldn:INT`, etc.) and prove them.

Hint: when applying the plain Hoare axioms, you have to invent an appropriate assertion before each statement above (this might be tricky for the second conditional statement). However, when applying the predicate transformer calculus, these conditions can be derived mechanically.