F-algèbres et F-coalgèbres

April 2024

Contents

1 Introduction au langage catégorique

La théorie des catégories est un sujet extrêmement 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*. C'est extrêmement large, c'est le but, 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 $mor(C)$ 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 $\forall X \in Ob(C)$, il existe un unique morphisme de I dans X. Un objet final F d'une catégorie C est un objet tel que $\forall X \in Ob(C)$, 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.
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, syntaxtiquement 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 $X \to Y$ et $Y \to Z$ se compose en $X \to Z$. 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 une transformation des objets et des morphismes de C en objets et morphismes de D, en envoyant les identités de C sur celles de D, et en respectant la composition : pour tout couple α, β de morphismes de C, $(\alpha : X \to Y,\ \beta : Y \to Z)$, $F(\beta \circ \alpha) = F(\beta) \circ F(\alpha) : F(X) \to F(Z)$. Voici des exemples de foncteurs courants :

- Pour toute catégorie C, un foncteur identité de C dans C est défini. La composition des foncteurs se faisant "correctement", on peut définir catégorie de foncteurs de C dans D : les morphismes sont appelés des transformations naturelles.

- Le foncteur constant envoie tous les objets de la catégorie de départ sur le même objet de la catégorie d'arrivée et qui envoie 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 permettent d'"oublier" 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. 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 fonctonrielle, jusqu'à envoyer Grp sur Set, en ne gardant que la structure d'ensemble (ensemble sous

jacent du groupe), et d'application (contenue dans la structure de 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, je pense que la meilleure façon d'apprendre la théorie des catégories est de se confronter à des exemples, et d'aller chercher la partie théorique nécessaire à leur compréhension. Une définition formelle et abordable le **MacLane** (page 62), une ressource régulièrement recommandée pour s'introduire à la théorie des catégories. Mais pour lire ce document, il suffit de pouvoir se représenter un coproduit (noté $+$) comme une extension l'union disjointe entre deux objets. Notons alors que $A + B$ est une forme d'union disjointe, donc on peut y différencier un objet de A d'un objet de B : ça nous permet entre autres de définir des morphismes avec un "pattern matching", qui se comprortent 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 :

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

Où inl et inr désignent les injections droites et gauche : $n + 1$ est un objet de $\mathbb{N}$, mais pour qu'il soit interprété comme un objet de $\mathbb{N}+\mathbb{Z}$ il faut l'y "injecter à gauche". Idem pour l'injection droite. Un coproduit, noté $\times$ peut être interprété comme un produit cartésien : ses éléments sont donc des couples.

2 F-algèbres

2.1 Définition et exemples

Soit C une catégorie, et $F : C \longrightarrow C$ un endofoncteur. Une F-algèbre est la donnée d'un couple (X, f) tels que $X \in ob(C)$ et $f \in Hom(FX, X)$. 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) :

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

2.1.1 Exemples (et non exemples) 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 monoïdes est simplement un objet de la catégorie monoïdes. Cette définition ne permet pas de regarder ce qu'il se passe *à l'intérieur* d'un monoïde : par exemple, un monoïde est par définition doté d'un neutre, mais cette définition ne permet pas de l'expliciter (en fait, on doit définir un morphisme $1 \to M$ qui envoie $*$ sur le neutre, mais cette définition n'est pas "canonique". On n'a pas un unique morphisme $1 \to M$ pour tous les monoïdes, donc décrire un monoïde simplement comme l'élément d'une catégorie ne permet pas d'"embarquer" son neutre explicite). Rien n'explicite non plus la façon dont se comporte la loi du monoïde. Les monoïdes induisent une F-algèbre sur la catégorie des monoïdes. En effet, on pose

$$\begin{aligned} F : M &\to M \\ M &\mapsto 1 + M \times M \end{aligned}$$

Et pour un monoïde donné M de neutre 1_M et de loi $\cdot$, on peut donc définir notre F-algèbre :

$$\begin{aligned} f : FM &\to M \\ * &\mapsto 1_M \\ \langle x, y \rangle &\mapsto x \cdot y \end{aligned}$$

permet d'expliciter notre loi, et notre neutre. Un monoïde induit donc une F-algèbre pour ce foncteur, mais l'inverse n'est pas vrai : on peut par exemple considérer

$$\begin{aligned} g : FM &\to M \\ * &\mapsto 1_M \\ \langle x, y \rangle &\mapsto 1_M \end{aligned}$$

qui ne définit aucun autre monoïde que le monoïde libre sur $\{1_M\}$. 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 monoïdes. Avec le même foncteur, sur la catégorie des magmas unifères, notre F-algèbre est induite par un magma, et non un monoïde. Tout est donc important : la source du foncteur, et le morphisme.

Des exemples analogues existent pour nombre de structures algébriques, par exemple :
Pour les groupes : on pose

$$\begin{aligned} F : &Grp \to Grp \\ &G \to 1 + G + G \times G \end{aligned}$$

Pour un groupe G, on pose

$$\begin{aligned} f : FG &\to G \\ * &\mapsto 1_G \\ x &\mapsto x^{-1} \\ \langle x, y \rangle &\mapsto x \cdot y \end{aligned}$$

Pour les anneaux (unitaires) : on pose

$$\begin{aligned} F : CRing &\to CRing \\ A &\to 1 + 1 + A + A \times A + A \times A \end{aligned}$$

Pour un anneau unitaire A, on pose

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

Pour les corps, on aurait envie de faire la même chose, en posant par exemple

$$\begin{aligned} F : CField &\to CField \\ K &\to 1 + 1 + K + K^* + K \times K + K \times K \end{aligned}$$

Mais c'est le K^* qui pose problème : ça ne s'écrit pas vraiment en théorie des catégories (d'ailleurs, ça ne s'écrit pas vraiment en théorie des types non plus). C'est un exemple de limitation de pouvoir expressif des F-algèbres (et de la théorie des catégories au sens général). En fait, la théorie des catégories permet de décrire très efficacement l'universalité, c'est à dire des propriétés qui peuvent se réécrire avec un $\forall$. Mais l'inversion multiplicative n'est pas de ces propriétés. Le corps est donc un *non-exemple* de structure algébrique descriptible par une F-algèbre.

2.1.2 Types inductifs

Les types inductifs sont très souvent représentés comme des points fixes de foncteurs bien choisis. Ce foncteur est un "collage" de la définition inductive du type, qui s'écrit dans un formalisme propice (en général une forme de Backus-Naur). Concrètement, une définition (de style "ocaml") d'un type inductif :

$$t := B_1|...|B_k|C_1 of A_1|...|C_n of A_n$$

veut dire "un élément du type t est soit une constante B_i, soit une construction d'un constructeur C_j sur un argument de type A_j". Cette définition inductive nous donne immédiatement un foncteur : il suffit que les type des "cas de base" et ceux des arguments puissent être représentés dans une catégorie (souvent, on a besoin qu'elle soit munie de coproduit et de produit pour des arguments qui seraient des n-uplets et des types non triviaux. Cependant, on se ramène très souvent à Set, donc on dispose de tous les outils catégoriques nécessaires). En appelant cette catégorie C, le foncteur associé est alors un endofoncteur de C :

$$F : X \mapsto \sum_{i=1}^{k} 1 + \sum_{j=1}^{n} A_j$$

Le type inductif peut alors être vu comme un point fixe de ce foncteur. Encore une fois, définir notre type inductif comme une F-algèbre fournit une définition plus constructive : le morphisme n'indique pas seulement que le type inductif est un point fixe de ce foncteur, mais aussi la façon dont l'application du foncteur au type se comporte : à quoi ressemble l'image d'un élément de C (c'est à dire, un élément du type) par le foncteur ? Cette information est donnée par les constructeurs de notre type inductif, qui seront les parties du morphisme associé à la F-algèbre suivante : on pose

$$F : X \mapsto \sum_{i=1}^{k} 1 + \sum_{j=1}^{n} A_j$$

et T son "plus petit" (c'est la notion que je ne définis pas vraiment ici, mais on peut se ramener au cas où la catégorie modélisant nos types est Set, donc on a un ordre partiel sur cette catégorie, donné par l'inclusion) point fixe. Alors, notre type inductif est T, et ses constructeurs sont explicités par la F-algèbre suivante :

$$\begin{aligned} \alpha :& FT \to T \\ *_i &\mapsto B_i \\ A_j &\mapsto C_j A_j \end{aligned}$$

Par exemple, en considérant la définition usuelle du type d'une liste d'entiers naturels : le type est donné par

$$l, l' := nil|(n : Nat) :: l'$$

Le foncteur associé est donc

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

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

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

Ici on a traité des listes d'entiers naturels, mais on peut très bien le faire en changeant $\mathbb{N}$ en A, où A décrit les éléments potentiels de notre liste. De la même façon, on peut définir un arbre d'entiers naturels comme une F-algèbre. Le type des arbres d'entiers naturels est donné par

$$t := Feuilleof\mathbb{N}|Aof(\mathbb{N}, t, t)$$

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

T l'ensemble des éléments qui peuvent figurer dans notre arbre, et α :

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

Ces deux exemples sont très basiques, mais presque toutes les structures de données représentables comme un type inductif peuvent être décrites par des F-algèbres.

2.2 Initialité dans les F-algèbres

Comem 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 suivant :

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

On considère C une catégorie qui modélise l'objet qu'on veut construire par récurrence (très souvent munie d'un coproduit), 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 est exactement la donnée d'un morphisme

il s'agit d'une fonction d'un singleton dans A qui associe à l'unique élément de 1 l'objet 0 voulu. Cet objet 0_A est 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 notre définition peut se définir comme un morphisme

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

Dans le cas d'une preuve par récurrence, g associe $P(n+1)$ à $P(n)$, par exemple.

Une construction par récurrence se pose donc comme une F-algèbre sur A :

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

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

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

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

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

On pose le morphisme

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

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

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

Sachant que :

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

Soit $x : 1+\mathbb{N}$. Si $x = * \in 1$, $\eta(\beta(x)) = \eta(\beta(*)) = \eta(0) = \alpha(*) = 0_A$ et $\alpha(F\eta(x)) = \alpha(F\eta(*)) = \alpha(inl(*)) = O_A$. Si $x = n \in \mathbb{N}$, $\eta(\beta(x)) = \eta(\beta(n)) = \eta(succ(n)) = \alpha^{n+2}(*)$ et $\alpha(F\eta(x)) = \alpha(F\eta(n)) = \alpha(inr(\eta(n)) = \alpha(\alpha^{n+1}(*)) = \alpha^{n+2}(*)$. 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 $(\mathbb{N}, \eta)$ vers la F-algèbre qui définit la construction par récurrence, donc on a simplement à fournir un élément de $F\mathbb{N}$ pour obtenir l'élément souhaité de A.

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'aucun élément est donné par un élément de base. En ocaml, le fold correspond à cette opération :

```
('a -> 'b -> 'b) -> 'a list -> 'b -> '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) (l : 'a list) (b : 'b) : 'b =
    match l with
    | [] -> b
    | x :: xs -> f x (fold f xs b)
    ;;

    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 qui modélise nos élément de type A , B, et L_A les listes d'éléments de A. On suppose cette catégorie munie de coproduits et de produits (ce n'est pas une supposition très forte, on peut tout à fait prendre Set pour modéliser nos éléments). Un fold peut donc être vu comme une suite de morphismes :

$$(A \to B \to B) \to L_A \to B \to B$$

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

$$(A \to B \to B) \to B \to L_A \to B \cong$$
$$(\mathrm{A} \to B \to B) \to (1 \to B) \to L_A \to B$$

(un élément de B est exactement la donnée d'un morphisme $1 \to B$)

$$(A \to B \to B) \to (1 \to B) \to L_A \to B \cong$$
$$(\mathrm{A} \times B \to B) \to (1 \to B) \to L_A \to B)$$

Un morphisme $A \to B \to B$ donne les mêmes informations qu'un morphisme $A \times B \to B$: c'est la Curryfication.

$$(A \times B \to B) \to (1 \to B) \to L_A \to B) \cong$$
$$(1 + \mathrm{A} \times B) \to B) \to L_A \to B$$

Un morphisme $X \to Y$ et un morphisme $Z \to Y$ nous donne un morphisme $X + Z \to Y$ dans une catégorie munie de coproduits : 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.

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

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

$$F : C \to C$$
$$X \mapsto 1 + A \times X$$

Or, cette F-algèbre admet une algèbre initiale. En effet, on pose L les listes d'éléments de A et

$$\alpha : 1 + A \times L \to L$$
$$* \mapsto \text{liste vide}$$
$$\langle x, l \rangle \mapsto \text{l à laquelle on a ajouté x}$$

(on peut vérifier l'initialité de l'algèbre exactement de la même façon que dans la partie sur l'induction). Les listes d'éléments de A forment une algèbre initiale. Par conséquent, en composant notre fold et le morphisme de l'objet initial, notre fold est exactement la donnée d'un morphisme $(FL \to L) \to L_A \to B$, or $FL \to L$ est déjà donné par α. Donc notre fold est équivalent à un morphisme $L_A \to B$ 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-algèbre fait commuter un diagramme dont on a retourné les flèches.

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

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 $A \to FA$. 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 infinies. On se limitera à l'étude de ces modélisations.

3.2 Types coinductifs

Les F-coalgèbres modélisent le plus 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" (prédecesseur) du type des entiers naturels, on peut poser :

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

La F-coalgèbre qui correspond à la destruction des entiers naturels (potentiellement infinis) est alors $(\bar{\mathbb{N}}, \alpha)$:

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

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

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

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

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) :

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

Cette définition laisse penser qu'on pourrait construire des automates infinis via des F-coalgèbres. C'est une juste intuition, développée à travers une très large théorie pour laquelle **Cet article** est une bonne introduction, à mon sens.

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

References

[1] Mac Lane *Categories for the working Mathemathician*

[2] Article de Brengos *https://arxiv.org/pdf/1902.02601.pdf*