PVS Cheat Sheet

Prerequisites

- PVS 6.0 (Allegro Lisp version is preferred) : `http://pvs.csl.sri.com/download.shtml`
- NASA PVS Library (development version is preferred): `https://github.com/nasa/pvslib`

Emacs Essentials

`C-x C-f`	Load file
`C-x C-s`	Save file
`C-x C-x`	Exit
`C-x i`	Insert file
`C-x u`	Undo
`C-x 2`	Split screen
`C-x 1`	Remove split screen
`C-x k`	Cut to end of line
`C-x y`	Paste
`C-a`	Go to beginning of line
`C-e`	Go to end of line
`C-g`	Cancel command
`C-s`	Incremental find
`M-%`	Find and replace
`M-<`	Go to beginning of file
`M->`	Go to end of file

PVS Emacs Commands

`M-x tc`	Type-check
`M-x tcp`	Type-check and prove TCCs
`M-x prt`	Prove theory
`M-x pri`	Prove theories in import chain
`M-x pr`	Interactive prover
`M-x ste`	Step proof
`M-x pvsio`	PVSio evaluator
`Tab 1`	Execute proof command (in step proof mode)
`M-s`	Auto-completion in prover and PVSio

PVS Proof Commands

`(assert)`	Decision procedure and auto-rewrites
`(case` *expr*`)`	Case analysis on boolean expression *expr*
`(decompose-equality` *fnum*`)`	Decompose equality in formula *fnum*
`(eval-expr` *expr*`)`	Evaluate the ground expr *expr*
`(eval-formula` *fnum*`)`	Evaluate the ground formula in *fnum*
`(expand` *name fnum n*`)`	Expand *n*-th occurrence of *name* in formula number *fnum*
`(expand*` *nm1* `...` *nmn*`)`	Expand *nm1* ... *nmn* everywhere in the sequent
`(grind)`	A super-duper strategy
`(grind-reals)`	Grind plus auto-rewrite with real number properties
`(ground)`	Propositional simplification plus decision procedures
`(help` *command*`)`	Display usage information of proof *command*
`(induct` *var*`)`	Inductive proof on variable *var*
`(inst` *fnum expr1* `...` *exprn*`)`	Instantiate universal formula in *fnum* with *expr1* ... *exprn*
`(lemma` *name*`)`	Introduce lemma called *name*
`(name` *name expr*`)`	Introduce constant *name* and make it equals to *expr*
`(name-replace` *name expr*`)`	Replace *name* by *expr* everywhere in the sequent
`(replace` *fnum*`)`	Left-right replacement of an equality in formula number *fnum*
`(replace` *fnum* `:dir rl)`	Right-left replacement of an equality in formula number *fnum*
`(rewrite` *name*`)`	Left-right rewriting of equality lemma *name*
`(rewrite` *name* `:dir rl)`	Right-left rewriting of equality lemma *name*
`(skeep` *fnum* `:preds? t)`	Eliminate universal quantifier in formula number *fnum* with skolem constants and introduce their type predicates
`(typepred` *expr*`)`	Introduce the type predicate of *expr* into the sequent

PVS Survival Tips

- Grind (or any other proof command) gone wild: `C-c C-c` and then `C-d`.
- PVS lost its mind: `M-x reset-pvs`.
- Solution to most type-checking problems:
  1. Exit PVS Emacs.
  2. Clean binaries from working directory, e.g., issue the Unix shell command `proveit -Clean`

Useful Links

- PVS: `http://pvs.csl.sri.com`
- PVS Source Code: `https://github.com/SRI-CSL/PVS`
- Emacs Basics: `http://doors.stanford.edu/~sr/computing/emacs.html`
- NASA PVS Library at NASA LaRC: `http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library`