

## PVS Survival Information

### Interrupting the prover

Read Section 3.32 of the *PVS system guide*.

If you think the prover has gotten lost, or that you've asked it to do too much, you can type **C-C C-C** while in the *\*pvs\** buffer. This will bring you to Lisp's top level where you will see:

```
Error: Received signal number 2 (Keyboard interrupt)
  [condition type: INTERRUPT-SIGNAL]
```

```
Restart actions (select using :continue):
```

```
0: continue computation
```

```
[1c]
```

At the `<rc1>` prompt type (**restore**) to get back to the previous sequent.

### Resetting the prover

If you type **M-x reset-pvs** while your cursor is in the *\*pvs\** buffer it interrupts the prover, as above. If you are positioned in another buffer, current and pending PVS commands are aborted.

### Managing proofs

Corresponding to each *.pvs* file is a *.prf* file that holds all the proofs you have done. If you want to see the stored proof, place your purpose on its theorem statement and type **M-x edit-proof**. This will open a buffer containing the Lisp representation of the proof. It looks something like this:

```
( " " (SKOSIMP*)
  (EXPAND "whatever")
  (LIFT-IF)
  (EXPAND "nonesuch")
  (PROPAX) )
```

If you're a real risk taker, you can change the proof and save the result

by typing **c-c c-i** in the `Proof` buffer.

### Stepping through a proof

You can step through a proof one command at a time by issuing either the command **M-x step-proof** or the command **M-x x-step-proof**.

The latter also displays a proof tree.

The PVS window is split into two parts, a buffer named `Proof` containing the proof script, and the `*pvs*` buffer showing the proof in progress.

The command **TAB 1** executes one proof step. The command **ESC n TAB 1** executes  $n$  proof steps.

Repositioning the cursor in the `Proof` window affects which step will be executed next. Therefore, use **c-x o** to switch between buffers rather than using mouse clicks.

### Hidden formulas

The PVS prover command (**hide  $f$** ) removes formula number  $f$  from the sequent. This helps reduce clutter. The command **M-x show-hidden-formulas** opens a buffer named `Hidden` displaying all the hidden formulas. The PVS prover command (**reveal  $f$** ) returns the formula numbered  $f$  in `Hidden` to the sequent.

### Making cosmetic changes

If you want to change the names of variables, formulas, and the like. You must be careful to do it *everywhere*. A procedure for doing this is (caution: needs to be tried):

- 1 Create a `top.pvs` file that imports directly or indirectly all of the theories you are working on.
- 2 Issue the **M-x dump-pvs-files** command in your `top.pvs` theory.
- 3 Make your changes to this dump file (using global match and replace commands, when possible, so you don't miss anything) and save the result.

4 (optional) Issue the Unix command **rm \*.bin** in your directory. This shouldn't be necessary, but there have been problems in the past. If you are the adventurous type, skip this step.

5 In PVS, issue the **M-x undump-pvs-files** command.

6 Enter `top.pvs` and re-typecheck.

The `top.pvs` theory is a good place to issue **M-x prove-inportchain** to reprove everything.