

Kurzanleitung zu PVS Teil 2

PVS Beweisregeln (Fortsetzung)

Regeln für Definitionen und Lemmata

Beweisregeln: (EXPAND) und (EXPAND*)

Diese Regeln dienen zur Expansion von Definitionen. Die zu expandierenden Vorkommen des Funktionsnamens können auf einzelne Formeln (mit :fnum) und bestimmte Stellen innerhalb dieser Formeln (mit :occurrence) eingeschränkt werden.

```
% Im Beispiel ist f(n) = n + 1

beispiel :

{-1} f(n!1) > g(a)
|-----
{1} f(g(a)) - 1 < f(n!1)

Rule? (expand "f" :fnum 1 :occurrence 1)
Expanding the definition of f,
this simplifies to:
beispiel :

[-1] f(n!1) > g(a)
|-----
{1} g(a) < f(n!1)
```

Mit der zweiten Form können alle angegebenen Namen auf einmal expandiert werden.

```
% Im Beispiel ist f(n) = n + 1 und g(n) = n * n

beispiel :

|-----
{1} g(f(n!1)) = f(n!1) * f(n!1)

Rule? (expand* "f" "g")
Expanding the definition(s) of (f g),
Q.E.D.
```

(EXPAND FUNCTION-NAME &OPTIONAL

((FNUM *) OCCURRENCE IF-SIMPLIFIES ASSERT?):

Expands (and simplifies) the definition of FUNCTION-NAME at a given OCCURRENCE. If no OCCURRENCE is given, then all instances of the definition are expanded. The OCCURRENCE is given as a number n referring to the n th occurrence of the function symbol counting from the left, or as a list of such numbers.

If the IF-SIMPLIFIES flag is T, then any definition expansion occurs only if the RHS instance simplifies (using the decision procedures). Note that the EXPAND step also applies simplification with decision procedures (i.e. SIMPLIFY with default options) to any sequent formulas where an expansion has occurred.

ASSERT? can be either NONE (meaning no simplification), NIL (meaning simplify using SIMPLIFY), or T (meaning simplify using ASSERT).

(EXPAND*/\$ &REST NAMES) :

Expands all the given names and simplifies.

Beweisregel: (LEMMA)

Führt ein Lemma in den Antezedenten ein. Optional kann eine partielle Instanzierung angegeben werden, vgl. (INST?).

beispiel :

|-----

{1} $x!1 * x!1 > 0$

Rule? (lemma " pos_times_gt")

Applying pos_times_gt

this simplifies to:

beispiel :

{-1} FORALL (x, y: real):

$x * y > 0$ IFF ($0 > x$ AND $0 > y$) OR ($x > 0$ AND $y > 0$)

|-----

[1] $x!1 * x!1 > 0$

(LEMMA NAME &OPTIONAL (SUBST)):

Adds lemma named NAME as the first antecedent formula after applying the substitutions in SUBST to it.

Example: (LEMMA "assoc" ("x" 1 "y" 2 "z" 3)).

Beweisregel: (USE)

Kombiniert (LEMMA) und (INST?).

```
beispiel :  
  |-----  
{1}  x!1 * x!1 > 0  
  
Rule? (use " pos_times_gt")  
Using lemma pos_times_gt,  
this simplifies to:  
beispiel :  
{-1}  x!1 * x!1 > 0 IFF (0 > x!1 AND 0 > x!1) OR (x!1 > 0 AND x!1 > 0)  
  |-----  
[1]  x!1 * x!1 > 0
```

```
(USE/$ LEMMA &OPTIONAL SUBST (IF-MATCH BEST) (INSTANTIATOR INST?)) :  
  Introduces lemma LEMMA, then does BETA and INST? (repeatedly) on the lemma.  
  The INSTANTIATOR argument may be used to specify an alternative to INST?.
```