PVS by examples: Lab 1

October 20, 2005

- Web site of the module:
  http://www.macs.hw.ac.uk/∼ lilia/ar/ar.html
- Web site of PVS:
  http://pvs.csl.sri.com/
- PVS documentation:
  http://pvs.csl.sri.com/documentation.shtml
- PVS Examples:
  http://www.macs.hw.ac.uk/ ∼ lilia/pvs/exercises/

Task 1: Running PVS

- Step 1: Type **cd**. This changes the current directory to your home directory.
- Step 2: Using a text editor open the following file **.pvs.lisp**, for example by typing **emacs .pvs.lisp**.
- Step 3: Cut and paste the following line in the **.pvs.lisp** file.
  (setq *decision-procedures* (remove 'ics *decision-procedures*))
  See the web site of the module.
- Step 4: That's all! Now type **pvs** and you are ready to run the prover.

Task 2 Download and read the overview of the basic features of PVS from the module website.

Task 3 Prove monotonicity of natural numbers, i.e.

$$\forall k, m, n \in \mathbb{N} : \ m \leq n \rightarrow m + k \leq n + k :$$

- Step 1: Download and save in a file with extension .pvs the corresponding PVS theory available from:

$$http://www.macs.hw.ac.uk/\tilde{l}ilia/pvs/exercises/nat\_xmpls.pvs$$

- Step 2: Using the help in PVS find and read the description of the built in decision procedure *grind*.

$$PVS \rightarrow Getting-help \rightarrow help-PVS-prover-command$$

- prove or x-prove proves the property using grind.
- To prove the property using PVS:

PVS → Prover invocation → Prove

- To see all rules:

PVS → Display commands

Task 4 The sequent $\Gamma \vdash A \vee (A \rightarrow B)$ is written in a file understandable by PVS as follows:

```
BEGIN

A, B: bool

seq1: THEOREM A OR (A IMPLIES B)


END seq_xmpls
```

1. Open a new file, called example1.pvs.
2. Save the representation of $\Gamma \vdash A \vee (A \rightarrow B)$ in example1.pvs.
3. Find and read the descriptions of the commands $flatten, split, propax, instantiate$ (Chapter 3, PVS prover guide).
4. Prove the sequent.

Task 5 Use PVS to prove the formula from Exercise 1
http://www.macs.hw.ac.uk/∼ lilia/pvs/exercises/
Use only the commands *case, flatten, split, propax, instantiate, and skolem*. These are the basic commands that implement the sequent calculus of PVS (see Chapter 3 of PVS prover guide).

Task 6 Prove that appending the results of splitting gives back the original list, i.e. prove the equation

$$app(spilt(n, xs)) = xs$$

Hint: Use induction over both n and xs.