Predicate Logic Proofs in PVS

Outline

- Review: Order of precedence & Dealing with quantifiers
- Universal closure of sentence forms in PVS
- (SKOLEM!) and (INST . . .) - PVS commands for eliminating quantifiers
- Example: Putting it all together
- Rules for dealing with =: (EXPAND . . .) & (REPLACE . . .)

Order of Precedence & Parenthesis

Recall: We use precedence of logical operators and associativity of $\wedge, \vee, \leftrightarrow$ to drop parentheses. It is understood that this is shorthand for the fully parenthesized expressions.

Huth+Ryan uses order of precedence:

$$\begin{array}{c}\forall\\ \exists\\ \neg\end{array}, \begin{array}{c}\wedge\\ \vee\end{array}, \begin{array}{c}\rightarrow\\ \leftrightarrow\end{array}$$

PVS uses order of precedence:

$$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \begin{array}{c}\forall\\ \exists\end{array}$$

$\forall x P(x) \rightarrow \exists y Q(x,y) \wedge P(y)$ becomes:
In Huth+Ryan:

$$(\forall x P(x)) \rightarrow ((\exists y Q(x,y)) \wedge P(y))$$

In PVS:

$$\forall x(P(x) \rightarrow (\exists y(Q(x,y) \wedge P(y))))$$

Review

As we will see, the PVS commands that deal with quantifiers and equality can all be understood in terms of the rules we already know.

	introduction	elmination
$\forall$	$\dfrac{\boxed{\begin{array}{l} x_0 \\ \quad\vdots \\ \quad \phi[x_0/x]\end{array}}}{\forall x \phi}\ \forall i$	$\dfrac{\forall x \phi}{\phi[t/x]}\ \forall e$
$\exists$	$\dfrac{\phi[t/x]}{\exists x \phi}\ \exists i$	$\dfrac{\exists x \phi \quad \boxed{\begin{array}{l} x_0 \quad \phi[x_0/x] \\ \quad\vdots \\ \quad \chi\end{array}}}{\chi}\ \exists e$
$=$	$\dfrac{}{t = t}\ {=}i$	$\dfrac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]}\ {=}e$

Removing & Adding $\forall x$

Rule $\forall e$: If $\Gamma \vdash \forall x \phi$ and $\phi[t/x]$ is a valid substitution then $\Gamma \vdash \phi[t/x]$.

Rule $\forall i$: If $\Gamma \vdash \phi[x_0/x]$, and

1. $x_0 \notin FV(\Gamma)$,
2. and $\phi[x_0/x]$ is a valid substitution.

Then $\Gamma \vdash \forall x \phi$.

Adding and Removing $\exists x$

Rule $\exists i$: If $\Gamma \vdash \phi[t/x]$ and $\phi[t/x]$ is a valid substitution then $\Gamma \vdash \exists x \phi$.

Rule $\exists e$: If $\Gamma, \phi[x_0/x] \vdash \chi$, and

1. $\phi[x_0/x]$ is a valid substitution, and
2. $x_0 \notin FV(\Gamma) \cup FV(\chi)$.

Then $\Gamma, \exists x \phi \vdash \chi$.

PVS Declarations

When proving things in propositional logic in Huth+Ryan, our universe A is a nonempty set or "type" of object.

```
basics  : THEORY
  BEGIN

 A:TYPE+      % Nonempty universe
 x, y:VAR A   % x and y are variables of type A
 a, b: A      % a, b are constant elements of A
 P: PRED[A]   % P is a unary predicate
 Q: PRED[[A,A]] % Q is a binary predicate
 f: [A -> A]  %  f is a function of 1 arg
 h(x):A       %  h is also a function of 1 arg
 g: [[A,A]-> A] % 2-ary function

  END basics
```

Note: PRED[A] is equivalent to:

```
  PRED: TYPE = [A -> bool]
```


Universal Closure in PVS

Def: For ϕ with free variables $x_1, \ldots, x_n$, the formula $\forall x_1 \forall x_2 \ldots \forall x_n \phi$ is the *universal closure of ϕ*. Note that

$$\mathcal{M} \models \phi \text{ iff } \mathcal{M} \models \forall x_1 \forall x_2 \ldots \forall x_n \phi$$

(follows immediately definition of $\models$.)

PVS uses this as a short cut to implicitly quantify theorem statements. E.g.

```
x,y,z:VAR nat
f(x,y):nat = x + y

T1: THEOREM f(x,y)=f(y,x)
```

in prover becomes:

```
T1 :

  |-------
{1} (FORALL (x: nat, y: nat): f(x, y) = f(y, x))

Rule?
```


Universal Closure in PVS (cont.)

Note: You must be careful stating negation of formulas!

Consider the following PVS:

```
x:VAR nat

 P1: PROPOSITION x+x>x
 WrongNotP1: PROPOSITION NOT(x+x>x)
 NotP1: PROPOSITION NOT(FORALL (x:nat): (x+x>x))
```

Neither `P1` nor `WrongNotP1` is provable but `NotP1` is provable. Why? Try proving `WrongNotP1` and due to universal closure you get:

```
WrongNotP1 :

  |-------
{1}   FORALL (x: nat): NOT (x + x > x)

Rule?
```

Predicate Logic Proofs in PVS

Predicate logic proofs are just propositional logic proofs with new rules for eliminating quantifiers.

Still use the commands for propositional rules: (FLATTEN), (SPLIT) & (BDDSIMP).

And add new PVS commands:

(SKOLEM!) does $\exists e$ and $\forall i$.

(INST eq# "term") does $\forall e$ and $\exists i$.

PVS commands: (SKOLEM!)

Let x_0 be a new "variable" (a.k.a skolem constant) not appearing in any of the formulas of sequents of the sequent.

(SKOLEM!) uses $\exists e$ to eliminate $\exists x$ in a premises:

$$
\begin{array}{|c} \phi_1 \\ \phi_2 \\ \vdots \\ \exists x \phi \\ \hline \psi_1 \\ \psi_2 \\ \vdots \end{array}
\quad \overset{\text{(SKOLEM!)}}{\Longrightarrow} \quad
\begin{array}{|c} \phi_1 \\ \phi_2 \\ \vdots \\ \phi[x_0/x] \\ \hline \psi_1 \\ \psi_2 \\ \vdots \end{array}
$$

or $\forall i$ to eliminate $\forall x$ in a conclusion:

$$
\begin{array}{|c} \phi_1 \\ \phi_2 \\ \vdots \\ \hline \psi_1 \\ \psi_2 \\ \vdots \\ \forall x \phi \end{array}
\quad \overset{\text{(SKOLEM!)}}{\Longrightarrow} \quad
\begin{array}{|c} \phi_1 \\ \phi_2 \\ \vdots \\ \hline \psi_1 \\ \psi_2 \\ \vdots \\ \phi[x_0/x] \end{array}
$$

How (SKOLEM!) uses Rule $\forall i$

Rule $\forall i$: If $\Gamma \vdash \phi[x_0/x]$, $x_0 \notin FV(\Gamma)$ and $\phi[x_0/x]$ is valid, then $\Gamma \vdash \forall x \phi$.

To try to prove $\Gamma \vdash \forall x \phi$, (SKOLEM!) says PVS can try to prove $\Gamma \vdash \phi[x_0/x]$ for some "new" x_0 so that

i) $x_0 \notin FV(\Gamma) \cup FV(\phi)$, and

ii) $\phi[x_0/x]$ is a valid substitution.

Why? If

$$
\begin{array}{ll} \Gamma \vdash \phi[x_0/x] & \text{then} \\ \Gamma \vdash \forall x \phi, & \text{by Rule } \forall i. \end{array}
$$

Why make sure $x_0 \notin FV(\phi)$ too?
E.g. Consider formula $\forall x P(x, x_0)$.

Why require $\phi[x_1/x]$ is a valid substitution?
E.g. Consider formula $\forall x \exists x_0 (x < x_0)$

PVS commands: (INST . . .)

Below let t be a term such that $\phi[t/x]$ is valid.

(INST . . .) uses EG to "remove"* $\exists x$ in a conclusion:

$$
\begin{array}{r|c} -1 & \phi_1 \\ -2 & \phi_2 \\ \vdots & \vdots \\ \hline 1 & \psi_1 \\ 2 & \psi_2 \\ \vdots & \vdots \\ n & \exists x \phi \end{array}
\quad \overset{\text{(INST n "t")}}{\Longrightarrow} \quad
\begin{array}{|c} \phi_1 \\ \phi_2 \\ \vdots \\ \hline \psi_1 \\ \psi_2 \\ \vdots \\ \phi[t/x] \end{array}
$$

*The original quantified formula is still available to use in proofs. Its just "hidden". Use the PVS menu command "M-x show-hidden-formulas" to see hidden formulas and the prover command (REVEAL eq#) to use a hidden equation.

PVS commands: (INST . . .)

Below let t be a term such that $\phi[t/x]$ is valid.

(INST) uses $\forall e$ to "remove"* $\forall x$ in a premises:

$$\begin{array}{r|l} -1 & \phi_1 \\ -2 & \phi_2 \\ \vdots & \vdots \\ -n & \forall x\phi \\ \hline 1 & \psi_1 \\ 2 & \psi_2 \\ \vdots & \vdots \end{array} \quad \overset{\text{(INST -n "t")}}{\Longrightarrow} \quad \begin{array}{|c} \phi_1 \\ \phi_2 \\ \vdots \\ \phi[t/x] \\ \hline \psi_1 \\ \psi_2 \\ \vdots \end{array}$$

*Again, the original quantified formula is still available to use in proofs. Its just "hidden". Use (INST-CP -n "t") to keep a copy of original formula in sequent.

How (INST . . .) uses Rule $\forall e$

Rule $\forall e$: If $\Gamma \vdash \forall x\phi$ and $\phi[t/x]$ is a valid substitution then $\Gamma \vdash \phi[t/x]$.

To try to prove $\Gamma, \forall x\phi \vdash \psi$, (INST -n "t") says PVS can try to prove

$$\Gamma, \phi[t/x] \vdash \psi$$

for some t such that $\phi[t/x]$ is a valid substitution. Why?

If

$$\begin{array}{ll} \Gamma, \phi[t/x] \vdash \psi & \text{then} \\ \Gamma, \forall x\phi \vdash \psi, & \text{because} \\ \Gamma, \forall x\phi \vdash \phi[t/x], & \text{by Rule } \forall e \end{array}$$

So you can finish proof of ψ from $\Gamma, \forall x\phi$ by using proof of ψ from $\Gamma, \phi[t/x]$.

Putting it all together

Ex 2 (revisited): Use PVS to prove inconsistency of:

$$\Gamma'' = \{\forall x(P(x) \vee Q(x)), \forall y(\neg P(y) \rightarrow \neg Q(y)), \exists x \neg P(x)\}$$

```
predicate  : THEORY

  BEGIN

  A:NONEMPTY_TYPE
  x,y: VAR A
  P,Q:PRED[A]

  I1: PROPOSITION (FORALL x:P(x) OR Q(x))&
    (FORALL y:NOT P(y) IMPLIES NOT Q(y)) &
    (EXISTS x:NOT P(x)) IMPLIES FALSE

  END predicate
```



```
I1 :

  |-------
{1}    (FORALL x: P(x) OR Q(x))
         & (FORALL y: NOT P(y) IMPLIES NOT Q(y)) &
         (EXISTS x: NOT P(x)) IMPLIES FALSE

Rule? (FLATTEN)
Applying disjunctive simplification to flatten sequent,
this simplifies to:
I1 :

{-1}    (FORALL x: P(x) OR Q(x))
{-2}    (FORALL y: NOT P(y) IMPLIES NOT Q(y))
{-3}    (EXISTS x: NOT P(x))
  |-------

Rule? (SKOLEM!)
Skolemizing,
this simplifies to:
I1 :

[-1]    (FORALL x: P(x) OR Q(x))
[-2]    (FORALL y: NOT P(y) IMPLIES NOT Q(y))
  |-------
{1}    P(x!1)
```

```
Rule? (INST -1 "x!1")
Instantiating the top quantifier in -1 with the terms:
 x!1,
this simplifies to:
I1 :

{-1}    P(x!1) OR Q(x!1)
[-2]    (FORALL y: NOT P(y) IMPLIES NOT Q(y))
  |-------
[1]    P(x!1)

Rule? (INST - "x!1")
Instantiating the top quantifier in - with the terms:
 x!1,
this simplifies to:
I1 :

[-1]    P(x!1) OR Q(x!1)
{-2}    NOT P(x!1) IMPLIES NOT Q(x!1)
  |-------
[1]    P(x!1)

Rule? (BDDSIMP)
Applying bddsimp,
this simplifies to:
I1 :

{-1}    FALSE
  |-------

which is trivially true.
Q.E.D.
```


PVS Commands for Dealing with =

(EXPAND "t1") and (EXPAND "t1" "t2" ...)

```
equality: THEORY
  BEGIN
  x,y:VAR real
  a:real=1
  f(x,y):real = x+y
  g(x,y):real = x+y

  Ia: THEOREM f(y,a)=g(y,1)
  END equality
```

To prove THEOREM Ia you can just use (SKOLEM!) to eliminate universal quantifiers and then use variants of the (EXPAND ...) command to expand definitions (EXPAND* "f" "g") (EXPAND "a").

PVS Commands for Dealing with =

Q: How do you use premises with top level "=" in PVS that are not definitions?

A: The PVS equivalent of Huth+Ryan's $=e$ rule, a.k.a. "Substitution of Equals", is:

(REPLACE -n *)

If equation -n in the premises is of the form

$$t_L = t_R$$

The command makes all valid substitutions of t_R for t_L in all other formulas of the sequent!

Changing the above command to

(REPLACE -n * RL)

would replace right-to-left, performing all valid substitutions of t_L for t_R.

Example: Rubin p.244 E11

```
equal11 : THEORY
  BEGIN
  A:TYPE+
  P:PRED[A]
  A,B,C,D:PRED[A]
  x,y : VAR A
  E11: THEOREM (FORALL x,y:A(x)&B(y)=> x=y)
 &(EXISTS x:A(x)&C(x)) & (EXISTS x:B(x)&D(x))
    =>(EXISTS x:C(x)&D(x))
  END equal11
```

Using a combination of (BDDSIMP), (SKOLEM!) and (INST?) reduces E11 to sequent

```
{-1}  A(x!1)
{-2}  B(x!2)
{-3}  x!1 = x!2
{-4}  C(x!1)
{-5}  D(x!2)
  |-------
{1}   D(x!1)
```

Now you can finish off the proof by replacing `x!1` by `x!2` as follows:

```
Rule? (REPLACE -3 * LR)
Replacing using formula -3,
this simplifies to:
E11 :

{-1}  A(x!2)
[-2]  B(x!2)
[-3]  x!1 = x!2
{-4}  C(x!2)
[-5]  D(x!2)
  |-------
{1}   D(x!2)

which is trivially true.
Q.E.D.
```


Failed Proofs & Counter Examples

Suppose you are asked if the a proof exists for the following sequent:

$$\begin{array}{c}\exists x[E(x) \land \forall y(F(y) \to G(x,y))], \\ \forall x \forall y[E(x) \to (G(x,y) \leftrightarrow H(y))]\end{array} \stackrel{?}{\vdash} \forall x(F(x) \leftrightarrow H(x))$$

Putting this into PVS and trying to prove it results can results the sequent:

```
[-1]  E(x!1)
{-2}  G(x!1, x!2)
{-3}  H(x!2)
  |-------
{1}   F(x!2)

Rule?
```

This sequent would be true if

$$E(x_1), G(x_1, x_2), H(x_2) \vdash F(x_2)$$

or

$$\vdash E(x_1) \land G(x_1, x_2) \land H(x_2) \to F(x_2)$$

Failed Proofs & Counter Examples

Just as we had for propositional logic, syntax and semantics agree so

$$\Gamma \vdash \psi \text{ iff } \Gamma \models \psi$$

Thus

$$\vdash E(x_1) \land G(x_1, x_2) \land H(x_2) \to F(x_2)$$

iff

$$\models E(x_1) \land G(x_1, x_2) \land H(x_2) \to F(x_2)$$

which by definition of $\models$ holds iff

$$\models \forall x_1 \forall x_2 (E(x_1) \land G(x_1, x_2) \land H(x_2) \to F(x_2))$$

but this would mean that for every model $\mathcal{M}$

$$\mathcal{M} \models \forall x_1 \forall x_2 (E(x_1) \land G(x_1, x_2) \land H(x_2) \to F(x_2))$$

But we can find an $\mathcal{M}$ such that:

$$\mathcal{M} \not\models \forall x_1 \forall x_2 (E(x_1) \land G(x_1, x_2) \land H(x_2) \to F(x_2))$$

Check that this model provides a counter example to the original sequent!