Formalisation de l'algèbre de Grassman en Coq

Sylvain Charneau Laurent Fuchs Laurent Théry

Introduction

Dualité

Factorisation

Corps

```
Structure Field := {
     K:> Set;
     0:  K;
     1:  K;
    ?=:  K → K → bool;
     -:  K → K;
     +:  K → K → K;
     *:  K → K → K;
  _⁻¹:  K → K
passoc:  forall x y z, (x + y) + z = x + (y + z);
        ...
}
```

Espace Vectoriel

```
Structure VectorSpace := {
     V:> Set;
     K: Field;
     0:  V;
    ?=:  V → V → bool;
     +:  V → V → V;
    .*:  K → V → V;
           ...
sdistr:  forall k x y,
  k .* (x + y) = (k .* x) + (k .* y);
           ...
}
```

Espace Vectoriel K^n

Function K^n := if n is $n_1 + 1$ then K $\times$ K^{n_1} else K.

$$K^2 \approx K^{S(S\,0)} \approx K \times K^1 \approx K \times (K \times K) \approx K \times K \times K$$

Function x $+_n$ y : K^n := if n is $n_1 + 1$ then
  let (k_x, x_1) := x in
  let (k_y, y_1) := y in (k_x + k_y, x_1 $+_{n_1}$ y_1)
  else x + y

Algèbre de Grassmann G^n

```
Function G^n := if n is n_1 + 1 then G^{n_1} × G^{n_1} else
    K.
```

$$G^2 \approx G^1 \times G^1 \approx ((K \times K) \times (K \times K)) \approx K \times K \times K \times K$$

```
Function x +_n y :  G^n := if n is n_1 + 1 then
    let (x_1, x_2) := x in
    let (y_1, y_2) := y in (x_1 +_{n_1} y_1, x_2 +_{n_1} y_2)
    else x + y
```

Algèbre de Grassman

```
Function 0_n : G^n :=
if n is n1+1 then (0_n1,0_n1) else 0.

Function 1_n : G^n :=
if n is n1+1 then (0_n1,1_n1) else 1.

Function [x]_n : G^n := if n is n1+1 then
let (x1, x2) := x in (0_n1,[x2]_n1)
else x.
```

Produit exterieur

$$a \wedge b \rightsquigarrow \quad a \ * \ b$$

Propriétés:

$e^i \ * \ e^j \ = \ 0$

$e^i \ * \ e^j \ = \ - \ e^j \ * \ e^i$

Produit exterieur

Function $\mathtt{e}_n^i$: G^n := if n is n_1+1 then
    if i is i_{1+1} then $(\mathtt{0}_{n_1}, \mathtt{e}_{n_1}^{i_1})$ else $(\mathtt{1}_{n_1}, \mathtt{0}_{n_1})$
else 1.

Function $(x : G^n)\uparrow$: G^{n+1} := $(\mathtt{0}_n, x)$.

$$(x_1, x_2) = \mathtt{e}^0 * x_1\uparrow + x_2\uparrow$$

Produit exterieur

$(x_1, x_2) * (y_1, y_2) = (\mathrm{e}^0 * x_1\uparrow + x_2\uparrow) * (\mathrm{e}^0 * y_1\uparrow + y_2\uparrow)$

$= \mathrm{e}^0 * x_1\uparrow * y_2\uparrow + x_2\uparrow * \mathrm{e}^0 * y_1\uparrow + x_2\uparrow * y_2\uparrow$

Function $\overline{x}^b_n : G^n$:= if n is $n_1 + 1$ then
let (x_1, x_2) := x in $(\overline{x1}^{\neg b}_{n_1}, \overline{x2}^{b}_{n_1})$
else if b then $-x$ else x.

Function $x *_n y$: G^n := if n is $n_1 + 1$ then
let (x_1, x_2) := x in
let (y_1, y_2) := y in
$(x_1 *_{n_1} y_2 +_{n_1} \overline{x2}^{\perp}_{n_1} *_{n_1} y_1, x_2 *_{n_1} y_2)$
else $x * y$.

Produit exterieur

```
Function x *₂ y :=
     let ((x1,x2), (x3, y4)) := x in
     let ((y1,y2), (y3, y4)) := y in
     ((x1 * y4 + x2 * y3 - x3 * y2 + x4 * y1,
          x2 * y4 + x4 * y2),
       (x3 * y4 + x4 * y3, x4 * y4)).
```

Dualité

Function $\overline{x}_n$:= if n is $n_1 + 1$ then
let (x_1, x_2) := x in $(\overline{x_2}_{n_1}, \overline{x_1}_{n_1})$
  else x

Function $\overline{x}_2$:=
let $((x_1,x_2), (x_3, x_4))$:= x in
$((x_4,x_3), (x_2, x_1))$.

Function $\overline{x}_3$:=
let $(((x_1,x_2), (x_3, x_4)),((x_5,x_6), (x_7, x_8)))$:= x in
$(((x_8,x_7), (x_6, x_5)),(x_4,x_3), (x_2, x_1))$.

Produit interieur

$$a \vee b = \overline{\overline{a} \wedge \overline{b}}$$

Function x #$_n$ y : G^n := if n is $n_1 + 1$ then
let (x_1, x_2) := x in
let (y_1, y_2) := y in
(x_1 #$_{n_1}$ y_1, x_2 #$_{n_1}$ y_1 +$_{n_1}$ $\overline{x_2}^{\perp}_{n_1}$ #$_{n_1}$ y_1)
else x * y.

Homogénéité

a + c * d + d * e * f + e * h

Function hom$_n$ k x := if n is $n_1 + 1$ then
  let (x_1, x_2) := x in
  if k is $k_1 + 1$ then hom$_{n_1}$ k_1 x_1 && hom$_{n_1}$ k x_2
  else x_1 ?= 0 && hom$_{n_1}$ 0 x_2
else k ?= 0 || x ?= 0

Factorisation

$$v * M = 0 \Rightarrow M = v * M'$$

conditions: hom 1 v, hom k M

```
Function factor1_n v M := if n is n_1 + 1 then
  let (v_1, v_2) := v in
  let (M_1, M_2) := M in
  if v_2 ?= 0 then (0, [v_1]^-1 .* M_1) else
  let M'_2 := factor1_{n_1} v_2 M_2 in
  (factor1_{n_1} v_2 ([v_1] .* M'_2 -_{n_1} M_1), M'_2)
else v^-1 * M
```

Obtenir un facteur

Function $\texttt{get}_n$ k M: option G_n := if n is n_1+1 then

  if M ?= 0 then None else

  if k is k_1+1 then

    let (M_1, M_2) := M in

    if $\texttt{get}_{n_1}$ k_1 M_1 is Some v_1 then Some (0, v_1) else

    if $\texttt{get}_{n_1}$ k M_2 is Some v_2 then Some (0, v_2) else

    None

  else Some M

Décomposition

```
Function decompose k M:= if k is k1 + 1 then
  if get_n k1 M is Some v
  then v::decompose k1 (factor v M)
  else []
else []
```