PVS Quick Reference

- PVS
  - Enter pvs pvs filename.pvs
  - Exit pvs c-x c-c
  - Abort command c-g
- Context
  - change context m-x cc
  - save context m-x sc
  - context path m-x cp
  - list pvs files m-x lf
- Window
  - split horizontal c-x 2
  - split vertical c-x 3
  - switch c-x o
  - delete windows c-x 1
- Buffers (temp work)
  - list buffers c-x c-b
  - select a buffer c-x b
  - kill buffer c-x k
- File handling
  - create new m-x new-pvs-file
  - save c-x c-s
  - open c-x c-f
- Specification
  - parse m-x parse
  - typecheck m-x typecheck
  - prove m-x prove
- prover
  - exit (exit)
  - continue :continue n
  - try prop strategy (prop)
  - try bddsimp strategy (bddsimp)
  - flatten (flatten fnum)
  - split (split fnum)
  - go to next subgoal (postpone)
  - add premise (lemma "lemma-name")
  - check status of proofs m-x status-pvs-file
  - dump prfs and pvs file m-x dump-pvs-file
  - undump m-x undump-pvs-file

EXAMPLE 1:

```
  myTheory: THEORY   % name of theory is myTHEORY
  BEGIN
    p, q, r: bool       % declaring boolean variables p, q, and r


    axiom_1: AXIOM p IMPLIES q

    axiom_2: AXIOM q IMPLIES r

    th_1:    THEOREM p IMPLIES r

  END myTheory
```

EXAMPLE 2:

```
  addsum: THEORY
  BEGIN

    n: var nat

    sum (n): RECURSIVE nat =
      (if n = 0 THEN 0 ELSE n + sum(n -1) ENDIF)
      MEASURE n

    lem1: LEMMA sum(n) = n*(n+1)/2

  END mySTACK
```

TUE 01/25/05

1. MISC

PVS site? _____

2. WHAT IS PVS?

How will one use PVS?

(say in your own words)

_____

_____

_____

3. SPECIFICATION LANGUAGE

- Difference with programming language?_____
- Know syntax for declaring vars and formulas.
- Difference between AXIOMS and THEOREMS? _____
- SPECIFY:

Given: p->q, q->p

To prove: (p OR q) -> (p AND q)

4. PROVER

What is sequent form? _____

how to move a formula from theory to antecedant?

what commands to use to prove?

5. DEMO

You need to know how to:

- CREATE a pvs specs file
- Use the prover to PROVE a formula given some previously proved formulas or axioms.

Exercise1 : THEORY
BEGIN

$a, b, c, n, t, h, s, p, q$: bool

lem0 : LEMMA $(a \wedge (a \Rightarrow b)) \Rightarrow b$

lem1 : LEMMA $(c \wedge n) \Rightarrow t$

lem2 : LEMMA $h \wedge \neg s$

lem3 : LEMMA $(h \wedge \neg (s \vee c)) \Rightarrow p$

lem4 : LEMMA $(n \wedge \neg t) \Rightarrow p$

lem5 : LEMMA $\neg (p \vee q) \equiv (\neg p \wedge \neg q)$

END Exercise1

Verbose proof for `lem0`.
`lem0:`

{1} $(a \wedge (a \Rightarrow b)) \Rightarrow b$

`lem0:`

{1} $(a \wedge (a \Rightarrow b)) \Rightarrow b$

Applying disjunctive simplification to flatten sequent,
`lem0:`

{-1}	a
{-2}	$(a \Rightarrow b)$
{1}	b

Splitting conjunctions,
we get 2 subgoals:
`lem0.1:`

{-1}	b
{-2}	a
{1}	b

which is trivially true.
This completes the proof of `lem0.1`.
`lem0.2:`

{-1}	a
{1}	a
{2}	b

which is trivially true.
This completes the proof of `lem0.2`.
Q.E.D.

Verbose proof for `lem4`.

`lem4:`

{1} $(n \wedge \neg\ t) \Rightarrow p$

`lem4:`

{1} $(n \wedge \neg\ t) \Rightarrow p$

Applying lem1

`lem4:`

{-1}	$(c \wedge n) \Rightarrow t$
{1}	$(n \wedge \neg\ t) \Rightarrow p$

Applying lem2

`lem4:`

{-1}	$h \wedge \neg\ s$
{-2}	$(c \wedge n) \Rightarrow t$
{1}	$(n \wedge \neg\ t) \Rightarrow p$

Applying lem3

`lem4:`

{-1}	$(h \wedge \neg\ (s \vee c)) \Rightarrow p$
{-2}	$h \wedge \neg\ s$
{-3}	$(c \wedge n) \Rightarrow t$
{1}	$(n \wedge \neg\ t) \Rightarrow p$

Applying disjunctive simplification to flatten sequent,

`lem4:`

{-1}	$(h \wedge \neg\ (s \vee c)) \Rightarrow p$
{-2}	$h \wedge \neg\ s$
{-3}	$(c \wedge n) \Rightarrow t$
{-4}	n
{1}	t
{2}	p

Applying disjunctive simplification to flatten sequent,

`lem4:`

{-1}	$(h \wedge \neg (s \vee c)) \Rightarrow p$
{-2}	h
{-3}	$(c \wedge n) \Rightarrow t$
{-4}	n
{1}	s
{2}	t
{3}	p

Splitting conjunctions,
we get 4 subgoals:
`lem4.1:`

{-1}	p
{-2}	h
{-3}	$(c \wedge n) \Rightarrow t$
{-4}	n
{1}	s
{2}	t
{3}	p

which is trivially true.
This completes the proof of `lem4.1`.
`lem4.2:`

{-1}	h
{-2}	$(c \wedge n) \Rightarrow t$
{-3}	n
{1}	h
{2}	s
{3}	t
{4}	p

which is trivially true.
This completes the proof of `lem4.2`.
`lem4.3:`

{-1}	s
{-2}	h
{-3}	$(c \wedge n) \Rightarrow t$
{-4}	n
{1}	s
{2}	t
{3}	p

which is trivially true.
This completes the proof of `lem4.3`.

`lem4.4:`

{-1}	c
{-2}	h
{-3}	$(c \wedge n) \Rightarrow t$
{-4}	n
{1}	s
{2}	t
{3}	p

Splitting conjunctions,
we get 3 subgoals:
`lem4.4.1:`

{-1}	t
{-2}	c
{-3}	h
{-4}	n
{1}	s
{2}	t
{3}	p

which is trivially true.
This completes the proof of `lem4.4.1`.
`lem4.4.2:`

{-1}	c
{-2}	h
{-3}	n
{1}	c
{2}	s
{3}	t
{4}	p

which is trivially true.
This completes the proof of `lem4.4.2`.
`lem4.4.3:`

{-1}	c
{-2}	h
{-3}	n
{1}	n
{2}	s
{3}	t
{4}	p

which is trivially true.
This completes the proof of `lem4.4.3`.

Q.E.D.

Verbose proof for `lem5`.
`lem5:`

{1} ¬ (p ∨ q) ≡ (¬ p ∧ ¬ q)

`lem5:`

{1} ¬ (p ∨ q) ≡ (¬ p ∧ ¬ q)

Splitting conjunctions,
we get 2 subgoals:
`lem5.1:`

{1} ¬ (p ∨ q) ⊃ (¬ p ∧ ¬ q)

Applying disjunctive simplification to flatten sequent,
`lem5.1:`

{1} p
{2} q
{3} (¬ p ∧ ¬ q)

Splitting conjunctions,
we get 2 subgoals:
`lem5.1.1:`

{-1} p
{1} p
{2} q

which is trivially true.
This completes the proof of `lem5.1.1`.
`lem5.1.2:`

{-1} q
{1} p
{2} q

which is trivially true.
This completes the proof of `lem5.1.2`.
`lem5.2:`

{1} (¬ p ∧ ¬ q) ⊃ ¬ (p ∨ q)

Applying disjunctive simplification to flatten sequent,

`lem5.2:`

{-1}	$(p \vee q)$
{1}	p
{2}	q

Splitting conjunctions,
we get 2 subgoals:
`lem5.2.1:`

{-1}	p
{1}	p
{2}	q

which is trivially true.
This completes the proof of `lem5.2.1`.
`lem5.2.2:`

{-1}	q
{1}	p
{2}	q

which is trivially true.
This completes the proof of `lem5.2.2`.
Q.E.D.