La généralisation de théorèmes

Olivier Pons

INRIA Sophia Antipolis

Réunion du Goupe Bip
Mardi 3 Mars 1998
Paris

Deux lemmes sur des fonctions de N dans N

```
Lemma plus_permute :(n,m,p:nat)((plus n (plus m p))=(plus m (plus n p))).
Proof.
Intros; Rewrite (plus_assoc_l m n p); Rewrite (plus_sym m n); Auto.
Qed.

Lemma mult_permute :(n,m,p:nat)((mult n (mult m p))=(mult m (mult n p))).
Proof.
Intros;Rewrite -> (mult_assoc_l m n p);Rewrite -> (mult_sym m n);Auto.
Qed.
```

Sur le terme de preuve du second lemme

```
Proof: [n,m,p:nat]
        (eq_ind_r nat (mult (mult m n) p)
          [n0:nat](mult n (mult m p))=n0
          (eq_ind_r nat (mult n m)
            [n0:nat](mult n (mult m p))=(mult n0 p)
            (mult_assoc_l n m p) (mult m n) (mult_sym m n))
          (mult m (mult n p)) (mult_assoc_l m n p))

generalize_permute < Check mult_sym.
mult_sym
     : (n,m:nat)(mult n m)=(mult m n)

generalize_permute < Check mult_assoc_l.
mult_assoc_l
     : (n,m,p:nat)(mult n (mult m p))=(mult (mult n m) p)
```

Le lemme generalise et sa preuve

```
Lemma generalized_permute :
(n,m,p:nat)
(f:nat->nat->nat)
(f_assoc_l :(n,m,p:nat) (f n (f m p))=(f (f n m) p))
(f_sym :(n,m:nat)(f n m)=(f m n))
((f n (f m p))=(f m (f n p))).
```

```
Intros.
Exact
        (eq_ind_r nat (f (f m n) p)
          [n0:nat](f n (f m p))=n0
          (eq_ind_r nat (f n m)
            [n0:nat](f n (f m p))=(f n0 p)
            (f_assoc_l n m p) (f m n) (f_sym m n))
          (f m (f n p)) (f_assoc_l m n p))
```

Le terme de preuve ainsi géneré

```
Proof: [n,m,p:nat]
        [f:nat->nat->nat]
         [f_assoc_l:(n0,m0,p0:nat)(f n0 (f m0 p0))=(f (f n0 m0) p0)]
          [f_sym:(n0,m0:nat)(f n0 m0)=(f m0 n0)]
           (eq_ind_r nat (f (f m n) p) [n0:nat](f n (f m p))=n0
             (eq_ind_r nat (f n m) [n0:nat](f n (f m p))=(f n0 p)
               (f_assoc_l n m p) (f m n) (f_sym m n)) (f m (f n p))
             (f_assoc_l m n p))
```

Ou l'on abstrait encore !!!

```
Lemma more_generalized_permute :
(E:Set)
(n,m,p:E)
(f:E->E->E)
(f_assoc_l :(n,m,p:E) (f n (f m p))=(f (f n m) p))
(f_sym :(n,m:E)(f n m)=(f m n))
((f n (f m p))=(f m (f n p))).

Do 7 Intro.
Exact
        (eq_ind_r E (f (f m n) p)
          [n0:E](f n (f m p))=n0
          (eq_ind_r E (f n m)
            [n0:E](f n (f m p))=(f n0 p)
            (f_assoc_l n m p) (f m n) (f_sym m n))
          (f m (f n p)) (f_assoc_l m n p))
```

Ou l'on pousse à l'extrême

```
Lemma  more2_generalized_permute :
(E:Set)
(n,m,p:E)
(f:E->E->E)
(R:(A:Set)A->A->Prop)
(R_ind_r:(A:Set)(x:A)(P:A->Prop)(P x)->(y:A)(R A y x)->(P y))
(f_assoc_l :((n,m,p:E)(R E (f n (f m p)) (f (f n m) p))))
(f_sym :(n,m:E)(R E (f n m) (f m n)))
(R E(f n (f m p)) (f m (f n p))).

Do 9 Intro.
Exact
        (R_ind_r E (f (f m n) p)
          [n0:E](R E (f n (f m p)) n0)
          (R_ind_r E (f n m)
            [n0:E](R E (f n (f m p)) (f n0 p))
            (f_assoc_l n m p) (f m n) (f_sym m n))
          (f m (f n p)) (f_assoc_l m n p)).
```

INRIA

l'instanciation

```
Lemma mult_permute:(n,m,p:nat)((mult n (mult m p))=(mult m (mult n p))).
Intros.
Exact (more2_generalized_permute
           nat n m p mult eq eq_ind_r mult_assoc_l mult_sym).
Subtree proved!
```