

F-algèbres et F-coalgèbres

April 2024

Table des matières

1	Introduction au langage catégorique	3
1.1	Catégorie	3
1.1.1	Exemples de catégories	3
1.2	Foncteurs	4
1.3	Produits, coproduits	5
2	F-algèbres	6
2.1	Définition et exemples	6
2.1.1	Exemples (et non exemple) algébriques	6
2.1.2	Types inductifs	8
2.2	Initialité dans les F-algèbres	9
2.2.1	Récurrence	10
2.2.2	Fold	11
3	F-coalgèbres	13
3.1	Définition et exemples	13
3.1.1	F-algèbres initiales	13
3.1.2	Un exemple algébrique : les groupes quotient	14
3.1.3	Types coinductifs	14
4	Bibliographie	16

1 Introduction au langage catégorique

La théorie des catégories est un sujet vaste et difficile à appréhender, si j'en crois mon expérience. Il existe de nombreux ouvrages qui permettent à un étudiant de s'y introduire, mais j'ai rencontré plus de succès en regardant seulement ce dont j'avais besoin au moment où j'en avais besoin. D'où la présence de cette partie : tout ce dont on a besoin pour lire ce document est expliqué ici.

1.1 Catégorie

Une catégorie sert à modéliser une *structure*. La notion est très large, c'est l'objectif, d'où l'intérêt d'assimiler la définition et de se cantonner à l'étude d'exemples (de toute façon, la théorie des catégories sert principalement à modéliser, donc comprendre ces modélisations est le plus important). Une catégorie C est la donnée de 5 éléments :

- Une classe d'objets $Ob(C)$. On a besoin de définir une classe afin d'éviter certains paradoxes ensemblistes classiques (paradoxe du barbier, par exemple), mais dans ce document on pourra autant utiliser un ensemble.
- Une classe de morphismes (ou flèches) entre ces objets, appelée $mor(C)$. Souvent, il s'agit d'applications qui préservent la structure des objets.
- Deux applications de $mor(C)$ dans $Ob(C)$: la source et le but (qui permettent de donner la source et le but d'un morphisme). L'ensemble des morphismes d'un objet A dans un objet B est noté $Hom(A, B)$
- Une flèche identité pour tous les objets : $Hom(A, A)$ contient toujours l'identité.
- Une loi de composition pour les morphismes qui est associative et qui admet les identités pour neutre

Enfin, dernières notions dont nous aurons besoin : un objet initial I d'une catégorie C est un objet tel que pour tout objet X , il existe un unique morphisme de I dans X . Un objet final F d'une catégorie C est un objet tel que pour tout objet X , il existe un unique morphisme de X dans F .

1.1.1 Exemples de catégories

Les exemples de catégories sont très nombreux, c'est le but de cette théorie : pouvoir modéliser le plus de choses possibles. Voici cependant 3 exemples communs.

La catégorie des ensembles, **Set**. Les objets sont des ensembles, les morphismes des applications, l'identité d'un objet est l'application identité définie sur un ensemble. La loi de composition est la composition usuelle. Les singletons sont tous des objets terminaux de cette catégorie : quel que soit l'ensemble E qu'on choisit, on peut définir une seule fonction de ce dernier dans un singleton (celle qui envoie E entier sur le seul élément du singleton). La catégorie n'admet en revanche qu'un seul objet initial : l'ensemble vide. Tout au long du document, on nommera 1 le singleton $\{*\}$ dans la catégorie des ensembles.

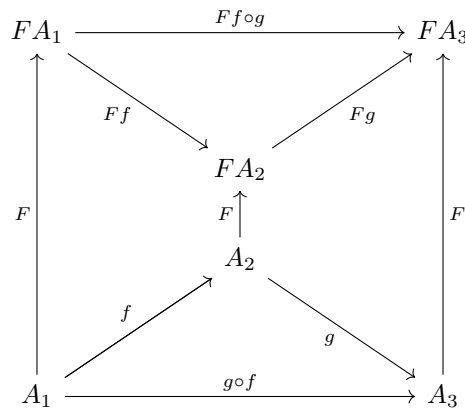
La catégorie des groupes, notée **Grp**. Les objets sont les groupes, les flèches

les morphismes de groupe, l'identité d'un objet est le morphisme identité, la composition est celle des morphismes (attention, syntaxiquement ce n'est pas celle des applications, puisqu'une composition de morphisme est un morphisme, et non pas une simple application). Cette catégorie admet un seul objet final : le groupe trivial à un élément, pour n'importe quel groupe G , le morphisme trivial est l'unique flèche de G dans $\{1\}$. La catégorie admet également un objet initial en la personne de $\{1\}$: pour tout groupe G , on peut définir un unique morphisme (le morphisme trivial) qui envoie le groupe trivial sur G .

"MétroToulousain" est la catégorie dont les objets sont des stations de métro, les flèches des déplacements entre deux stations (éventuellement identiques). Alors, l'identité existe pour tous les objets : on se déplace d'une station X à la station X , et la loi de composition est la suivante : le déplacement $X \rightarrow Y$ et $Y \rightarrow Z$ se compose en $X \rightarrow Z$. La catégorie n'admet aucun objet initial : pour n'importe quel couple de stations X, Y , plusieurs chemins permettent d'accéder de X à Y . Elle n'admet pas non plus d'objet terminal, pour cette même raison.

1.2 Foncteurs

Un foncteur F d'une catégorie C vers une catégorie D est constitué d'un objet FA de D pour tout objet A de C , et d'un morphisme $Ff : FA \rightarrow FB$ pour tout morphisme $A \rightarrow B$ de C , en respectant la composition : pour tout couple α, β de morphismes de C , ($\alpha : X \rightarrow Y$, $\beta : Y \rightarrow Z$), $F(\beta \circ \alpha) = F(\beta) \circ F(\alpha) : F(X) \rightarrow F(Z)$. Les identités des objets doivent également être envoyées sur les identités de leurs images. Le respect de la composition pour les foncteurs s'exprime par la commutativité du diagramme suivant pour $F : C_1 \rightarrow C_2$, A_1, A_2, A_3 des objets de C_1 et $f \in \text{Hom}(A_1, A_2)$, $g \in \text{Hom}(A_2, A_3)$:



Voici des exemples de foncteurs courants :

- Pour toute catégorie C , un foncteur identité de C dans C est défini : il envoie tout élément sur lui même, et tout morphisme sur lui même.
- Les foncteurs constants envoient tous les objets de la catégorie de départ sur le même objet de la catégorie d'arrivée et chaque flèche de la catégorie

de départ sur l'identité de l'objet image. C'est l'objet terminal de la catégorie des foncteurs.

- Les foncteurs d'oubli "oublient" certaines propriétés des objets d'une catégorie. Par exemple, dans la catégorie des groupes, les objets sont des groupes donc des monoïdes munis d'un axiome d'inversibilité, et les morphismes sont (en particulier) des morphismes de monoïdes. Donc on peut "oublier" la propriété d'inverse, c'est le foncteur d'oubli de la catégorie **Grp** dans *Mon* qui envoie tout groupe sur son monoïde sous jacent, et tout morphisme de groupe sur lui même (si la compatibilité de l'inversion avait fait partie de la définition de morphisme, elle aurait été *oubliée*, mais ce n'est qu'une propriété qui découle des axiomes du groupe. Donc la définition d'un morphisme de groupe est la même que celle d'un morphisme de monoïde). On peut faire la même chose avec toute structure algébrique "strictement plus faible" qu'un groupe : on peut oublier toutes les propriétés d'un groupe de façon fonctorielle, jusqu'à envoyer **Grp** sur **Set**, en ne gardant que l'information d'ensemble (ensemble sous jacent du groupe), et d'application (contenue dans l'information du morphisme de groupes).

1.3 Produits, coproduits

Définir formellement ici la notion de produit et coproduit ne serait pas forcément pertinent. Comme dit plus tôt, l'objectif de la section est de donner uniquement les éléments catégoriques nécessaires à la compréhension du document. Aussi, nous ne définirons pas ici la notion de produit/coproduit dans une catégorie. Une définition formelle et abordable se trouve dans le **MacLane** (page 62), une ressource régulièrement recommandée pour s'introduire à la théorie des catégories. Mais pour lire ce document, il suffit de savoir que sur la catégorie des ensembles **Set**, le coproduit (noté $+$) est l'union disjointe entre deux ensembles. Notons alors que dans une union disjointe $A + B$ on peut différencier un élément de A d'un élément de B : ça nous permet entre autres de définir des application avec un "pattern matching", qui se comportent différemment sur la première et la seconde composante de du coproduit. Par exemple, on peut définir le morphisme de $\mathbb{N} + \mathbb{Z}$ qui à un entier naturel associe son successeur, et à un entier associe son prédécesseur :

$$\begin{aligned}\alpha : \mathbb{N} + \mathbb{Z} &\rightarrow \mathbb{Z} \\ n &\mapsto n + 1 \\ k &\mapsto k - 1\end{aligned}$$

Les donnée d'un élément de \mathbb{Z} et d'un ensemble X nous donnent la donnée d'un élément de $\mathbb{Z} + X$, donc si on veut avoir pour codomaine un coproduit, on peut "injecter" l'élément d'un ensemble A dans le coproduit $A + B$. Concrètement, un élément $a \in A$ s'injecte dans $A + B$ comme $(0, a)$. Pour réaliser cette injection, on peut utiliser *inr*. L'équivalent pour l'injection à gauche est *inl*. Si on veut définir α comme un endomorphisme de $\mathbb{N} + \mathbb{Z}$, on peut donc le réécrire en utilisant ces

injections :

$$\begin{aligned}\alpha : \mathbb{N} + \mathbb{Z} &\rightarrow \mathbb{N} + \mathbb{Z} \\ n &\mapsto \text{inl}(n + 1) \\ k &\mapsto \text{inr}(k - 1)\end{aligned}$$

Dans **Set** le coproduit, noté \times est simplement el produit cartésien.

2 F-algèbres

2.1 Définition et exemples

Soit C une catégorie, et $F : C \rightarrow C$ un endofoncteur. Une F -algèbre est la donnée d'un couple (X, f) tels que $X \in \text{ob}(C)$ et $f \in \text{Hom}(FX, X)$. Les F -algèbres forment une catégorie, ce qui s'exprime par la commutativité de ce diagramme (pour X et Y des objets de C , (X, f) et (Y, g) des F -algèbres) :

$$\begin{array}{ccc} FX & \xrightarrow{f} & X \\ Fh \downarrow & & \downarrow h \\ FY & \xrightarrow{g} & Y \end{array}$$

2.1.1 Exemples (et non exemple) algébriques

Les F -algèbres fournissent une définition (un peu plus) constructive des structures algébriques usuelles. Par exemple, d'un point de vue catégorique, on pourrait dire qu'un monoïde est simplement un objet de la catégorie monoïdes. Cette définition ne permet pas d'embarquer (d'un point de vue catégoriel) ce qu'il se passe *à l'intérieur* d'un monoïde : rien n'explicite la façon dont se comporte la loi du monoïde. Les monoïdes induisent une F -algèbre sur la catégorie des ensembles. En effet, soit M un monoïde, et M_{Ens} son ensemble sous jacent.

$$\begin{aligned}F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\mapsto 1 + X \times X\end{aligned}$$

Est bien un endofoncteur de **Set**, et M est pleinement défini par la F -algèbre (M_{Ens}, α) , où

$$\begin{aligned}\alpha : FM_{\text{Ens}} &\rightarrow M_{\text{Ens}} \\ * &\mapsto 1_M \\ \langle x, y \rangle &\mapsto xy\end{aligned}$$

Un monoïde induit donc une F -algèbre pour ce foncteur, mais l'inverse n'est pas vrai : on peut par exemple considérer pour un ensemble E qui contient un

élément 1 :

$$\begin{aligned} g : FE &\rightarrow M \\ * &\mapsto 1 \\ \langle x, y \rangle &\mapsto 1 \end{aligned}$$

qui ne définit aucun autre monoïde que le monoïde trivial $\{1_M\}$. Il faut aussi bien prendre en compte qu'il s'agit d'une F -algèbre, et que F embarque avec lui sa source, c'est à dire la catégorie des ensembles.

Des exemples analogues existent pour plusieurs structures algébriques, par exemple : Soit G un groupe d'ensemble sous-jacent G_{Ens} .

$$\begin{aligned} F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\rightarrow 1 + X + X \times X \end{aligned}$$

Est bien un endofoncteur. La F -algèbre (G_{Ens}, α) définit notre groupe G :

$$\begin{aligned} \alpha : FG_{Ens} &\rightarrow G_{Ens} \\ * &\mapsto 1_G \\ x &\mapsto x^{-1} \\ \langle x, y \rangle &\mapsto x \cdot y \end{aligned}$$

Soit A un anneau d'ensemble sous-jacent A_{Ens} .

$$\begin{aligned} F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\rightarrow 1 + 1 + X + X \times X + X \times X \end{aligned}$$

Est bien un endofoncteur de \mathbf{Set} . A est défini par la F -algèbre (A_{Ens}, α) où

$$\begin{aligned} \alpha : FA &\rightarrow A \\ *_1 &\mapsto 1_A \\ *_2 &\mapsto 0_A \\ x &\mapsto -x \\ \langle x, y \rangle_1 &\mapsto xy \\ \langle x, y \rangle_2 &\mapsto x + y \end{aligned}$$

Ici on peut différencier $*_1$ et $*_2$: même si 1 et 1 sont les mêmes ensembles, $1 + 1 = \{(0, *), (1, *)\}$.

Soit K un corps d'ensemble sous-jacent K_{Ens} , d'ensemble sous-jacent privé du neutre $K_{Ens} - \{0_K\}$.

$$\begin{aligned} F_K : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\rightarrow 1 + 1 + X + (X - \{0_K\}) + X \times X + X \times X \end{aligned}$$

est bien un endofoncteur de **Set**. K est donc bien défini par la F_K -algèbre (K, α) où

$$\begin{aligned}\alpha F_K K &\rightarrow K \\ *_1 &\mapsto 1_K \\ *_2 &\mapsto 0_K \\ x_1 &\mapsto -x \\ x_2 &\mapsto x^{-1} \\ \langle x, y \rangle_1 &\mapsto x + y \\ \langle x, y \rangle_2 &\mapsto xy\end{aligned}$$

On remarque cependant que le foncteur F tel qu'il est défini dépend de K : il doit inclure la donnée de son neutre additif. La définition est donc moins forte : les corps ne sont pas la donnée d'une F -algèbre, ils sont la donnée d'un foncteur F_K et d'une F_K -algèbre. La faiblesse de ce modélisation illustre un fait sur l'interprétation catégorique des corps : les propriétés ne sont pas toutes universelles (en particulier, tous les éléments n'admettent pas d'inverse multiplicatif), ce qui donne un comportement assez pauvre d'un point de vue catégorique.

2.1.2 Types inductifs

Un outil puissant pour décrire les types inductifs est la forme de Backus-Naur. Dans cette section, on s'intéresse à des types représentés par un genre de forme de Backus-Naur. Concrètement, une forme de Backus-Naur :

$$t := B_1 | \dots | B_k | C_1 \text{ of } P_1(T) | \dots | C_n \text{ of } P_n(T)$$

veut dire "un élément du type t est soit une constante B_i , soit une construction d'un constructeur C_j sur un argument qui est un "polynôme" de T , c'est à dire une expression formée de produits et coproduits de T , et de types constants. Cette définition inductive nous donne immédiatement un foncteur (sous l'hypothèse qu'on peut modéliser tous les types en jeu par des objets de **Set**).

$$F : \mathbf{Set} \rightarrow \mathbf{Set}$$

$$X \mapsto \sum_{i=1}^k 1 + \sum_{j=1}^n C_j P_j(X)$$

Est bien un endofoncteur de **Set**. Alors, si T décrit l'ensemble des éléments de notre type, le type inductif et ses constructeurs sont décrits par (T, α) où

$$\begin{aligned}\alpha : FT &\rightarrow T \\ *_i &\mapsto B_i \\ P_j(T) &\mapsto C_j P(T)\end{aligned}$$

C'est une définition abstraite et généralisée pour les types inductifs : voyons des exemples d'application de cette dernière.

On considère la définition usuelle du type d'une liste d'entiers naturels donnée par

$$l, l' := nil \mid (n : \mathbb{N}) :: l'$$

Le foncteur associé est donc

$$\begin{aligned} F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\mapsto 1 + \mathbb{N} \times X \end{aligned}$$

et la F -algèbre associée, pour l'ensemble L des listes d'entiers naturels, est (L, α) , où

$$\begin{aligned} \alpha : FL &\rightarrow L \\ 1 &\mapsto nil \\ \langle n, l \rangle &\mapsto n :: l \end{aligned}$$

Cet exemple se restreint aux listes d'entiers naturels, mais on peut généraliser pour obtenir la F -algèbre associée au type inductif des listes d'éléments de A , on pose alors $F : X \mapsto 1 + A \times X$: on fait varier la partie "constante" du polynôme en X .

On considère la définition usuelle du type "arbres d'entiers naturels" donnée par

$$t := F \text{ of } \mathbb{N} \mid A \text{ of } \mathbb{N} \times t \times t$$

Alors,

$$\begin{aligned} F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\mapsto \mathbb{N} + \mathbb{N} \times (X \times X) \end{aligned}$$

est bien un endofoncteur de \mathbf{Set} . Soit T le type des arbres d'entiers, alors T (et ses constructeurs) est défini par la F -algèbre (T, α) où

$$\begin{aligned} \alpha : FT &\rightarrow T \\ n &\mapsto F(n) \\ \langle n, \langle sag, sad \rangle \rangle &\mapsto A(n, sad, sag) \end{aligned}$$

Ces deux exemples sont élémentaires, mais modéliser un type inductif comme une F -algèbre est une méthode courante pour pouvoir l'étudier d'un point de vue théorique, et notamment pour la formalisation sur ordinateur fondée sur une théorie des catégories d'ordre supérieur.

2.2 Initialité dans les F -algèbres

Comme dit précédemment, pour un foncteur F , les F -algèbres forment une catégorie. L'existence d'un objet initial dans ces catégories nous donne divers éléments utiles, en voici deux.

2.2.1 Récurrence

Soit F le foncteur :

$$\begin{aligned} F : C &\rightarrow C \\ X &\mapsto 1 + X \end{aligned}$$

On considère C une catégorie qui modélise l'objet qu'on veut construire par récurrence (encore une fois : on se ramène au cas $C = \mathbf{Set}$), et A l'objet qui décrit ce qu'on veut construire par récurrence (il peut s'agir d'une suite ou d'une propriété qu'on montre par récurrence, par exemple). Alors, notre construction par induction est exactement la donnée d'un élément 0_A de A , et d'un morphisme de A dans A qui sert à obtenir un "successeur" pour notre définition. Un élément 0_A est exactement la donnée d'un morphisme

$$\begin{aligned} f : 1 &\rightarrow A \\ * &\mapsto 0_A \end{aligned}$$

Cet objet 0_A est, par exemple, un prédicat $P(0)$ dans le cas d'une preuve par récurrence, et un premier terme u_0 dans le cas d'une construction par récurrence d'une suite, c'est une généralisation du "premier élément" d'une définition par récurrence. Le "successeur" de la définition peut se définir comme un morphisme

$$\begin{aligned} g : A &\rightarrow A \\ x &\mapsto g(x) \end{aligned}$$

À titre d'exemple, dans le cas d'une preuve par récurrence, g associe $P(n+1)$ à $P(n)$. Une construction par récurrence se décrit donc comme F -algèbre (A, α) où

$$\begin{aligned} \alpha : 1 + A &\rightarrow A \\ * &\mapsto 0_A \\ x &\mapsto g(x) \end{aligned}$$

Or, la catégorie des F -algèbres admet un objet initial : (β, \mathbb{N}) où :

$$\begin{aligned} \beta : 1 + \mathbb{N} &\rightarrow \mathbb{N} \\ * &\mapsto 0 \\ n &\mapsto Succ(n) \end{aligned}$$

Prouvons son initialité. Soit (A, α) une F -algèbre qui décrit une construction par récurrence, où

$$\begin{aligned} \alpha : 1 + A &\rightarrow A \\ * &\mapsto 0_A \\ x &\mapsto g(x) \end{aligned}$$

On introduit

$$\begin{aligned}\eta : \mathbb{N} &\rightarrow A \\ n &\mapsto \alpha^{n+1}(*)\end{aligned}$$

Montrons qu'il s'agit bien d'un morphisme de F -algèbre. Il suffit de montrer la commutativité du diagramme suivant pour η :

$$\begin{array}{ccc} 1 + \mathbb{N} & \xrightarrow{\beta} & \mathbb{N} \\ \downarrow F\eta & & \downarrow \eta \\ 1 + A & \xrightarrow{\alpha} & A \end{array}$$

Sachant que :

$$\begin{aligned}F\eta : 1 + \mathbb{N} &\rightarrow 1 + A \\ * &\mapsto \text{inl}(*) \\ x &\mapsto \text{inr}(\eta(x))\end{aligned}$$

Soit $x : 1 + \mathbb{N}$. Si $x = * \in 1$, $\eta(\beta(x)) = \eta(\beta(*)) = \eta(0) = \alpha(*) = 0_A$ et $\alpha(F\eta(x)) = \alpha(F\eta(*)) = \alpha(\text{inl}(*)) = 0_A$. Si $x = n \in \mathbb{N}$, $\eta(\beta(x)) = \eta(\beta(n)) = \eta(\text{succ}(n)) = \alpha^{n+2}(*)$ et $\alpha(F\eta(x)) = \alpha(F\eta(n)) = \alpha(\text{inr}(\eta(n))) = \alpha(\alpha^{n+1}(*)) = \alpha^{n+2}(*)$. Ce morphisme est donc bien un morphisme d' F -algèbre. Vérifier l'unicité n'est pas plus difficile : on prend un morphisme η' qui fait commuter le diagramme précédent, et on a forcément $\eta'(n+1) = \eta'^n(F\eta'(*))$. On a établi qu'une construction par récurrence sur des objets de A était une F -algèbre avec A pour support, or on a une flèche de (\mathbb{N}, η) vers la F -algèbre qui définit la construction par récurrence, donc on a simplement à donner un élément de $F\mathbb{N}$ pour obtenir l'élément souhaité de FA : l'élément de base 0_A , ou bien la n -ième itération.

2.2.2 Fold

Le fold est une opération très classique en informatique. Il s'agit de *plier* une liste sur elle même, pour la compacter en un élément. La façon de *plier* un élément avec un autre est donnée par une fonction, et le pli d'un élément sur lui même est donné par un élément de base. En Ocaml, le fold correspond à cette opération :

```
('a -> 'b -> 'b) -> 'b -> 'a list -> 'b
```

Une fonction qui prend un élément de type A , et s'évalue en fonction de B dans B , une liste d'éléments de type A , un élément "base" de type B , pour donner un type B . En voici le code :

```
let rec fold (f : 'a -> 'b -> 'b) (b : 'b) (l : 'a list)
: 'b =
```

```

match l with
| [] -> b
| x :: xs -> f x (fold f xs b)
;;

```

Et quelques exemples d'application directe :

```

      fold ( + ) (1 :: 2 :: 3 :: []) 0 ;;
- : int = 6
      fold ( ^ ) ("Qui-" :: "est-" :: "John-" :: "Galt" :: []) "?" ;;
- : string = "Qui-est-John-Galt?"
      fold (fun x -> (fun y -> string_of_int(x) ^ "-,-" ^ y))
(1 :: 2 :: 3 :: []) "et-c'est-fini" ;;
- : string = "1,-,-2,-,-3,-,-et-c'est-fini"

```

Prenons une catégorie (en pratique, **Set**) qui modélise nos élément de type A , B , et L_A les listes d'éléments de A . Un fold peut donc être vu comme une suite de morphismes (au sens de **Set**, donc une suite d'applications) :

$$(A \rightarrow B \rightarrow B) \rightarrow B \rightarrow L_A \rightarrow B$$

On utilise alors un certain nombre d'isomorphismes pour transformer cette définition :

$$\begin{aligned} (A \rightarrow B \rightarrow B) \rightarrow B \rightarrow L_A \rightarrow B &\cong \\ (A \rightarrow B \rightarrow B) \rightarrow (1 \rightarrow B) \rightarrow L_A \rightarrow B \end{aligned}$$

Un élément de B est exactement la donnée d'un morphisme $1 \rightarrow B$.

$$\begin{aligned} (A \rightarrow B \rightarrow B) \rightarrow (1 \rightarrow B) \rightarrow L_A \rightarrow B &\cong \\ (A \times B \rightarrow B) \rightarrow (1 \rightarrow B) \rightarrow L_A \rightarrow B \end{aligned}$$

Un morphisme $A \rightarrow B \rightarrow B$ donne les mêmes informations qu'un morphisme $A \times B \rightarrow B$: c'est la Curryfication. Dans l'idée ça revient à dire qu'une fonction f d'un couple x, y qui donne un résultat z est une fonction d'un élément x qui donne une fonction $y \rightarrow z$.

$$\begin{aligned} (A \times B \rightarrow B) \rightarrow (1 \rightarrow B) \rightarrow L_A \rightarrow B &\cong \\ (1 + A \times B) \rightarrow B \rightarrow L_A \rightarrow B \end{aligned}$$

Un morphisme $X \rightarrow Y$ et un morphisme $Z \rightarrow Y$ nous donne un morphisme $X + Z \rightarrow Y$ dans **Set** : il suffit de construire le morphisme par disjonction de cas, comme un filtrage de motif.

Un fold est donc la donnée de cette suite de morphisme.

$$(1 + A \times B \rightarrow B) \rightarrow L_A \rightarrow B$$

On remarque que le premier élément peut s'exprimer comme une F -algèbre. En effet, on pose :

$$\begin{aligned} F : C &\rightarrow C \\ X &\mapsto 1 + A \times X \end{aligned}$$

Ce morphisme est donc un morphisme $FB \rightarrow B$. Or, cette F -algèbre admet une algèbre initiale. En effet, on pose L les listes d'éléments de A et

$$\begin{aligned} \alpha : 1 + A \times L_A &\rightarrow L_A \\ * &\mapsto \text{liste vide} \\ \langle x, l \rangle &\mapsto l \text{ à laquelle on a ajouté } x \end{aligned}$$

Les listes d'éléments de A forment une algèbre initiale (on peut le vérifier exactement de la même manière que dans la partie sur la récursion). Par conséquent, en composant notre fold et le morphisme de l'objet initial, notre fold est exactement la donnée d'un morphisme $(FL_A \rightarrow L_A) \rightarrow L_A \rightarrow B$, or $FL_A \rightarrow L_A$ est déjà donné par α . Donc notre fold est équivalent à un morphisme $L_A \rightarrow B$ grâce à l'initialité des listes d'éléments de A .

3 F-coalgèbres

3.1 Définition et exemples

Une F -coalgèbre est l'objet *dual* d'une F -algèbre. Cette notion abstraite peut se simplifier : une F -coalgèbre fait commuter le diagramme d'une F -algèbre dont on a "retourné" les flèches.

$$\begin{array}{ccc} X & \xrightarrow{f} & FX \\ Fh \downarrow & & \downarrow h \\ Y & \xrightarrow{g} & FY \end{array}$$

Pour un endofoncteur F défini sur une catégorie C , une F -coalgèbre est donc la donnée d'un support A (élément de C) et d'un morphisme $A \rightarrow FA$. Cette définition peut paraître troublante une fois qu'on a pleinement compris celle d'une F -algèbre : la F algèbre nous permettait d'exprimer une construction, via un morphisme. La coalgèbre, au contraire, permet d'exprimer la "destruction". C'est un formalisme particulièrement approprié pour décrire des structures potentiellement infinies, ou des processus qui *pourraient* ne pas s'arrêter.

3.1.1 F-algèbres initiales

Une F -algèbre initiale induit une F -coalgèbre. En effet, soit C une catégorie, F un endofoncteur sur C , et A, α une F -algèbre initiale. Alors, pour toute F -algèbre supportée par X , il existe une unique flèche α' de A vers X . On pose

$X = FA$, on a donc un unique morphisme :

$$\alpha' : A \rightarrow FA$$

C'est à dire que (A, α') constitue une F -coalgèbre. En général, α' est d'ailleurs déjà explicité à ce stade : lorsqu'on montre que A est support d'une F -algèbre initiale, on construit l'unique morphisme pour un objet quelconque, il suffit de remplacer cet objet par A .

3.1.2 Un exemple algébrique : les groupes quotient

Un groupe quotient reste un groupe, on peut donc exprimer cette transformation par un endofoncteur de **Set**. Le premier exemple : le quotient de G par G . Le foncteur F associé envoie alors un groupe G sur le groupe G/G isomorphe au groupe trivial, et tout morphisme sur le morphisme trivial.

Deuxième exemple : le quotient de G par $\{1_G\}$. Le foncteur F associé envoie alors G sur $G/\{1_G\}$ isomorphe à G (foncteur identité), et tout morphisme sur lui-même.

Troisième exemple un peu moins simple : le quotient de G par $D(G)$ le sous groupe dérivé de G . Le foncteur envoie un objet G sur $G/D(G)$, et le morphisme sur le morphisme associé, "abélianisé".

Dans tous ces exemples, le quotient par un groupe induit une F -algèbre. Soit F le "foncteur quotient", G un groupe, alors (G, π) est une F -algèbre si π est la projection canonique surjective. On a :

$$\begin{aligned} \pi : G &\rightarrow FG \\ g &\mapsto \bar{g} \end{aligned}$$

Où \bar{g} est la classe de g suivant le groupe par lequel on quotiente.

3.1.3 Types coinductifs

Les F -coalgèbres modélisent souvent un type coinductif. L'intuition d'un type coinductif est la suivante : au lieu de lire une BNF de façon "inductive", i.e avec des cas de base, des constructeurs, on l'interprète de façon coinductive : avec des destructeurs.

Pour modéliser le "destructeur" (prédécesseur) du type des entiers naturels, on peut poser :

$$\begin{aligned} F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\mapsto 1 + X \end{aligned}$$

La F -coalgèbre qui correspond à la destruction des entiers naturels (poten-

tiellement infinis) est alors $(\bar{\mathbb{N}}, \alpha)$:

$$\begin{aligned}\alpha : \bar{\mathbb{N}} &\rightarrow 1 + \bar{\mathbb{N}} \\ 0 &\mapsto * \\ n + 1 &\mapsto n \\ \infty &\mapsto \infty\end{aligned}$$

En fait, cette coalgèbre est terminale pour ce foncteur : exactement comme pour le cas inductif, cette coalgèbre nous fournit toutes les constructions par coinduction, y compris des raisonnements coinductifs

Un autre exemple est donné par les arbres binaires coinductifs dont les noeuds sont de type A . On considère

$$\begin{aligned}F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\mapsto A \times X \times X\end{aligned}$$

Alors, F est un endofoncteur \mathbf{Set} . En posant S l'ensemble des arbres coinductifs, le type des arbres coinductifs et ses constructeurs est décrit par la F -coalgèbre suivante (S, α) :

$$\begin{aligned}\alpha : S &\rightarrow FS \\ t &\rightarrow (n, \text{sag}, \text{sad})\end{aligned}$$

Où n est la valeur portée par le noeud de t , sag et sad ses sous arbres droits et gauches. Définir un arbre coinductif peut permettre d'exprimer beaucoup de propriétés "rationnelles", c'est à dire des propriétés infinies mais dont l'infini se comporte de façon modélisée par un arbre.

Pour modéliser un flux de donnée continu (stream), on peut utiliser une F -coalgèbre. Pour A un alphabet, on pose

$$\begin{aligned}F : \mathbf{Set} &\rightarrow \mathbf{Set} \\ X &\mapsto 1 + A \times X\end{aligned}$$

On modélise l'ensemble de nos états par X , et l'état suivant obtenu depuis un état x par $\langle x', t \rangle$ où t est un symbole de l'alphabet. Alors, la F -coalgèbre des flux de données contenant des états stockés dans X est donnée par (X, α) :

$$\begin{aligned}\alpha : X &\rightarrow FX \\ a &\mapsto \text{inl}(*) \\ x &\mapsto \text{inr}(\langle x', t \rangle)\end{aligned}$$

Cette définition laisse penser qu'on pourrait construire des automates infinis via des F -coalgèbres, et c'est vrai (cette construction est présentée dans le **Jacobs**

On peut également modéliser un grand nombre de structures de données "infinies" : notamment des arbres, des λ -calculs. La notion de F -coalgèbre permet donc de modéliser des choses qui *pourraient* ne pas s'arrêter.

4 Bibliographie

Références

- [1] Mac Lane *Categories for the working Mathematician*
- [2] Jacobs <https://www.cs.ru.nl/B.Jacobs/CLG/JacobsCoalgebraIntro.pdf>