Espèces de structure en théorie des types homotopiques

Sébastien Draux

Mars 2021

1 Introduction

Les espèces de structures sont des objets combinatoires introduits par Ehresmann dans [1] puis qui ont été surtout développés par Joyal [2]. Ce dernier a mis en évidence l'utilité des séries formelles dans leur étude. La définition et les principales propriétés des espèces ont été reformulées dans le langage de la théorie des types homotopiques par John Dougherty [3]. Celui-ci a également donné une formalisation de ces résultats en Coq.

Le travail qui suit reprend celui de John Dougherty en détaillant les preuves et propose une autre formalisation utilisant une librairie cubique pour Agda [4]. Le but est d'une part de se familiariser avec la théorie des types homotopiques à travers une application et d'autre part de découvrir un assistant de preuve cubique. L'ensemble du code produit est disponible dans un repository git à l'adresse suivante : `https://github.com/SebastienDraux/Species`.

2 Théorie des types homotopiques

La théorie des types homotopiques est une nouvelle théorie développée en 2013 à l'initiative de Voevodsky dans le livre [5]. Il s'agit d'une extension de la théorie des types standards. Rappelons rapidement les principes de la théorie des types avant de voir les innovations apportées par la théorie des types homotopiques.

En théorie des types, on construit des types de la façon suivante. Si A et B sont des types alors
— $A + B$ est le type de l'union disjointe de A et B,
— $A \times B$ est le types des paires de A et B,
— $A \to B$ est le type des fonctions de A dans B.
— ...
On note $a : A$ pour indiquer que l'élément a est de type A. On note $A : \mathcal{U}$ pour indiquer que A est un type. $\mathcal{U}$ est en quelques sorte le "type des types". En réalité, il faut développer tout une hiérarchie de types pour ne pas aboutir à des paradoxes.

Un élément $P : A \to \mathcal{U}$ est une famille de types indexés par les éléments de A. On peut étendre la définition des paires en faisant varier le type de second élément de la paire en fonction du premier. On appelle cela une paire dépendante et pour une famille $P : A \to \mathcal{U}$, on note

$$\sum_{a:A} P\ a$$

le type des paires $(a,\ x)$ avec $a : A$ et $x : P\ a$. De même, on a le type des fonctions dépendante où le type d'arrivée dépend de l'entrée de la fonction. On note cela :

$$\prod_{a:A} P\ a$$

Chaque type vient avec des constructeurs qui donne les règles pour construire un élément de ce type et un principe de récurrence qui donne une façon de construire des fonctions à partir de ce type. Par exemple, le type $A + B$ a deux constructeurs :

— $\textsc{inl} : A \to A + B$

— $\textsc{inr} : B \to A + B$

correspondant respectivement à l'injection de A et B dans $A+B$. Le principe de récurrence énonce que si on a deux fonctions $f_A : A \to C$ et $f_B : B \to C$ alors on a une fonction $f : A + B \to C$ telle que f coïncide avec f_A pour les éléments de la forme $\textsc{inl}\ a$ et avec f_B pour les éléments de la forme $\textsc{inl}\ b$. On note $\bot$ le type qui n'a pas de constructeur et $\top$ le type à un unique constructeur, noté $\mathbf{1}$.

Pour tout $a, a' : A$, on dispose d'un type $a = a'$, ces types sont appelés types identités. On dispose d'un constructeur naturel de $a = a$ noté $\textsc{refl}$ qui correspond à la réflexivité de l'égalité. En théorie des types classiques, on travaille avec un axiome dit d'unicité des preuves d'identité qui assure qu'il n'y a pas d'autres égalités que les $\textsc{refl}$. Si on a $p : a = a'$ alors nécessairement a et a' sont le même élément, ce qu'on note $a \equiv a'$ par opposition à $=$ qui est une égalité propositionnelle et $p \equiv \textsc{refl}$. Pour $a, a', a'' : A$ et $p : a = a'$, $q : a' = a''$, on a $p \cdot q : a = a''$ et $p^{-1} : a' = a$.

Cet axiome n'est plus supposé en théorie des types homotopiques. On peut donc parfaitement avoir des égalités non-triviales. Le type $A \simeq B$ est le type des équivalences entre A et B. Pour montrer qu'une fonction $f : A \to B$ est une équivalence, il suffit de montrer qu'elle admet un inverse à gauche et à droite, c'est-à-dire qu'il existe $g : B \to A$ telle que pour tout $a : A$, $g\ (f\ a) = a$ et pour tout $b : B$, $f\ (g\ b) = b$. On peut construire des égalités non-triviales à partir des équivalence. En effet, un axiome de la théorie des types homotopiques est l'axiome d'univalence qui permet de transformer des équivalence en égalité $\textsc{ua} : (A \simeq B) \simeq (A = B)$. Cet axiome est fonctoriel, c'est-à-dire que $\textsc{ua}\ (g \circ f) = (\textsc{ua}\ f) \cdot (\textsc{ua}\ g)$, $\textsc{ua}\ id_A = \textsc{refl}$ et $\textsc{ua}\ (f^{-1}) = (\textsc{ua}\ f)^{-1}$.

Étant donné deux fonctions $f, g : A \to B$ on dispose d'un moyen de prouver $f = g$ de la façon suivante. Si pour tout $a : A$ on a $f\ a = g\ a$ alors $f = g$. On appelle cette preuve une preuve par extensionnalité. Ce résultat est une conséquence de l'axiome d'univalence.

Enfin, le dernier concept fondamental de la théorie des types homotopiques est celui de transport. Soit $P : A \to \mathcal{U}$, considérons le type des paires dépendantes

$$\sum_{a:A} P\ a$$

Soit (a, x) et (a', x') deux éléments de ce type. Comment prouver $(a, x) = (a', x')$? Si c'était des paires non dépendantes, il suffirait de prouver $a = a'$ et $x = x'$. Or ici, x et x' ne sont pas des éléments du même type donc il n'y a pas de sens à considérer leur égalité. Si on a $p : a = a'$, on peut légitimement à s'attendre $P\ a \simeq P\ a'$. On a effectivement une telle équivalence donnée par :

$$x \mapsto \textsc{transport}^P(p, x)$$

d'inverse

$$x \mapsto \text{TRANSPORT}^P(p^{-1}, x)$$

Ainsi, pour prouver $(a, x) = (a', x')$, exhiber $p : a = a'$ et montrer $\text{TRANSPORT}^P(p, x) = x'$.

Un cas particulier de transport est celui de la congruence de l'égalité. Soit $f : A \to B$ et $a, a' : A$ avec $p : a = a'$, on veut en déduire $f\ a = f\ a'$. Ceci est donné par $\text{CONG}\ f\ p$ où par définition

$$\text{CONG}\ f\ p :\equiv \text{TRANSPORT}^{b \mapsto f\ a = b}(p,\ \text{REFL})$$

où l'exposant $b \mapsto f\ a = b$ est à comprendre à la façon d'un λ, c'est-à-dire comme la famille $P : B \to \mathcal{U}$ telle que $P\ b :\equiv f\ a = b$.

Lorsque le contexte est clair et pour alléger les notations, on peut ne pas préciser explicitement la famille P. Si a, a' tels que $a = a'$ et $x : P\ a$, on note :

$$p_*\ x :\equiv \text{TRANSPORT}^P(p, x)$$

3 Motivation

L'idée des espèces de structures est d'associer à tout ensemble fini un autre ensemble fini par une fonction Φ. Si A est un ensemble fini, les éléments de $\Phi(A)$ sont alors munis d'une Φ-structure. On demande de plus que cette application soit fonctorielle, c'est-à-dire que si deux ensembles finis A et B sont en bijections alors on a une bijection entre $\Phi(A)$ et $\Phi(B)$ et que ceci soit compatible avec la composition et respecte l'identité.

Par exemple, on peut considérer Φ qui à A associe l'ensemble des graphes dont l'ensemble des sommets est A, les éléments de $\Phi(A)$ ont une structure de graphe (sur A). On voit avec cet exemple que la théorie des espèces de structures peut s'appliquer à une très grande classe d'objets combinatoires.

A une espèce de structure Φ, on peut associer une série génératrice :

$$\sum_{n \in \mathbb{N}} \frac{\varphi_n}{n!} z^n$$

où $\varphi_n = |\Phi(\{1, \cdots n\})|$ est le nombre de Φ-structures sur un ensemble de cardinal n. Pour Φ qui a un ensemble associe l'ensemble de ses permutations, on a $\varphi_n = n!$ et donc sa série génératrice vaut $1/(1 - z)$. Les opérations sur les séries formelles telles que l'addition, la multiplication et même la dérivation, trouvent une interprétation combinatoire si bien qu'on dispose d'outils pour étudier les espèces par leur série génératrice.

Il existe un autre point de vue que celui évoqué ci-dessus, on peut raisonner dans l'autre sens. Si on reprend l'exemple des graphes, on peut étant donné un graphe lui associer son ensemble de sommets. Ainsi, une espèce peut aussi se voir comme un ensemble (ici l'ensemble des graphes) muni d'une application qui à un élément de cet ensemble associe un ensemble fini. On retrouvera ces deux façons de voir les choses dans la suite lorsqu'on étudiera les structures en théorie des types homotopiques.

Les espèces de structures ont d'abord été utilisés par Ehresmann [1] mais c'est Joyal [2] qui le premier a mis en évidence toute la richesse et l'intérêt de ces objets. Pour une étude détaillée des espèces de structures, on peut se référer à [6].

4 Ensembles finis

Avant de définir les espèces, une discussion préalable est nécessaire sur les ensembles finis. Tout d'abord, on commence par définir des types représentant des ensembles finis canoniques de cardinal fixé. On présente plusieurs définitions équivalentes de ces types.

Définition 1. *Soit* $n : \mathbb{N}$.

— *Soit* $\textsc{Fin} : \mathbb{N} \to \mathcal{U}$ *défini de la façon suivante :*

$$\textsc{Fin}\ n :\equiv \sum_{m:\mathbb{N}} m < n$$

— *Soit* $\textsc{Fin'} : \mathbb{N} \to \mathcal{U}$ *défini par les constructeurs :*

$$\textsc{zero} : \textsc{Fin'}\ (\textsc{suc}\ n)$$
$$\textsc{suc} : \textsc{Fin'}\ n \to \textsc{Fin'}\ (\textsc{suc}\ n)$$

— *Soit* $\textsc{Fin"} : \mathbb{N} \to \mathcal{U}$ *défini de façon récursive par :*

$$\textsc{Fin"}\ 0 :\equiv \bot$$
$$\textsc{Fin"}\ (\textsc{suc}\ n) :\equiv (\textsc{Fin"}\ n) + \top$$

Ces trois définitions correspondent intuitivement à l'ensemble canonique à n éléments $\{0, \ldots, n-1\}$ vu de plusieurs façons différentes. La première définition exprime $\{0, \ldots, n-1\}$ comme l'ensemble des $m \in \mathbb{N}$ strictement inférieurs à n. La deuxième définition décrit les éléments de $\{0, \ldots, n\}$ comme étant soit 0 soit le successeur d'un élément de $\{0,\ ...,\ n-1\}$. Enfin la troisième définition définit $\{0, \ldots, n\}$ comme $\{0, \ldots, n-1\}$ auquel on adjoint un élément. Ces définitions sont toutes équivalentes.

Proposition 1. *Pour tout* $n : \mathbb{N}$

$$\textsc{Fin}\ n \simeq \textsc{Fin'}\ n \simeq \textsc{Fin"}\ n$$

Démonstration. L'équivalence $\textsc{Fin'}\ n \simeq \textsc{Fin"}\ n$ est assez claire mathématiquement. L'idée intuitive est bien sûr de faire correspondre d'une part le $\textsc{zero}$ de $\textsc{Fin'}\ (\textsc{suc}\ n)$ à l'élément de $\top$ dans $\textsc{Fin"}\ (\textsc{suc}\ n)$ et d'autre part le constructeur $\textsc{suc} : \textsc{Fin'}\ n \to \textsc{Fin'}\ (\textsc{suc}\ n)$ à l'injection de $\textsc{Fin"}\ n$ dans $\textsc{Fin"}\ (\textsc{suc}\ n)$.
Formalisons cette démonstration par disjonction de cas sur n. Le type $\textsc{Fin'}\ 0$ n'a pas de constructeur, on peut donc définir un élément de $\textsc{Fin'}\ 0 \to \bot$ par disjonction de cas. De même, on définit un élément de $\bot \to \textsc{Fin'}\ 0$. On peut alors montrer que ces éléments sont inverses l'une de l'autre une nouvelle fois par disjonction de cas.
Soit $f : \textsc{Fin'}\ (\textsc{suc}\ n) \to \textsc{Fin"}\ (\textsc{suc}\ n)$ défini de la façon suivante :

$$f\ \textsc{zero} :\equiv \textsc{inr}\ \mathbf{1}$$
$$f\ (\textsc{suc}\ k) :\equiv \textsc{inl}\ k$$

Soit $g : \text{FIN"}\ (\text{SUC}\ n) \to \text{FIN'}\ (\text{SUC}\ n)$ tel que :

$$g\ (\text{INR}\ \mathbf{1}) :\equiv \text{ZERO}$$
$$g\ (\text{INL}\ k) :\equiv \text{SUC}\ k$$

On a facilement par disjonction de cas que ces deux fonctions sont inverses l'une de l'autre dans les deux sens.

Pour démontrer que $\text{FIN}\ n \simeq \text{FIN"}\ n$, il faut préciser un peu plus la définition du premier type. Pour tout $m, n : \mathbb{N}$ on définit :

$$m < n :\equiv \sum_{k:\mathbb{N}} \text{SUC}\ (m + k) = n$$

Définissons un concept supplémentaire.

Définition 2 (Proposition)**.** *Un type X est une proposition si pour tout $x, x' : X$ alors $x = x'$ ce qui s'écrit formellement*

$$\prod_{x,x':X} x = x'$$

Les types identités sur $\mathbb{N}$ sont des propositions. Il s'en suit que pour montrer l'égalité entre deux éléments du type $m < n$, il suffit de vérifier l'égalité de leur première composante. La démonstration de ce fait est donnée dans le lemme 1 ci-dessous. Soient $(k,\ p)$, $(k',\ p')$ deux éléments de $m < n$. On a donc $p : \text{SUC}\ (m+k) = n$ et $p' : \text{SUC}\ (m+k') = n$ d'où on déduit $k = k'$ par injectivité de SUC et de l'addition. Ainsi $(k,\ p) = (k',\ p')$. Ceci montre que $m < n$ est une proposition. Toujours d'après le lemme 1, pour montrer l'égalité de deux éléments de $\text{FIN}\ n$, il suffit de démontrer l'égalité de leur première composante.

Montrons maintenant que les types $\text{FIN}\ n$ vérifient la même relation de récurrence que les $\text{FIN"}\ n$, c'est-à-dire :

$$\text{FIN}\ 0 = \bot$$
$$\text{FIN}\ (\text{SUC}\ n) = (\text{FIN}\ n)\ +\ \top$$

Traitons le premier cas. Pour $m, k : \mathbb{N}$, on a $\text{SUC}\ (m+k) = 0 \to \bot$ et ainsi $\text{SUC}\ (m+k) = 0 \simeq \bot$ d'où $m < 0 \simeq \bot$ et on en déduit $\text{FIN}\ 0 \simeq \bot$ puis l'égalité par univalence.
Soit maintenant $f : \text{FIN}\ (\text{SUC}\ n) \to (\text{FIN}\ n) + \top$ défini de la façon suivante :

$$f\ (0,\ ineq) :\equiv \text{INR}\ \mathbf{1}$$
$$f\ (\text{SUC}\ m,\ k,\ p) :\equiv \text{INR}\ (m,\ k,\ p')$$

avec $p' : \text{SUC}\ (m + k) = n$ déduite de $p : \text{SUC}\ ((\text{SUC}\ m) + k) = \text{SUC}\ n$ en utilisant :

$$\text{SUC}\ ((\text{SUC}\ m) + k) = \text{SUC}\ n \to (\text{SUC}\ m) + k = n \to \text{SUC}\ (m + k) = n$$

Soit $g : \text{FIN}\ n + \top \to \text{FIN}\ (\text{SUC}\ n)$ défini par :

$$g\ (\text{INR}\ \mathbf{1}) :\equiv (0,\ n,\ \text{REFL})$$
$$g\ (\text{INL}\ (m,\ k,\ p')) :\equiv (\text{SUC}\ m,\ k,\ p)$$

où p est déduite de p' par l'implication réciproque :

$$\text{SUC } (m + k) = n \to \text{SUC } ((\text{SUC } m) + k) = \text{SUC } n$$

On a alors facilement par disjonction de cas que les deux fonctions sont inverses l'une de l'autre en utilisant la remarque préalable qui stipule qu'il suffit de s'intéresser à la première composante. Les familles FIN et FIN" vérifient les mêmes relation de récurrence, on montre alors par récurrence que $\text{FIN } n \simeq \text{FIN" } n$ □

Lemme 1. *Soit $P : A \to \mathcal{U}$ tel que pour tout $a : A$ le type $P\ a$ soit une proposition. Soit*

$$w, w' : \sum_{a:A} P\ a$$

tels que $\text{PR}_1\ w = \text{PR}_1\ w'$, alors $w = w'$

Démonstration. Soit $p : \text{PR}_1\ w = \text{PR}_1\ w'$, il faut montrer que $p_*\ (\text{PR}_2\ w) = \text{PR}_2\ w'$. Or ce sont deux élément de $P\ (\text{PR}_2\ w')$, ils sont donc égaux par hypothèse. □

En utilisant la troisième définition, on peut démontrer par récurrence les résultats suivants sur les types FIN.

Proposition 2. *Pour tout $m,\ n : \mathbb{N}$*

$$\text{FIN } (n + m) \simeq \text{FIN } n + \text{FIN } m$$
$$\text{FIN } (n \cdot m) \simeq \text{FIN } n \times \text{FIN } m$$

On a enfin le résultat très intuitif suivant d'injectivité.

Proposition 3. *Pour tout $m, n : \mathbb{N}$*

$$\text{FIN } n \simeq \text{FIN } m \to n = m$$

Cette propriété est une conséquence du lemme suivant.

Lemme 2. *Pour tout type $A,\ B$*

$$A + \top \simeq B + \top \to A \simeq B$$

Démonstration. De nouveau, le résultat est assez naturel mathématiquement parlant. En effet, si on se donne une équivalence $f : A + \top \to B + \top$. On distingue deux cas :

— Si $f\ (\text{INR } \mathbf{1}) = \text{INR } \mathbf{1}$ alors la restriction de f à A est une équivalence $A \simeq B$.

— Si $f\ (\text{INR } \mathbf{1}) = \text{INL } b_0$, alors $f^{-1}(\text{INR } \mathbf{1}) = \text{INL } a_0$ pour un certain $a_0 : A$. On peut alors définir $f' : A \to B$ de la façon suivante. D'une part $f'\ a_0 = b_0$ et si $a \neq a_0$ alors $f\ (\text{INL } a) = \text{INL } b$ pour un certain $b : B$ et on pose alors $f'\ a = b$.
Il faut cependant faire attention, le raisonnement ci-dessus n'est pas tout à fait valable. En effet, il est tout à fait possible que l'égalité ne soit pas décidable dans A et donc que la disjonction de cas dans la définition de f' ne soit pas permise. Il faut définir f' de la façon suivante pour contourner le problème :

$$f'\ a :\equiv \begin{cases} b_0 & \text{si } f\ (\text{INL } a) = \text{INR } \mathbf{1} \\ b & \text{si } f\ (\text{INL } a) = \text{INL } b \text{ pour un certain } b : B \end{cases}$$

Cette disjonction de cas est cette fois parfaitement valide.

□

Démontrons maintenant la propriété 3

Démonstration. On raisonne par double récurrence.

— Si Fin $0 \simeq$ Fin 0 alors refl donne une preuve de $0 = 0$.
— Si Fin $0 \simeq$ Fin (suc m) alors comme on sait construire un élément de Fin (suc m) avec inr **1**, on a $\bot$ et donc $0 =$ suc m. Le cas Fin (suc n) $\simeq$ Fin 0 est identique.
— Si Fin (suc n) $\simeq$ Fin (suc m) alors Fin $n \simeq$ Fin m d'après le lemme précédent et donc $n = m$ par récurrence et donc suc $n =$ suc m

□

La formalisation de ce résultat est disponible dans le fichier `lemma.agda` du repository git et s'écrit :

```
inj-⊎-Unit:(A ⊎ Unit) ≃ (B ⊎ Unit) → A ≃ B
```

Ainsi, d'après les propriétés précédentes, les types Fin n se comportent exactement comme on s'attend à ce qu'il le fasse. On a réussi à formaliser dans le langage de la théorie des types homotopiques les principales propriétés de l'ensemble $\{0, ..., n-1\}$. On souhaite maintenant définir le type des ensembles fini. Intuitivement, un type A est un ensemble fini si pour un certain $n : \mathbb{N}$, A est équivalent à Fin n, ce qui est la même chose que de dire que $A =$ Fin n par univalence. Une première définition des ensembles finis pourrait être la suivante :

Définition 3 (Ensembles finis naïfs)**.** *On définit le type des ensembles finis naïf de la façon suivante :*

$$\text{naiveFinSet} :\equiv \sum_{A:\mathcal{U}} \sum_{n:\mathbb{N}} A = \text{Fin } n$$

Malheureusement, cette définition n'est pas satisfaisante. En effet, on a le résultat suivant.

Proposition 4. naiveFinSet $\simeq \mathbb{N}$

Démonstration. Il suffit d'appliquer le lemme suivant avec $X :\equiv \mathbb{N}$, $B :\equiv \mathcal{U}$ et $f :\equiv$ Fin. Ce lemme est le lemme 4.8.2 de [5]. □

Lemme 3. *Pour tout type X et B et toute application $f : X \to B$*

$$\sum_{b:B} \text{fib}_f\ b \simeq X$$

où par définition, $\text{fib}_f\ b$ *est le type des $x : X$ tels que $f\ x = b$, c'est-à-dire*

$$\text{fib}_f\ b :\equiv \sum_{x:X} f\ x = b$$

Ce lemme est lui-même une conséquence des deux lemmes suivants. Le premier lemme est le lemme 3.11.9 i de [5]. Le second lemme énonce que tous les singletons sont contractibles. Dire qu'un type X est contractible signifie qu'on peut prouver

$$\sum_{x_0:X} \prod_{x:X} x = x_0$$

c'est-à-dire qu'on peut exhiber un élément x_0 de X, appelé le centre, tel que tous les éléments $x : X$ soit égaux à ce x_0. Autrement dit un type contractible est un type qui n'a "qu'un seul élément".

Lemme 4. *Soit $P : X \to \mathcal{U}$ une famille au-dessus de X tel que pour tout $x : X$ le type $P\ x$ soit contractible alors*

$$\sum_{x:X} P\ x \simeq X$$

Démonstration. De gauche à droite, on prend simplement PR_1. Dans l'autre sens, on pose $g\ x :\equiv (x, c_x)$ où c_x est le centre de $P\ x$. On a clairement $\text{PR}_1\ (g\ x) = x$. Réciproquement soit $y : P\ x$ alors : $g\ (\text{PR}_1\ (x, y)) \equiv (x, c_x)$. On a $x = x$ avec REFL et :

$$\text{TRANSPORT}^P(\text{REFL}, c_x) = c_x = y$$

où la première égalité vient du fait que le transport le long de REFL est l'identité et la seconde de la contractibilité de $P\ x$. □

Lemme 5. *Soit $x_0 : X$ alors*

$$\text{SINGL}\ x_0 :\equiv \sum_{x:X} x = x_0$$

est contractible.

Démonstration. On choisit comme centre (x_0, REFL). Soit $(x, p) : \text{SINGL}\ x$. On a $x = x_0$ par p et de plus :

$$\text{TRANSPORT}^{x \mapsto x = x_0}(p, p) = p^{-1} \cdot p = \text{REFL}$$

D'après le lemme 2.11.2 de [5] sur le transport pour les types identité. □

Revenons à la démonstration de lemme 3.

Démonstration. On a

$$\sum_{b:B} \text{FIB}_f\ b \equiv \sum_{b:B} \sum_{x:X} f\ x = b \simeq \sum_{x:X} \sum_{b:B} f\ x = b \equiv \sum_{x:X} \text{SINGL}\ (f\ x) \simeq X$$

La première équivalence vient du fait que les Σ commutent et la seconde est une conséquence des lemmes 4 et 5. □

Il faut donc une autre notion d'ensemble fini. La solution pour contourner ce problème est d'utiliser une troncation propositionnelle.

Définition 4 (Troncation propositionnelle)**.** *Soit X un type, on définit $|X|$, la troncation propositionnelle de X par les constructeurs suivants :*

- *Pour $x : X$, on a $|x| : \|X\|$,*
- *Pour $\tilde{x}_1, \tilde{x}_2 : \|X\|$, on a $\tilde{x}_1 = \tilde{x}_2$*

La troncation propositionnelle contracte donc tous les élément de X en un seul.

Proposition 5. *Pour tout $n, m : \mathbb{N}$*

$$\|\text{FIN}\ n \simeq \text{FIN}\ m\| \to n = m$$

Démonstration. Il s'agit d'une conséquence du principe de récurrence des proposition qui stipule $(X \to P) \to \|X\| \to P$ lorsque P est une proposition. On applique cela à $X :\equiv \text{FIN}\ n \simeq \text{FIN}\ m$ et $P :\equiv n = m$ en appliquant la proposition 3. □

La bonne définition des ensembles finis est la suivante.

Définition 5 (Ensembles finis)**.** *On définit le type des ensembles finis de la façon suivante :*

$$\text{FINSET} :\equiv \sum_{A:\mathcal{U}} \sum_{n:\mathbb{N}} \|A = \text{FIN}\ n\|$$

5 Espèces

Pour définir les espèces en théorie des types homotopiques, on utilise le deuxième point de vue évoqué plus haut.

Définition 6 (Espèces)**.** *On pose*

$$\textsc{Species} :\equiv \sum_{X:\mathcal{U}} X \to \textsc{finSet}$$

Comme annoncé, cette façon de voir les espèce est équivalente à l'autre point de vue développé.

Proposition 6. $\textsc{Species} \simeq \textsc{finSet} \to \mathcal{U}$

Ce résultat est une conséquence de la proposition plus générale suivante.

Proposition 7. *Pour tout type B :*

$$\sum_{X:\mathcal{U}} X \to B \simeq B \to \mathcal{U}$$

Démonstration. Comme remarqué plus haut, la correspondance se fait par les fibres. Pour (X, f) une espèce, on pose

$$F\ (X, f) :\equiv \textsc{fib}_f$$

et pour P une famille de type sur B, on pose

$$G\ P :\equiv \left(\sum_{b:B} P\ b, \textsc{pr}_1 \right)$$

Vérifions que ces deux applications sont inverses l'une de l'autre.
D'une part, pour une espèce $(X,\ f)$

$$G\ (F\ (X, f)) \equiv \left(\sum_{b:B} \textsc{fib}_f\ b, \textsc{pr}_1 \right)$$

Le lemme 3 nous donne que la première composante est équivalente (et donc égale par univalence) à X. Notons $e\ f$ cette équivalence. Il faut alors démontrer que

$$\textsc{transport}^{X \mapsto X \to B}(\textsc{ua}\ e,\ \textsc{pr}_1) = f$$

On a

$$\begin{aligned}
(\textsc{transport}^{X \mapsto X \to B}(\textsc{ua}\ e, \textsc{pr}_1))\ x \\
&= \textsc{transport}^{X \mapsto B}(\textsc{ua}\ (e\ f), \textsc{pr}_1\ (\textsc{transport}^{X \mapsto X}((\textsc{ua}\ (e\ f))^{-1}, x))) \\
&= \textsc{pr}_1\ (\textsc{transport}^{X \mapsto X}((\textsc{ua}\ (e\ f))^{-1}, x)) \\
&= \textsc{pr}_1\ (\textsc{transport}^{X \mapsto X}(\textsc{ua}\ ((e\ f)^{-1}), x)) \\
&= \textsc{pr}_1\ ((e\ f)^{-1}\ x)
\end{aligned}$$

La première égalité vient de l'égalité (2.9.4) de [5] sur le transport pour les familles de fonctions non dépendantes. La seconde égalité résulte du fait que le transport pour une famille constante est l'identité. La troisième égalité découle de la fonctorialité de UA et la dernière de la règle de calcul pour UA.

L'équivalence construite dans le lemme 3 est telle que $(e\ f)^{-1}\ x = (f\ x, x, \text{REFL})$. Ainsi pour tout $x : X$, on a

$$(\text{TRANSPORT}^{X \mapsto X \to B}(\text{UA}e, \text{PR}_1))\ x = f\ x$$

On conclue alors par extensionnalité. On a donc montré que

$$G\ (F\ (X, f)) = (X, f)$$

Montrons maintenant que $F\ (G\ P) = P$ pour $P : B \to \mathcal{U}$. Pour cela, on montre que pour tout $b : B$ on a $F\ (G\ P)\ b \simeq P\ b$. On en déduira par univalence $F\ (G\ P)\ b = P\ b$ puis $F\ (G\ P) = P$ par extensionnalité. On a :

$$F\ (G\ P)\ b \equiv \text{FIB}_{\text{PR}_1}\ b \equiv \sum_{w:\sum_{b':B} P\ b'} \text{PR}_1\ w = b \simeq \sum_{b':B} \sum_{y':P\ b'} b' = b \equiv \sum_{b':B} (P\ b' \times (b' = b))$$

L'équivalence

$$\sum_{w:\sum_{b':B} P\ b'} \text{PR}_1\ w = b \simeq \sum_{b':B} \sum_{y':P\ b'} b' = b$$

provient d'un propriété "d'associativité" des types Σ : l'application $((b', y'), p) \mapsto (b', y', p)$ est une équivalence. Voir exercice 2.10 de [5].

Posons $g\ (b', y', p) :\equiv \text{TRANSPORT}^P(p, y')$ et $h\ y :\equiv (b, y, \text{REFL})$. On a d'une part :

$$g\ (h\ y) \equiv \text{TRANSPORT}^P(\text{REFL}, y) = y$$

D'autre part :

$$h\ (g\ (b', y', p)) \equiv (b, \text{TRANSPORT}^P(p, y'), \text{REFL})$$

On montre $b = b'$ avec p^{-1}. De plus :

$$\begin{aligned} p_*^{-1}(\text{TRANSPORT}^P(p, y'),\ \text{REFL}) \\ &= (\text{TRANSPORT}^P(p^{-1}, \text{TRANSPORT}^P(p, y')),\ \text{TRANSPORT}^{b' \to b' = b}(p^{-1}, \text{REFL})) \\ &= (y', p) \end{aligned}$$

La première égalité provient du théorème 2.6.4 de [5] sur le transport pour les produits. La seconde égalité, elle, vient du fait que $\text{TRANSPORT}^P p^{-1} = (\text{TRANSPORT}^P\ p)^{-1}$ pour la première composante et de la propriété du transport pour les types identité pour la seconde composante. Ceci termine la démonstration. □

La formalisation de cette preuve est plus laborieuse pour des raisons qui seront détaillées en conclusion. Elle est disponible dans le fichier `lemma.agda` et s'écrit :

$$\texttt{equivClassification:\{B:Type}\ \ell\texttt{\}} \to (\Sigma[\texttt{X} \in \texttt{Type}\ell]\ (\texttt{X} \to \texttt{B})) \simeq (\texttt{B} \to \texttt{Type}\ \ell)$$

6 Cardinal

On dispose d'une notion naturelle de cardinal sur les ensembles finis.

Définition 7. *On définit* $\textsc{card} : \textsc{finSet} \to \mathbb{N}$ *de la façon suivante :*

$$\textsc{card}\ (A, n, p) :\equiv n$$

Dans la suite, on notera $\textsc{card}\ A$ pour $\textsc{card}\ (A,\ n,\ p)$. Cet abus de notation est sans danger d'après la proposition suivante.

Proposition 8. *Si* $(A,\ n,\ p)$ *et* $(A,\ m,\ q)$ *sont deux éléments de* finSet *alors* $n = m$

Énonçons d'abord le lemme suivant.

Lemme 6. *Si* P *est une proposition, alors* $(A \to B \to P) \to \|A\| \to \|B\| \to P$

Démonstration. Il s'agit d'une conséquence du principe de récurrence des propositions. □

Démontrons la propriété précédente.

Démonstration. On a $p : \|A = \textsc{Fin}\ n\|$ et $q : \|A = \textsc{Fin}\ m\|$ d'où on peut déduire $\|\textsc{Fin}\ n = \textsc{Fin}\ m\|$ d'après le lemme 6 et donc $n = m$ d'après la proposition 5. □

Le cardinal vérifie la propriété suivante de compatibilité avec les équivalences.

Proposition 9. *Si* $A,\ A'$ *sont des ensembles finis tels que* $\|A \simeq A'\|$ *alors* $\textsc{card}\ A = \textsc{card}\ A'$

Démonstration. Soient $n,\ m : \mathbb{N}$ tels que $\|A = \textsc{Fin}\ n\|$ et $\|A' = \textsc{Fin}\ m\|$ alors $\|A = \textsc{Fin}\ m\|$ puis $\|\textsc{Fin}\ n = \textsc{Fin}\ m\|$ d'après le lemme 6 et donc $n = m$. □

On peut généraliser la notion de cardinal pour une classe de types beaucoup plus grande que les ensembles finis. Ce qu'on peut faire dans un premier temps, c'est de compter les éléments d'un type X à égalité près, c'est-à-dire considérer $\textsc{card}\ \pi_0(X)$. Bien sûr ce cardinal peut tout à fait ne pas être fini. Lorsque X est un ensemble, cette notion de cardinalité est suffisante mais en générale les types sont beaucoup plus riches. On peut donc aller plus loin et regarder en détail ce qui se passe dans chaque composante de X en considérant sa structure de groupoïde. Si $x : X$ a des automorphismes non triviaux (si le type $x = x$ n'est pas restreint à refl), on aimerait quotienter X par l'action de $\pi_1(X, x)$. Ceci nous amène à la notion de cardinal de groupoïde défini par :

$$\sum_{x:\pi_0(X)} \frac{1}{\textsc{card}\ (\pi_1(X, x))}$$

On peut maintenant prendre en compte la structure de $\infty-$groupoïde des types pour définir une nouvelle notion de cardinalité.

Définition 8 (Cardinal d'∞-groupoïde)**.** *Le cardinal d'*∞*-groupoïde d'un type* X *est*

$$\textsc{gcard}\ X := \sum_{x:\pi_0(X)} \frac{1}{\textsc{card}\ (\pi_1(X, x))} \textsc{card}\ (\pi_2(X, x)) \frac{1}{\textsc{card}\ (\pi_3(X, x))} \dots$$

lorsque cette quantité est bien définie.

Ce cardinal à les propriétés suivantes :

Proposition 10. *Pour $X, Y : \mathcal{U}$, on a :*
— $\textsc{gcard}\ \top = 1$
— $\textsc{gcard}\ (X + Y) = (\textsc{gcard}\ X) + (\textsc{gcard}\ Y)$
— $\|X \simeq Y\| \to \textsc{gcard}\ X = \textsc{gcard}\ Y$

De ces propriétés, on peut déduire que ce nouveau cardinal coïncide bien avec celui utilisé précédemment pour les ensembles finis.

Proposition 11. *Pour tout ensemble fini A :* $\textsc{gcard}\ A = \textsc{card}\ A$

Démonstration. On a d'abord par récurrence que pour tout $n : \mathbb{N}$: $\textsc{gcard}\ (\textsc{Fin}\ n) = n$ par les deux premiers points de la proposition 10. On en déduit par le troisième point que $\|A = \textsc{Fin}\ n\|$ implique $\textsc{gcard}\ A = n$. □

7 Séries formelles

A partir des notions d'espèce et de cardinal définis ci-dessus, John Dougherty [3] propose la définition suivante de la série génératrice associée à une espèce.

Définition 9 (Série génératrice d'une espèce)**.** *Soit (X, f) une espèce, on pose :*

$$|X|(z) :\equiv \sum_{n \in \mathbb{N}} X_n z^n$$

avec

$$X_n :\equiv \textsc{gcard}\ (\textsc{fib}_{\textsc{card} \circ f}\ n)$$

La suite n'a pas fait l'objet d'une étude aussi approfondie que ce qui précède et n'a pas été formalisé dans un assistant de preuve en entier. On se contente de reprendre rapidement les résultats de Dougherty.

On peut démontrer que le coefficient X_n peut être calculé uniquement à partir de la fibre de de f au-dessus de $\textsc{Fin}\ n$:

$$X_n = \frac{1}{n!}(\textsc{gcard}\ (\textsc{fib}_f\ (\textsc{Fin}\ n)))$$

On retrouve ainsi la définition de la série génératrice associée à une espèce donnée plus haut. La première étape du raisonnement consiste à montrer que

$$\textsc{fib}_{\textsc{card} \circ f}\ n = \sum_{w : \sum_{A:\mathcal{U}} \|A = \textsc{Fin}\ n\|} \textsc{fib}_f\ (\textsc{Fin}\ n)$$

Ceci se démontre avec le lemme suivant qui constitue l'exercice 4.4 de [5]. Cette démonstration a été formalisé.

Lemme 7. *Soit $f : A \to B$ et $g : B \to C$ alors pour tout $c : C$, on a*

$$\textsc{fib}_{g \circ f}\ c \simeq \sum_{w : \textsc{fib}_g\ c} \textsc{fib}_f\ (\textsc{pr}_1\ w)$$

Démonstration. Un élément de

$$\sum_{w:\text{FIB}_g\ c} \text{FIB}_f\ (\text{PR}_1\ w)$$

est un triplet (w, a, q) où $w : \text{FIB}_g\ c$ est lui-même un couple (b, p) avec $b : B$, $p : g\ b = c$, $a : A$ et $q : f\ a = b$.

Soit

$$F : \text{FIB}_{g\circ f}\ c \to \sum_{w:\text{FIB}_g\ c} \text{FIB}_f\ (\text{PR}_1\ w)$$

tel que $F\ (a, p) :\equiv ((f\ a, p), a, \text{REFL})$ et

$$G : \sum_{w:\text{FIB}_g\ c} \text{FIB}_f\ (\text{PR}_1\ w) \to \text{FIB}_{g\circ f}\ c$$

tel que $G\ ((b, p), a, q) :\equiv a, (\text{CONG}\quad g\ q) \cdot p$.

D'une part, on a immédiatement :

$$G\ (F\ (a, p)) = (a, p)$$

D'autre part :

$$F\ (G\ ((b, p), a, q)) \equiv ((f\ a, (\text{CONG}\quad g\ q) \cdot p), a, \text{REFL})$$

On a $f\ a = b$ par q et

$$\text{TRANSPORT}^{x\mapsto g\ x=c}(q,\ (\text{CONG}\quad g\ q) \cdot p) = (\text{CONG}\quad g\ q)^{-1} \cdot (\text{CONG}\quad g\ q) \cdot p = p$$

On obtient l'égalité des deux dernières composantes de façon similaire. □

La suite consiste à utiliser des propriétés sur le cardinal des types Σ pour montrer que

$$X_n = \left(\text{GCARD}\ \left(\sum_{A:\mathcal{U}} \|A = \text{FIN}\ n\|\right)\right) \cdot (\text{GCARD}\ (\text{FIB}_f\ (\text{FIN}\ n)))$$

Enfin la dernière étape est d'étudier le cardinal de différents types d'automorphismes pour montrer que

$$\text{GCARD}\ \left(\sum_{A:\mathcal{U}} \|A = \text{FIN}\ n\|\right) = (\text{GCARD}\ (\text{AUT}\ (\text{FIN}\ n)))^{-1} = 1/n!$$

où pour un type X :

$$\text{AUT}\ X :\equiv (X = X)$$

8 Conclusion et remarques

Ce projet à été une très bonne opportunité de découvrir la théorie des types homotopiques par une application aux espèces de structures. De nombreuses difficultés sont apparues tout au long du travail. Tout d'abord il a fallu assimiler les concepts de la théorie des types homotopiques et apprendre à les manipuler. Ensuite, la découverte de la librairie cubique d'Agda n'a pas été évidente

non plus. En effet, le vocabulaire de la théorie des types homotopiques cubiques est différent de celui développé dans [5] ce qui a constitué un obstacle important, en particulier au début. N'étant pas initialement totalement à l'aise avec la théorie des types homotopiques, cette couche cubique supplémentaire a donc été d'autant plus difficile à prendre en main.

J'ai donc eu l'impression, pendant une partie du projet de ne pas vraiment comprendre les preuves que j'écrivais en Agda et de remplir les trous avec des objets du type adéquat en utilisant les outils et raccourcis mis à disposition par Agda comme par exemple les disjonctions de cas automatiques ou le raffinement. Ceci menait à des preuves pas forcement les plus courtes et souvent assez illisibles. J'ai finalement véritablement compris ce qui était en jeu en reprenant les preuves mathématiquement pour l'écriture de ce rapport. Il s'agit typiquement du cas de la formalisation de la preuve de la proposition 6.

L'autre difficulté majeure rencontrée relève plus des assistants de preuves en général. Les preuves sont longues et fastidieuses. Je pense ici à la preuve du lemme 2. Il a fallu traiter de très nombreux cas d'abord pour définir les deux applications puis ensuite pour montrer qu'elles sont inverses l'une de l'autre dans les deux sens. De plus la majorité des cas conduisent à des contradictions mais doivent tout de même être traités pour que l'assistant de preuve valide bien que la preuve est exhaustive. Ceci a rendu ce lemme très difficile d'utilisation dans la suite du travail.

Pour la suite, il pourrait être intéressant d'explorer plus en détail les opérations sur les séries formelles et de formaliser leurs propriétés en Agda cubique. Enfin, on pourrait également imaginer formaliser un exemple concret de calcul de la série génératrice d'une espèce particulière.

Références

[1] C. Ehresmann, "Catégories structurées," in *Annales scientifiques de l'École Normale Supérieure*, vol. 80, pp. 349–426, 1963.

[2] A. Joyal, "Une théorie combinatoire des séries formelles," *Advances in mathematics*, vol. 42, no. 1, pp. 1–82, 1981.

[3] J. Dougherty, "Species in hott." `https://github.com/jdoughertyii/hott-species`.

[4] "Cubical agda." `https://github.com/agda/cubical`.

[5] The Univalent Foundations Program, *Homotopy Type Theory : Univalent Foundations of Mathematics*. Institute for Advanced Study : `https://homotopytypetheory.org/book`, 2013.

[6] F. Bergeron, G. Labelle, and P. Leroux, "Introduction to the theory of species of structures," *Université du Québeca Montréal, Montreal*, 2008.