

THÉORIE DES TYPES DÉPENDANTS ET AXIOME D'UNIVALENCE

par **Thierry COQUAND**

INTRODUCTION

Le but de cet exposé est de présenter l'axiome d'univalence de V. Voevodsky ainsi que son programme pour fonder les mathématiques non sur la théorie des ensembles [Bou1], mais sur une théorie des types qui utilise de manière essentielle la notion de types dépendants.

Ce programme repose sur la stratification suivante des objets mathématiques. Le niveau de base est celui des ensembles. On peut considérer à ce niveau les structures algébriques (groupes, anneaux, modules, ...) ou les structures d'ordre par exemple et il est possible de parler de structures initiales, qui sont uniques à isomorphisme près. Le niveau suivant est celui des *groupoïdes*⁽¹⁾. À ce niveau se situe la notion de catégorie qui sera décrite ici comme une *structure* au niveau des groupoïdes (et qui correspond à la notion de structure d'ordre au niveau des ensembles). La notion d'*isomorphisme* est remplacée par la notion d'*équivalence* catégorique. En continuant cette stratification, on arrive à la vision que les objets mathématiques consistent en des n -groupoïdes de niveau arbitraire et des structures sur ces n -groupoïdes. En prenant en compte la correspondance envisagée par Grothendieck entre les ∞ -groupoïdes et les types d'homotopies [Gr], on peut reformuler cette description en disant que les mathématiques consistent en *l'analyse de structures sur les types d'homotopie*.

La remarque suivante, due à Voevodsky, est que les objets mathématiques obéissent à des lois uniformes par rapport à cette stratification. De plus, et de manière inattendue, ces lois sont décrites avantageusement par la théorie des *types dépendants*, théorie développée pour la représentation formelle des preuves mathématiques [Br, Go, FT]. En fait, on verra qu'à partir de lois purement logiques de l'égalité, en particulier la loi 2.1 mise en évidence par Martin-Löf [M-L], tout type a automatiquement une structure d' ∞ -groupoïde. Cette interprétation permet à son tour de formuler de nouvelles lois logiques pour l'égalité, et en particulier *l'axiome d'univalence*, qui est une vaste

1. On considère usuellement que le niveau suivant est celui des catégories, comme par exemple dans le papier de Makkai [Mak] qui a influencé les travaux présentés ici. Une des contributions de Voevodsky est de reconnaître que la notion de groupoïde est en fait plus fondamentale que celle de catégorie.

généralisation de l’axiome d’extensionnalité, et qui suggère une meilleure représentation formelle des concepts mathématiques dans la théorie des types.

Pour introduire la théorie des types dépendants, je commence par une présentation rapide de la logique d’ordre supérieur. Ce système est essentiellement celui utilisé par T.C. Hales pour vérifier formellement sa preuve de la conjecture de Kepler [Ha]. C’est aussi la logique qui correspond à la notion de « topos élémentaire », de Lawvere et Tierney [Ca]. Le calcul mis au point par Voevodsky en est une extension. Deux axiomes de la logique d’ordre supérieur jouent un rôle clef : l’axiome de l’extensionnalité pour les propositions (« deux propositions équivalentes sont égales »), dont l’axiome d’univalence est une généralisation directe, et l’axiome de description qui permet de nommer un objet si cet objet est défini de manière unique. Ce dernier axiome devient *prouvable* dans le système de Voevodsky qui en contient une vaste généralisation.

Un aspect inhabituel de ce papier, de même que pour le texte [Voe] et le livre [UnFo], est qu’il consiste essentiellement en un commentaire informel d’un texte formalisé. En fait la plupart des résultats qui sont présentés ici ont d’abord été exprimés directement de manière formelle [Voe, AHR].

1. LOGIQUE D’ORDRE SUPÉRIEUR

1.1. Types et termes

Ce système logique, d’une structure simple et naturelle [Ch, Ha], servira d’introduction à la théorie des types dépendants. On classe les termes de cette logique par des *types*, et les *propositions* sont des termes d’un type particulier. On donne ensuite les règles de déduction, qui décrivent quand une proposition est *prouvable*.

Les types de base sont le type des propositions *bool* et le type des *individus* I . Si A et B sont des types, on peut former le type $A \rightarrow B$ des *fonctions* du type A dans le type B . Par exemple, $I \rightarrow I$ sera le type des fonctions, alors que $I \rightarrow \text{bool}$ sera le type des prédicats sur le type I . Le type des quantificateurs sera $(I \rightarrow \text{bool}) \rightarrow \text{bool}$.

On notera $u : A$ pour exprimer le fait que u est un terme de type A . Pour former les termes du calcul d’un type donné, on dispose des opérations suivantes. Si $t : A \rightarrow B$ et $u : A$, on peut *appliquer* t à u et obtenir le terme $t(u)$ de type B . Si t est de type $A_1 \rightarrow (A_2 \rightarrow B)$ on notera $t(u_1, u_2)$ pour $(t(u_1))(u_2)$. Si t est un terme de type B , qui peut faire référence à la variable x de type A , on peut introduire une fonction f de type $A \rightarrow B$ et définie par le schéma $f(x) = t$. On a alors $f(u) = (u/x)t$ si u est un terme de type A et $(u/x)t$ dénote l’opération de substitution de la variable x par u dans le terme t .

Les *propositions* sont les termes de type *bool*, qui sont formés à partir des connecteurs usuels, constantes $\vee, \wedge, \rightarrow$ ⁽²⁾ de type $\text{bool} \rightarrow (\text{bool} \rightarrow \text{bool})$ et $\neg : \text{bool} \rightarrow \text{bool}$, et des

2. Nous utilisons le même symbole \rightarrow pour l’implication logique et le type des fonctions, et ces deux notions seront identifiées dans la théorie des types dépendants.

quantificateurs \forall, \exists , constantes de type $(A \rightarrow \text{bool}) \rightarrow \text{bool}$. La différence avec le calcul des prédicats ordinaire est que l'on peut quantifier sur des fonctions, fonctionnelles, ... En particulier, si t et u sont deux termes de type A on peut former

$$\text{Eq}_A(t, u) = (\forall P: A \rightarrow \text{bool})(P(t) \rightarrow P(u))$$

qui est un terme de type bool (une proposition) exprimant que toute propriété de t est aussi vérifiée pour u . Ceci définit de manière interne la relation d'égalité.

Le système comporte finalement des *règles de déduction* qui permettent d'affirmer qu'une proposition est prouvable. La version utilisée par T.C. Hales [Ha] utilise les lois de la logique classique. Ce système a une sémantique naturelle dans la théorie des ensembles en interprétant bool par l'ensemble $\{0, 1\}$ et le type des individus I par un ensemble quelconque. Il existe aussi des versions intuitionnistes qui correspondent exactement à la logique des topos élémentaires [Ca] et où bool correspond à l'objet des valeurs de vérités du topos.

1.2. Axiomes d'extensionnalité

L'*axiome d'extensionnalité* est le premier axiome de la théorie des ensembles [Bou1]. Il énonce que deux ensembles sont égaux dès qu'ils ont les mêmes éléments. Dans le système de Church, cet axiome se présente sous deux formes indépendantes. L'axiome d'extensionnalité pour les *fonctions* déclare que deux fonctions qui sont égales en tout point sont égales

$$((\forall x: A)\text{Eq}_B(f(x), g(x))) \rightarrow \text{Eq}_{A \rightarrow B}(f, g).$$

L'axiome d'extensionnalité pour les *propositions* énoncent que deux propositions qui sont équivalentes sont égales

$$((P \rightarrow Q) \wedge (Q \rightarrow P)) \rightarrow \text{Eq}_{\text{bool}}(P, Q).$$

L'axiome d'univalence est une généralisation de ce deuxième énoncé et, ainsi généralisé, il entraîne l'axiome d'extensionnalité pour les fonctions.

1.3. Opérateur de description

Le système de Church contient deux représentations distinctes de la notion de *fonction*. La première représentation est celle de fonction comme donnée par une définition explicite $f(x) = t$, qui est apparentée à la notion de programme. (Le système de Church est en fait à la base des langages de programmation fonctionnelle [Tu].) La deuxième représentation est celle de fonction comme graphe fonctionnel [Bou1], qui consiste en une relation $R: A \rightarrow (B \rightarrow \text{bool})$ satisfaisant la propriété

$$(\forall x: A)(\exists! y: B)R(x, y),$$

où $(\exists! y: B)Q(y)$ est définie par

$$(\exists y: B)(Q(y) \wedge (\forall y': B)(Q(y') \rightarrow \text{Eq}_B(y, y'))).$$

L’axiome de description de Church permet de montrer que ces deux notions de fonction sont en fait équivalentes. Il consiste à ajouter un opérateur $\iota : (A \rightarrow \text{bool}) \rightarrow A$ qui vérifie

$$((\exists! y : B)Q(y)) \rightarrow Q(\iota(Q)).$$

Cette opération et cet axiome correspondent à la pratique mathématique de pouvoir donner un nom à un objet mathématique si celui-ci est caractérisé de manière unique. Si R est un graphe fonctionnel, on peut alors définir la fonction correspondante par l’équation $f(x) = \iota(Q)$ avec $Q(y) = R(x, y)$.

Si l’on oublie la condition d’unicité on obtient l’opérateur de choix de Hilbert [Hi] $\epsilon : (A \rightarrow \text{bool}) \rightarrow A$ (noté τ dans [Bou1], pour éviter la confusion avec le symbole d’appartenance, et qui est présent dans le système HOL [Ha]) qui vérifie

$$((\exists y : B)\psi(y)) \rightarrow \psi(\epsilon(\psi)).$$

Dans le système de Voevodsky l’axiome de description devient *prouvable* et on peut montrer la *négation* de l’axiome du choix global.

2. THÉORIE DES TYPES DÉPENDANTS

La logique d’ordre supérieur, bien qu’élégante, a des limitations artificielles pour la représentation des objets mathématiques. Il n’est par exemple pas possible d’exprimer dans ce cadre la notion de structure algébrique sur un type quelconque. Ce problème est résolu dans le système que nous présentons maintenant. Un autre aspect intéressant de la théorie des types dépendants est qu’elle utilise *uniquement* la notion de déclaration de types, sans nécessiter la formulation de règles logiques. L’idée va être d’utiliser la structure plus riche des types pour représenter les propositions par des types. Ceci rend cette approche appropriée pour une représentation formelle des preuves (cf. par exemple [Go, FT, Br]).

2.1. Types dépendants et contextes

Chaque objet mathématique a est objet d’un certain type A . On notera $a : A$ pour exprimer que l’objet a est de type A . Les types sont eux-mêmes des objets mathématiques, qui ont pour type des *univers*. La notion fondamentale de ce calcul est celle de *famille de types* $B(x)$, $x : A$. Intuitivement $B(x)$ représente un *type dépendant* qui peut dépendre d’un objet x de type A . Si $a : A$ alors $B(a)$ est un type. Il est pertinent, comme l’a remarqué N.G. de Bruijn [Br] de considérer les propositions comme des types, et, dans ce cas, de lire $a : A$ comme exprimant que l’objet a est une *preuve* de la proposition A . Si $B(x)$, $x : A$ est une famille de types et $B(x)$ représente une proposition, alors $B(a)$ représente une propriété de l’élément a de type A . On peut alors introduire la notion de *contexte* [Br]

$$x_1 : A_1, x_2 : A_2(x_1), \dots, x_n : A_n(x_1, \dots, x_{n-1})$$

qui représente formellement la liste d'hypothèse et de notions introduites au cours d'un raisonnement. Par exemple, le texte «soient x un nombre entier, satisfaisant la condition $P(x)$, et y un autre nombre en relation $R(x, y)$ avec x » correspondra à un contexte de la forme, si \mathbf{N} est le type des nombres entiers

$$x:\mathbf{N}, p:P(x), y:\mathbf{N}, q:R(x, y).$$

2.2. Types produits et types sommes

Si $B(x)$, $x:A$ est une famille de types sur A on peut former le type $(\Pi x:A)B(x)$ des *sections*. Si b est un objet de type $B(x)$ dépendant de l'hypothèse $x:A$ on peut définir un objet f de type $(\Pi x:A)B(x)$ par l'équation $f(x) = b$. Si $a:A$ l'objet $f(a)$ est de type $B(a)$ et on a $f(a) = (a/x)b$. Si $B(x) = B$ ne dépend pas de $x:A$ on écrira aussi $A \rightarrow B$ pour $(\Pi x:A)B$. Si A et B représentent des propositions, ceci codera l'opération d'implication.

De manière duale, on peut former le type $(\Sigma x:A)B(x)$. Un élément de ce type est un couple (a, b) avec $a:A$ et $b:B(a)$. Si c est un élément de $(\Sigma x:A)B(x)$ on peut former les deux projections $\mathbf{p}(c):A$ et $\mathbf{q}(c):B(\mathbf{p}(c))$. Si $a:A$ et $b:B(a)$ on aura les équations $\mathbf{p}((a, b)) = a$ et $\mathbf{q}((a, b)) = b$. Si $B(x) = B$ ne dépend pas de $x:A$ on écrira aussi $A \times B$ pour $(\Sigma x:A)B$. Si A et B représentent des propositions, ceci codera l'opération de conjonction.

2.3. Représentation des règles logiques

Décrivons les preuves des deux premiers axiomes pour l'implication tels qu'ils apparaissent dans [Hi] pour illustrer l'idée de de Bruijn [Br] et Howard [How] de représenter les propositions comme type. La preuve du premier axiome

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

consiste en la fonction f définie par l'équation $f(x, y) = x$. Le deuxième axiome

$$(B \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$$

est prouvé en définissant $g(x, y, z) = x(y(z))$, qui peut être aussi vu comme l'opération de composition. L'opération d'application justifie le *modus-ponens*, qui permet de déduire B à partir de $A \rightarrow B$ et de A .

Si $B(x)$ représente une proposition, on pourra considérer que $(\Pi x:A)B(x)$ représente $(\forall x:A)B$ et que $(\Sigma x:A)B(x)$ représente $(\exists x:A)B(x)$ [How, M-L]. Ces analogies seront précisées plus loin.

Howard, puis Martin-Löf [How, M-L], ont complété cette représentation des opérations logiques en ajoutant d'autres opérations primitives sur les types. Ceci permet de représenter concrètement les idées de Gentzen [Ge] sur la «déduction naturelle».

La *somme disjointe* de deux types $A + B$ est définie par les règles d'introduction : si a est de type A (resp. b de type B) alors $\text{inl}(a) : A + B$ (resp. $\text{inr}(b) : A + B$). On peut définir une fonction $h : (\Pi z : A + B)C(z)$ par cas

$$h(\text{inl}(x)) = f(x) \quad h(\text{inr}(y)) = g(y)$$

étant données deux fonctions $f : (\Pi x : A)C(\text{inl}(x))$ et $g : (\Pi y : B)C(\text{inr}(y))$. Si $C(z)$ ne dépend pas de z on retrouve la règle usuelle logique pour $A \vee B$, telle que formulée par Gentzen [Ge], qui permet de dériver $(A \vee B) \rightarrow C$ à partir de $A \rightarrow C$ et $B \rightarrow C$.

De manière similaire, on pourra définir le type des entiers \mathbf{N} par les règles d'introduction : 0 est de type \mathbf{N} et $n + 1$ est de type \mathbf{N} si $n : \mathbf{N}$. On peut définir une fonction $f : (\Pi x : \mathbf{N})C(x)$ par les équations

$$f(0) = a \quad f(n + 1) = g(n, f(n))$$

si on suppose donnés $a : C(0)$ et $g : (\Pi n : \mathbf{N})(C(n) \rightarrow C(n + 1))$. Ceci exprime à la fois le principe de définition d'une fonction par *réursion primitive* (pour la première fois mise en évidence par Hilbert [Hi]) dans le cas où $C(x)$ ne dépend pas de x et le principe de preuve par *induction* dans le cas où $C(x)$ représente une proposition.

Le type des booléens \mathbf{N}_2 sera défini par les règles d'introduction : 0 et 1 sont de type \mathbf{N}_2 . Si $C(z)$, $z : \mathbf{N}_2$ est une famille de types on peut définir une fonction $f : (\Pi z : \mathbf{N}_2)C(z)$ par les équations $f(0) = a_0$ et $f(1) = a_1$ étant donnés $a_0 : C(0)$ et $a_1 : C(1)$.

Finalement, le type vide \mathbf{N}_0 sera défini comme n'ayant aucune règle d'introduction. Si $C(z)$, $z : \mathbf{N}_0$ est une famille de types on peut définir une fonction $f : (\Pi z : \mathbf{N}_0)C(z)$.

Ceci raffine la règle «ex falso quodlibet», qui énonce seulement $\mathbf{N}_0 \rightarrow T$ pour une proposition T . Ce raffinement a été introduit par Martin-Löf [M-L] et il semble avoir été vraiment utilisé pour la première fois par Voevodsky (il joue un rôle crucial pour montrer que \mathbf{N}_0 est une proposition au sens de la définition 4.1). Le type vide représente la proposition fausse $\perp = \mathbf{N}_0$. On peut définir l'opération de négation par $\neg A = A \rightarrow \perp$.

2.4. Conversion et normalisation

L'égalité $f(u) = (u/x)t$ peut être vue comme une *réduction* $f(u) \rightarrow (u/x)t$ qui consiste à remplacer $f(u)$ par sa définition. Un terme est en *forme normale* si on ne peut pas effectuer de telles réductions dans ce terme. Deux termes a et b sont *convertibles* s'ils ont des formes normales identiques. Par exemple, si on définit $f : \mathbf{N} \rightarrow \mathbf{N}$ par $f(0) = 0$ et $f(n + 1) = f(n)$ alors les termes $f(0 + 1)$ et 0 ont pour forme normale 0 et sont donc convertibles. On peut montrer de manière métamathématique la propriété de *normalisation*, que tout terme a une forme normale. Il en résulte que la relation de conversion de deux termes est *décidable* (il suffit en effet de comparer les formes normales). Un autre corollaire moins direct est, qu'étant donnés deux termes a and A , il est possible de décider algorithmiquement si on a $a : A$ ou non. En particulier, comme on représente les propositions comme des types, il sera possible de décider si un élément est bien une *preuve* d'une proposition donnée. Cette propriété justifie l'utilisation de

ce formalisme pour la vérification automatique de preuves. Par contre, le problème de décider si un type contient au moins un élément n'est en général *pas* décidable.

Il est possible de caractériser de manière purement syntaxique les formes normales de type \mathbf{N} : ce sont exactement les termes de la forme $(\dots((0+1)+1)\dots)+1$. Les formes normales de type \mathbf{N}_2 sont exactement 0 et 1. Il n'y a pas de terme en forme normale de type \mathbf{N}_0 . On voit ainsi de manière combinatoire que la propriété de normalisation entraîne la *cohérence* : il n'y a pas de preuve de la proposition $\perp = \mathbf{N}_0$. En effet, un tel terme devrait avoir une forme normale, et il n'y a pas de forme normale de type \mathbf{N}_0 . (D'après le théorème de Gödel toutefois, la propriété de normalisation elle-même ne peut pas être établie de manière purement combinatoire.)

Notons qu'il n'est pas possible d'exprimer dans ce système qu'un élément a n'est *pas* élément d'un type A ou que deux types A et B ne sont *pas* convertibles. Les relations de convertibilité et de typage sont des relations «externes» qui ne sont pas elles-mêmes représentées de manière interne par des types.

2.5. Univers

Le type des propositions doit être un type \mathbf{U} dont les éléments sont eux-mêmes des types. Une suggestion serait d'introduire un type de tous les types, qui vérifierait $\mathbf{U} : \mathbf{U}$, comme l'a d'abord envisagé Martin-Löf [M-L]. On pense bien sûr au paradoxe de Russell, mais il ne s'applique pas directement car on ne peut pas former $(\Sigma X : \mathbf{U}) \neg (X : X)$ puisque $X : X$ n'est pas un type. Toutefois, Girard [Gi] a pu montrer que ce système avec $\mathbf{U} : \mathbf{U}$ est contradictoire. Le type \perp , et donc tout type, devient prouvable, ce qui veut dire que l'on peut construire un terme sans variables libres de type $\mathbf{N}_0 = \perp$. L'idée est de reproduire le paradoxe de Burali-Forti au moyen de \mathbf{U} et du type somme pour coder le type de *tous* les ordres bien fondés. (Comme expliqué plus haut, un tel terme de type \mathbf{N}_0 ne peut pas avoir de forme normale.)

Pour contourner ce problème, Martin-Löf [M-L] introduit une hiérarchie d'univers, par analogie avec la notion d'univers introduite par Grothendieck [Un]. On se donne une suite de types $\mathbf{U}_n : \mathbf{U}_{n+1}$ indexée (de manière *externe*) par les entiers, et telle que $A : \mathbf{U}_{n+1}$ si $A : \mathbf{U}_n$. Les types $\mathbf{N}, \mathbf{N}_2, \mathbf{N}_0$ sont de type \mathbf{U}_0 . De plus $(\Pi x : A)B(x)$ (resp. $(\Sigma x : A)B(x)$) est de type \mathbf{U}_n si A et $B(x)$ sont de type \mathbf{U}_n et $C + D$ est de type \mathbf{U}_n si C et D sont de type \mathbf{U}_n . Dans ce système, un terme A est un type si, et seulement si, $A : \mathbf{U}_n$ pour un certain n . La référence [M-L] contient aussi une preuve de normalisation, et donc de cohérence de ce système. Dans la suite du texte, \mathbf{U} dénotera un des univers \mathbf{U}_n .

En utilisant la notion d'univers et les sommes, on peut représenter le type des structures pour un univers donné. Par exemple le type des types munis d'une opération binaire pour un univers \mathbf{U} sera $(\Sigma X : \mathbf{U})(X \rightarrow X \rightarrow X)$. Un élément de ce type est un couple X, f avec $f : X \rightarrow X \rightarrow X$. Pour pouvoir représenter des structures algébriques on doit pouvoir exprimer l'égalité de manière interne et donc introduire un type qui représente l'égalité.

2.6. Égalité

Pour chaque type A , et éléments a_0, a_1 de type A on introduit un nouveau *type* $\mathbf{Eq}_A(a_0, a_1)$. Remarquons que, comme $\mathbf{Eq}_A(a_0, a_1)$ est lui-même un type, il est possible d'itérer cette construction et de former $\mathbf{Eq}_{\mathbf{Eq}_A(a_0, a_1)}(p, q)$ dès que p et q sont de type $\mathbf{Eq}_A(a_0, a_1)$. Une telle itération n'est pas possible dans une formulation usuelle de la logique, qui distingue les types et les propositions. C'est précisément cette itération qui contient en germe la connection avec la notion de ∞ -groupeïde.

Quelles doivent être les lois de ce type représentant l'égalité? On doit exprimer que tout élément est égal à lui-même, et que deux éléments égaux vérifient les mêmes propriétés. On introduit donc un terme $1_a : \mathbf{Eq}_A(a, a)$ pour tout $a : A$ et une opération

$$\mathbf{transp}_B : B(a) \rightarrow (\mathbf{Eq}_A(a, u) \rightarrow B(u))$$

pour toute famille de types $B(x)$, $x : A$. Il est naturel d'imposer aussi la condition $\mathbf{Eq}_{B(a)}(\mathbf{transp}(v, 1_a), v)$ si $v : B(a)$.

En utilisant les univers, on peut montrer par exemple que l'on a $\neg(\mathbf{Eq}_N(0, 1))$. Il suffit pour cela de définir $C : N \rightarrow \mathbf{U}_0$ par les équations $C(0) = N$ et $C(n+1) = \perp$. À partir de $\mathbf{Eq}_N(0, 1)$ et de l'opération \mathbf{transp}_C on peut déduire $C(0) \rightarrow C(1)$ et donc \perp puisque $C(0) = N$ et $C(1) = \perp$.

2.7. Groupeïde

Martin-Löf introduit dans [M-L] une loi nouvelle pour l'égalité. On peut énoncer cette loi de la manière suivante.

FAIT 2.1. — *Pour tous éléments a et x de type A et pour tout élément p de type $\mathbf{Eq}_A(a, x)$, on a*

$$\mathbf{Eq}_{(\Sigma x:A)\mathbf{Eq}_A(a,x)}((a, 1_a), (x, p)).$$

La raison pour Martin-Löf d'introduire cette loi est purement formelle. Sous cette forme, la règle d'élimination de l'égalité devient

$$C(a, 1_a) \rightarrow (\Pi p : \mathbf{Eq}_A(a, x))C(x, p),$$

ce qui renforce $B(a) \rightarrow \mathbf{Eq}_A(a, x) \rightarrow B(x)$ de la même manière que la règle

$$(\Pi x : N_0)C(x)$$

renforce la loi d'élimination $\perp \rightarrow A$ [M-L].

Si A est un type ayant pour éléments a_0, a_1, a_2 et $p : \mathbf{Eq}_A(a_0, a_1)$ et $q : \mathbf{Eq}_A(a_1, a_2)$ on définit $p \cdot q : \mathbf{Eq}_A(a_0, a_2)$ par $p \cdot q = \mathbf{transp}_C(p, q)$ où $C(x) = \mathbf{Eq}_A(a_0, x)$. On obtient donc de manière purement logique une opération de composition sur les preuves d'égalité. Si $f : A \rightarrow B$ on voit aussi que l'on a $\mathbf{transp}_C(1_{f(a_0)}, p) : \mathbf{Eq}_B(f(a_0), f(a_1))$ si $p : \mathbf{Eq}_A(a_0, a_1)$ et $C(x) = \mathbf{Eq}_B(f(a_0), f(x))$.

Une conséquence remarquable de la loi 2.1 est le résultat suivant.

THÉORÈME 2.2 ([HS]). — *Tout type vérifie les lois de groupeïde en considérant les éléments de A comme objets et les preuves d'égalité comme morphismes.*

Par exemple, pour prouver $\mathbf{Eq}_{\mathbf{Eq}_A(a_0, a_3)}(p \cdot (q \cdot r), (p \cdot q) \cdot r)$ étant donnés

$$p : \mathbf{Eq}_A(a_0, a_1), q : \mathbf{Eq}_A(a_1, a_2), r : \mathbf{Eq}_A(a_2, a_3)$$

on utilise la loi 2.1 pour se ramener au cas où p, q, r sont de la forme $1_a : \mathbf{Eq}_A(a, a)$, et le fait que l'on a directement $\mathbf{Eq}_{\mathbf{Eq}_A(a, a)}(1_a \cdot 1_a, 1_a)$.

Comme on peut itérer la formation du type d'égalité, le théorème 2.2 suggère en fait que tout type est automatiquement muni d'une structure de ∞ -groupeïde. Ceci a été conjecturé par Hofmann et Streicher [HS] et a été vérifié par Lumsdaime [Lum].

Ce résultat illustre en quel sens la notion de ∞ -groupeïde est plus fondamentale dans cette approche que la notion de catégorie : tout type a automatiquement une structure de ∞ -groupeïde, alors que la notion de catégorie sera introduite comme une *structure* sur un groupeïde donné.

3. SÉMANTIQUE

Arrivé à ce point, le formalisme suggère qu'une sémantique possible doit pouvoir être d'interpréter un type par un ∞ -groupeïde, et donc, au vu de la correspondance suggérée par Grothendieck, par un type d'homotopie [AW, Voe]. On obtient effectivement ainsi un modèle de la théorie des types où un type est interprété par un ensemble simplicial qui vérifie la propriété de Kan [KLV]. (Une variation effective de ce modèle qui utilise des ensembles cubiques est décrite dans la référence [BCH].) Une famille de types $B(x)$, $x : A$ sera interprétée par une *fibration* $E \rightarrow A$ au sens de Kan. Le type $(\Sigma x : A)B(x)$ sera alors l'espace E lui-même, tandis que le type $(\Pi x : A)B(x)$ sera interprété par l'espace des *sections* de l'application $E \rightarrow A$, espace qui vérifie bien la condition de Kan [KLV].

Le type des égalités $\mathbf{Eq}_A(x_0, x_1)$ vu comme une famille de types indexée par $x_0 : A, x_1 : A$ sera interprété par l'espace des *chemins*. Si $E \rightarrow A$ est une fibration de Kan, et a_0, a_1 des points de A , tout chemin entre a_0 et a_1 définit bien une application entre les fibres $E(a_0) \rightarrow E(a_1)$, ce qui permet d'interpréter la règle d'élimination de l'égalité.

Considérons l'interprétation dans ce modèle de la loi 2.1, formulée par Martin-Löf [M-L]. Elle correspond au fait que la fibration obtenue en considérant l'espace des chemins qui partent d'un point donné a d'un espace A , et associant à un chemin son point terminal, définit un espace total qui est contractile. C'est un des points de départ de l'utilisation de l'espace des chemins en topologie algébrique [Se], et il est assez remarquable qu'il corresponde ainsi à une propriété fondamentale de l'égalité dans la théorie des types.

4. L'AXIOME D'UNIVALENCE

4.1. Stratification

L'étape suivante, due à Voevodsky et motivée par la sémantique que l'on vient de décrire, est d'introduire une stratification des types, définie de manière purement logique.

DÉFINITION 4.1. — *Le type*

$$\mathbf{prop}(A) = (\Pi x_0 : A)(\Pi x_1 : A)\mathbf{Eq}_A(x_0, x_1)$$

exprime que le type A est une proposition.

Le type

$$\mathbf{set}(A) = (\Pi x_0 : A)(\Pi x_1 : A)\mathbf{prop}(\mathbf{Eq}_A(x_0, x_1))$$

exprime que le type A est un ensemble.

Le type

$$(\Pi x_0 : A)(\Pi x_1 : A)\mathbf{set}(\mathbf{Eq}_A(x_0, x_1))$$

exprime que le type A est un groupoïde.

Les notions de *propositions*, d'*ensembles* et de *groupoïdes* ont ainsi acquis une signification formelle précise et seront désormais, sauf mention explicite du contraire, utilisées uniquement en ce sens.

Un type est une proposition s'il a au plus un élément. Le type $\perp = \mathbf{N}_0$ est par exemple une proposition. Ceci résulte directement du fait que l'on a $(\Pi x : \mathbf{N}_0)C(x)$ pour toute famille de types $C(x)$, $x : \mathbf{N}_0$.

Le type $\mathbf{contr}(A) = \mathbf{prop}(A) \times A$ exprime que le type A est *contractile*.

DÉFINITION 4.2 ([Voe]). — *A est de niveau 0 si A est contractile, et A est de niveau $n + 1$ si $\mathbf{Eq}_A(a_0, a_1)$ est de niveau n pour tous a_0, a_1 de type A .*

PROPOSITION 4.3. — *Si $\mathbf{prop}(A)$ et a_0, a_1 sont de type A alors $\mathbf{prop}(\mathbf{Eq}_A(a_0, a_1))$. Il s'ensuit que tout type de niveau n est aussi de niveau $n + 1$.*

Un type A a une égalité décidable si on a $(\Pi x : A)(\Pi y : A)\mathbf{Eq}_A(x, y) + \neg\mathbf{Eq}_A(x, y)$. Il est direct de montrer que les types \mathbf{N} et \mathbf{N}_2 ont une égalité décidable.

THÉORÈME 4.4 ([He]). — *Si A a une égalité décidable alors A est un ensemble.*

On en déduit que les types \mathbf{N} et \mathbf{N}_2 sont des *ensembles* qui ne sont pas des propositions. Ce résultat est un des points de départ de la formalisation présentée dans [Go].

Si $B(x)$, $x : T$ alors le type $(\Pi x : T)\mathbf{prop}(B(x))$ exprime que $B(x)$ est une *propriété* de x sur T .

PROPOSITION 4.5. — *Le fait pour un type d'être d'un niveau donné n est une propriété de ce type.*

PROPOSITION 4.6. — Si $B(x)$ est une famille de types de niveau n indexée au-dessus d'un type A de niveau n alors $(\Sigma x : A)B(x)$ est de niveau n .

COROLLAIRE 4.7. — $A \times B$ est une proposition dès que A est une proposition et que l'on a $A \rightarrow \text{prop}(B)$.

4.2. Équivalence et l'axiome d'univalence

DÉFINITION 4.8. — Si $f : A \rightarrow B$ et $b : B$ la fibre de f en b est le type

$$F_f(b) = (\Sigma x : A)\text{Eq}_B(f(x), b).$$

La fonction f est une équivalence si $F_f(b)$ est contractile pour tout $b : B$

$$\text{isEquiv}(f) = (\Pi b : B)\text{contr}(F_f(b)).$$

On notera $A \simeq B$ pour $(\Sigma f : A \rightarrow B)\text{isEquiv}(f)$.

Cette définition généralise de manière uniforme les notions suivantes

- l'équivalence logique au niveau des propositions
- la notion de bijection au niveau des ensembles
- la notion d'équivalence catégorique au niveau des groupoïdes, et ainsi de suite.

Par le fait 2.1 la fonction identité est une équivalence. On a donc $A \simeq A$ pour tout type A et il en résulte que l'on a une fonction canonique

$$\text{Eq}_U(A, B) \rightarrow A \simeq B.$$

L'axiome d'univalence exprime que cette fonction est elle-même une équivalence.

4.3. Quelques conséquences de l'axiome d'univalence

On dit que $g : B \rightarrow A$ est inverse de $f : A \rightarrow B$ si on a $(\Pi x : A)\text{Eq}_A(g(f(x)), x)$ et $(\Pi y : B)\text{Eq}_B(f(g(y)), y)$.

LEMME 4.9. — Si $f : A \rightarrow B$ a un inverse, alors f est une équivalence.

La fonction $f : \mathbb{N}_2 \rightarrow \mathbb{N}_2$ définie par les équations $f(0) = 1$ et $f(1) = 0$ est son propre inverse. Par le lemme 4.9, f est donc une équivalence. Elle définit une preuve d'égalité $\text{Eq}_{\mathbb{U}_0}(\mathbb{N}_2, \mathbb{N}_2)$ qui n'est pas triviale et donc ce type $\text{Eq}_{\mathbb{U}_0}(\mathbb{N}_2, \mathbb{N}_2)$ n'est pas une proposition. On en déduit que \mathbb{U}_0 n'est pas un ensemble. Le type \mathbb{U}_0 est donc au moins un groupoïde. On peut montrer de même que \mathbb{U}_1 n'est pas un groupoïde, que \mathbb{U}_2 n'est pas un 2-groupoïde, et, en général, que \mathbb{U}_n est au moins de niveau $n + 3$ [KS].

Il résulte de cette discussion que l'on a une preuve de l'égalité

$$\text{Eq}_{(\Sigma X : \mathbb{U})X}((\mathbb{N}_2, 0), (\mathbb{N}_2, 1)),$$

ce qui montre que l'on peut avoir

$$\text{Eq}_{(\Sigma x : A)B(x)}((a, b_0), (a, b_1))$$

et

$$\neg(\text{Eq}_{B(a)}(b_0, b_1)).$$

Ceci peut paraître étrange mais est en fait clair sémantiquement puisque le groupe $\text{Eq}_A(a, a)$ peut avoir une action non triviale sur la fibre $B(a)$. Ce phénomène ne peut pas se produire si A est un ensemble, en particulier si A a une égalité décidable. Ceci explique a posteriori pourquoi certaines formalisations de preuves dans la théorie des types [Go, FT] se limitaient à considérer des types à égalité décidables, qui se comportent suivant l'intuition ensembliste.

De même la fonction $A \times B \rightarrow B \times A$, $(a, b) \mapsto (b, a)$ est son propre inverse et est une équivalence. On a donc $\text{Eq}_U(A \times B, B \times A)$. On peut aussi montrer

$$\text{Eq}_U(A \times (B \times C), (A \times B) \times C).$$

Les égalités correspondantes ne sont en général pas valables dans la théorie des ensembles. Il est intéressant de comparer ce qui se passe ici avec la discussion dans [Bou4], p. 68-69. Il faut aussi noter que $A \times B$ et $B \times A$ ne sont en général *pas* convertibles.

Si A et B sont des propositions, il résulte du lemme 4.9 qu'une application $A \rightarrow B$ est une équivalence dès que l'on a aussi $B \rightarrow A$. On voit ainsi que l'axiome d'univalence est une *généralisation* à tous les types de l'axiome d'extensionnalité de Church (deux propositions équivalentes sont égales).

THÉORÈME 4.10 ([Voe]). — *L'axiome d'univalence entraîne le principe d'extensionnalité pour les fonctions.*

COROLLAIRE 4.11. — *Si $B(x)$ est une famille de types de niveau n indexée au-dessus d'un type A arbitraire alors $(\Pi x : A)B(x)$ est de niveau n .*

En particulier un produit d'une famille de propositions est une proposition.

5. REPRÉSENTATION FORMELLE DE QUELQUES NOTIONS MATHÉMATIQUES

5.1. Structures algébriques

Les structures algébriques sont représentées par des ensembles. Par exemple le type de tous les *groupes* pour un univers U sera

$$\text{Grp} = (\Sigma X : U)(\Sigma f : X \times X \rightarrow X)(\Sigma g : X \rightarrow X)(\Sigma e : X)(\text{set}(X) \times P(X, f, g, e))$$

où $P(X, f, g, e)$ désigne la conjonction des axiomes de groupe

$$P(X, f, g, e) = (\Pi x : X)\text{Eq}_X(f(x, e), x) \times (\Pi x : X)\text{Eq}_X(f(x, g(x)), e) \times \dots$$

En utilisant la proposition 4.5 et le corollaire 4.7, on voit que $\text{set}(X) \times P(X, f, g, e)$ est une *proposition* pour tout type X et que le type Grp est un *groupoïde*.

Si on note

$$G(X) = (\Sigma f : X \times X \rightarrow X)(\Sigma g : X \rightarrow X)(\Sigma e : X)(\text{set}(X) \times P(X, f, g, e))$$

on peut écrire $\mathbf{Grp} = (\Sigma X : \mathbf{U})G(X)$ et $G(X)$, qui est toujours un *ensemble*, représente le type des *structures de groupe* sur le type $X : \mathbf{U}$.

Il est essentiel dans cette approche de distinguer la notion de *propriété*, comme $\mathbf{set}(X) \times P(X, f, g, e)$ et de *structure*, comme $G(X)$. Par exemple, «être équivalent» $A \simeq B$ est en général une structure sur A et B , alors que le type $\mathbf{isEquiv}(f)$ qui exprime qu’une application donnée f est une équivalence est une propriété de f .

On peut de même définir le groupoïde des *ensembles ordonnés* : on doit se donner un *ensemble* X , muni d’une famille $R(x, y)$ de *propositions*, qui vérifient les conditions usuelles de réflexivité et de transitivité. On peut énoncer la condition d’antisymétrie comme le fait que l’implication suivante entre deux propositions (qui résulte de la réflexivité)

$$\mathbf{Eq}_X(x, y) \rightarrow R(x, y) \times R(y, x)$$

est en fait une équivalence logique.

Comme un produit d’ensembles est toujours un ensemble par le corollaire 4.11, si I est un type *quelconque* et $A(i)$, $i : I$ une famille d’anneaux par exemple, alors $(\Pi i : I)A(i)$ est un ensemble par le corollaire 4.11 et on peut munir cet ensemble d’une structure d’anneau. L’interprétation de cette construction est directe si I est lui-même un ensemble, mais plus subtile si I est de niveau plus élevé. Si par exemple I est un groupoïde avec un seul point 0 et G est le groupe $\mathbf{Eq}_I(0, 0)$, se donner $A(i)$, $i : I$ revient à se donner un anneau $A = A(0)$ muni d’une action du groupe G , et $(\Pi i : I)A(i)$ est en fait l’anneau A^G des éléments invariants par cette action.

5.2. Transports de structures

En général, une notion de *structures*, telles que celle décrite dans [Bou4], sera donnée par une famille d’ensembles $T(X)$, $X : \mathbf{U}$ telle que $T(X) \rightarrow \mathbf{set}(X)$. Le type de toutes ces structures $(\Sigma X : \mathbf{U})T(X)$ est alors un *groupoïde* et un élément de ce type est un couple (X, u) où $u : T(X)$. Si $f : X \rightarrow Y$ est une bijection entre les ensembles X et Y de type \mathbf{U} , l’axiome d’univalence associée à f une preuve d’égalité $\mathbf{Eq}_\mathbf{U}(X, Y)$ et donc une fonction $T(X) \rightarrow T(Y)$. On obtient ainsi une fonction de *transport de structures* [Bou4]. Deux structures (X, u) et (Y, v) sont *isomorphes* si (Y, v) est égale à la structure obtenue par transport de structures au moyen d’une bijection $X \rightarrow Y$. On voit alors que deux structures isomorphes sont *égales*. En particulier, toute *propriété* valide pour une structure donnée est automatiquement valide pour une structure isomorphe, et l’on n’a pas besoin dans ce cadre de «critères de transportabilité» [Bou4].

5.3. Univers : axiome de redimensionnement

Une intuition ensembliste pour les univers est qu’ils permettent de représenter des collections de plus en plus grandes. Le résultat que \mathbf{U}_n est au moins de niveau $n + 3$ [KS] montre qu’un univers «grand» suivant l’intuition ensembliste a nécessairement une égalité «complexe». Les *axiomes de redimensionnement* suivants, suggérés par Voevodsky [Voe], expriment que les propositions sont des types «petits».

Le premier axiome énonce que, si A est un type arbitraire qui est une proposition, alors on a $A : \mathbf{U}_0$.

Le deuxième axiome déclare que

$$\Omega = (\Sigma X : \mathbf{U}_0)\mathbf{prop}(X)$$

qui est a priori de type \mathbf{U}_1 , est aussi de type \mathbf{U}_0 .

(On peut également considérer l'axiome que si l'on a $A : \mathbf{U}_n$ et $B : \mathbf{U}_m$ avec $m < n$ et $\mathbf{Eq}_{\mathbf{U}_n}(A, B)$, alors A est aussi de type \mathbf{U}_m .)

Il est possible de montrer sans ces axiomes que le type $(\Sigma X : \mathbf{U})\mathbf{prop}(X)$ est toujours un ensemble qui joue le rôle de l'objet des valeurs de vérités. En utilisant les axiomes de redimensionnement, on montre que les *ensembles* d'un univers donné forment un *topos élémentaire* [Ca].

Voevodsky utilise ces axiomes pour définir la modalité

$$\mathbf{inh}(A) = (\Pi X : \mathbf{U}_0)(\mathbf{prop}(X) \rightarrow ((A \rightarrow X) \rightarrow X))$$

de telle sorte que $\mathbf{inh}(A)$ soit de type \mathbf{U}_0 . Par le corollaire 4.11, le type $\mathbf{inh}(A)$ est une proposition pour un type quelconque A , et par construction, on a $\mathbf{inh}(A) \rightarrow B$ dès que B est une proposition qui est impliquée par A . Intuitivement, cette proposition $\mathbf{inh}(A)$ exprime que le type A est non vide, sans toutefois indiquer explicitement un élément de type A .

Cette modalité permet d'introduire les opérations logiques

$$A \vee B = \mathbf{inh}(A + B) \quad (\exists x : A)B(x) = \mathbf{inh}((\Sigma x : A)B(x))$$

qui définissent des propositions. On vérifiera par exemple que si A et B sont des propositions *incompatibles*, i.e. $\neg(A \wedge B)$, alors on a $\mathbf{Eq}_{\mathbf{U}_0}(A + B, A \vee B)$.

Bien qu'un produit d'ensembles soit toujours un ensemble, il n'est pas possible d'avoir un axiome de redimensionnement qui dirait qu'un ensemble est toujours dans \mathbf{U}_0 . En effet, le type suivant

$$A = (\Pi X : \mathbf{U}_0)(\mathbf{set}(X) \rightarrow ((T(X) \rightarrow X) \rightarrow X))$$

avec $T(X) = (X \rightarrow \Omega) \rightarrow \Omega$, est un ensemble, par le corollaire 4.11, qui ne peut pas être dans \mathbf{U}_0 . On aurait en effet dans ce cas une bijection entre A et $(A \rightarrow \Omega) \rightarrow \Omega$, ce qui entraîne \perp .

On peut former des propositions qui sont dans des univers arbitraires, par exemple $(\Sigma X : \mathbf{U}_1)\mathbf{Eq}_{\mathbf{U}_1}(\mathbf{U}_0, X)$. Dans la théorie des ensembles un tel objet devrait être dans l'univers \mathbf{U}_2 . Mais ce type est en fait une proposition par le fait 2.1 et donc de type \mathbf{U}_0 par l'axiome de redimensionnement.

5.4. Opérateur de description

Si $(\Sigma x : A)B(x)$ est une proposition, on a

$$(\exists x : A)B(x) \rightarrow (\Sigma x : A)B(x)$$

et donc, dans ce cas, on peut «extraire» un témoin à partir d'une preuve d'existence.

Si A est un ensemble et $B(x)$ une propriété sur A , on définit

$$(\exists! x : A)B(x) = (\exists x : A)(B(x) \wedge (\Pi u : A)(B(u) \rightarrow \mathbf{Eq}_A(x, u))).$$

Si cette propriété est vérifiée, alors le type $(\Sigma x : A)B(x)$ est une proposition et on a donc

$$(\exists! x : A)B(x) \rightarrow (\Sigma x : A)B(x),$$

ce qui exprime l'opérateur de description dans la théorie des types dépendants.

Si A est un type quelconque et $R(x, y)$ une famille de propositions pour $x : A$, $y : A$ qui est une relation d'équivalence, on peut définir un *ensemble* A/R dont les éléments sont les *classes d'équivalence* de R , i.e. les propriétés $P(x)$, $x : A$ vérifiant

$$P(x) \wedge P(y) \rightarrow R(x, y) \quad P(x) \wedge R(x, y) \rightarrow P(y) \quad (\exists x : A)P(x).$$

En utilisant l'opérateur de description, on peut alors montrer que la surjection canonique $A \rightarrow A/R$ est universelle parmi les applications $f : A \rightarrow B$ dans un ensemble B telles que $R(x, y) \rightarrow \mathbf{Eq}_B(f(x), f(y))$.

Si l'opérateur de description est justifié, on peut par contre montrer la *négation* de l'axiome du choix global pour les ensembles

$$\neg((\Pi X : \mathbf{U})(\mathbf{set}(X) \wedge \mathbf{inh}(X) \rightarrow X))$$

simplement parce qu'un tel choix ne peut être invariant par automorphismes.

5.5. Catégories

La notion de *catégorie* sera une notion de *structure* au niveau des *groupoïdes*, analogue de la structure d'ordre au niveau des ensembles.

Une catégorie sera donc donnée par un *groupoïde* X avec une famille d'*ensembles* $\mathbf{Hom}_X(x_0, x_1)$, munie du morphisme identité $1_x : \mathbf{Hom}_X(x, x)$ et des opérations de composition et des lois d'associativité et d'élément neutre. (Ces opérations correspondent aux axiomes de réflexivité et de transitivité de la structure d'ordre.) On peut alors définir l'*ensemble* des isomorphismes $\mathbf{Iso}_X(x_0, x_1)$. Comme 1_x définit un élément de $\mathbf{Iso}_X(x, x)$, on obtient une application canonique

$$\mathbf{Eq}_X(x_0, x_1) \rightarrow \mathbf{Iso}_X(x_0, x_1).$$

La condition correspondante à l'antisymétrie est alors que cette application est une *équivalence* (i.e. ici une bijection) entre les ensembles $\mathbf{Eq}_X(x_0, x_1)$ et $\mathbf{Iso}_X(x_0, x_1)$.

On remarquera que la condition que $\mathbf{Hom}_X(x_0, x_1)$ soit un ensemble est formellement similaire à la condition pour une catégorie d'être localement petite. Mais cette condition concerne ici cruciallement la *complexité de l'égalité* du type $\mathbf{Hom}_X(x_0, x_1)$ et non sa «taille» ensembliste.

Grâce à l’axiome d’univalence, on peut munir le groupoïde des groupes par exemple, ou des ensembles ordonnés, d’une structure de catégorie.

Le type des catégories, pour un univers donné, forme un 2-groupoïde. Il est direct de définir dans ce cadre la notion de foncteur entre deux catégories, qui généralise la notion d’application monotone entre deux ensembles ordonnés, la notion d’adjonction et l’équivalence de deux catégories. On peut montrer que deux catégories équivalentes sont égales, et on peut donc transporter les propriétés d’une catégorie suivant une équivalence [AHR, UnFo].

L’exemple suivant montre une utilisation de l’opérateur de description au niveau des groupoïdes. On dira qu’un foncteur $F : A \rightarrow B$ entre deux catégories A et B est *pleinement fidèle* si l’application entre ensembles

$$\mathrm{Hom}_A(a_0, a_1) \rightarrow \mathrm{Hom}_B(F(a_0), F(a_1))$$

est une bijection. Le type

$$(\Pi b : B)(\exists x : A)\mathrm{Iso}_B(F(x), b)$$

exprime que F est *essentiellement surjectif*. On remarquera, en utilisant le corollaire 4.11, qu’être pleinement fidèle et être essentiellement surjectif sont des *propriétés* d’un foncteur.

LEMME 5.1 ([AHR]). — *Si F est pleinement fidèle et b est un objet de B , le type*

$$(\Sigma x : A)\mathrm{Iso}_B(F(x), b)$$

est une proposition.

Il en résulte que si F est pleinement fidèle, on a

$$(\exists x : A)\mathrm{Iso}_B(F(x), b) \rightarrow (\Sigma x : A)\mathrm{Iso}_B(F(x), b)$$

et ceci permet de montrer qu’un foncteur essentiellement surjectif et pleinement fidèle définit une adjonction sans utiliser l’axiome du choix [AHR].

RÉFÉRENCES

- [AHR] B. AHRENS, Ch. KAPULKIN et M. SHULMAN – *Univalent categories and Resk completion*, à paraître dans *Mathematical Structures in Computer Science* (arXiv :1303.0584), 2011.
- [AW] S. AWODEY et M.A. WARREN – *Homotopy theoretic models of identity types*, *Math. Proc. of the Cambridge Phil. Soc.* (2009), 45–55.
- [APW] S. AWODEY, A. PELAYO et M.A. WARREN – *Voevodsky’s Univalence Axiom in homotopy type theory*, *Notices Amer. Math. Soc.* 60 (2013), 1164–1167.
- [BCH] M. BEZEM, Th. COQUAND et S. HUBER – *A cubical set model of type theory*, à paraître dans *Types for Proofs and Programs*, post-proceeding of TYPES 2013.

- [Bou1] N. BOURBAKI – *Théorie des ensembles, Ch. 1 et 2*, Hermann, 1955 ; réimpression Springer, 2007.
- [Bou4] N. BOURBAKI – *Éléments de Mathématique. Théorie des ensembles, Ch. 4, Structures*, Hermann, 1957 ; réimpression Springer, 2007.
- [Un] N. BOURBAKI – *Univers*, SGA 4, M. Artin, A. Grothendieck, J.-L. Verdier eds, Lecture Notes in Math. 269 (1964), 185–217.
- [Br] N.G. de BRUIJN – *The mathematical language Automath, its usage, and some of its extensions* in Symposium on Automatic Demonstration, Lecture Notes in Math. 125 (1970), 73–100.
- [Ca] P. CARTIER – *Logique, catégories et faisceaux*, Sémin. Bourbaki 1977/78, Exp. n° 513, Astérisque 332 (2010), 123–146.
- [Ch] A. CHURCH – *A formulation of the simple theory of types*, The Journal of Symbolic Logic 5 (1940), 56–68.
- [Ge] G. GENTZEN – *Untersuchungen über das logische Schließen. I*, Math. Z. 39 (1934), 176–210.
- [Gi] J.-Y. GIRARD – *Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur*, Thèse de Doctorat d'Etat, Université Paris VII, 1972.
- [Go] G. GONTHIER – *Formal proof – The Four-Color theorem*, Notices Amer. Math. Soc. 55 (2008), 1370–1381.
- [FT] G. GONTHIER et al. – *A machine-checked proof of the Odd Order theorem*, Interactive Theorem Proving Conference 2013, LNCS Vol. 7998 (2013), 163–179.
- [Gr] A. GROTHENDIECK – *Esquisse d'un programme*, in «Geometric Galois actions», L. Schneps, P. Lochak, eds., London Math. Soc. Lecture Notes 242, Cambridge Univ. Press (1997), 5–48.
- [Ha] T.C. HALES – *Formal Proof*, Notices Amer. Math. Soc. 55 (2008), 1363–1369.
- [Hi] D. HILBERT – *Über das Unendliche*, Math. Annalen (Berlin), vol. 95 (1926), pp. 161–90.
- [He] M. HEDBERG – *A coherence theorem for Martin-Löf type theory*. J. Functional Programming 8 (1998), 413–436.
- [HS] M. HOFMANN et TH. STREICHER – *The groupoid interpretation of type theory*, in «Twenty-five years of constructive type theory» (Venice 1995), 43–111.
- [How] W. HOWARD – *The formulae-as-types notion of construction*, in «To H.B. Curry : Essays of Combinatory Logic, Lambda Calculus and Formalism», ed. Seldin et Hindley (1980), 479–490.
- [KLV] Ch. KAPULKIN, P. LEFANU LUMSDAINE, V. VOEVODSKY – *The simplicial model of univalent foundation*, arXiv :1203.2553, 2013.

- [KS] N. KRAUSS et Ch. SATTLER – *On the hierarchy of univalent universes*, arXiv :1311.4002, 2013
- [Lum] P. LEFANU LUMSDAINE – *Weak ω -categories from intensional type theory*, Logical Methods in Computer Science, Vol. 6 (2010), 1–19.
- [Mak] M. MAKKAI – *First-order logic with dependent sorts with application to category theory*, Preprint, 1995.
- [M-L] P. MARTIN-LÖF – *An intuitionistic theory of types : Predicative part*, in «Logic Colloquium '73», H.E. Rose and J.C. Shepherdson eds., (1975), 73–118.
- [Se] J-P. SERRE – *Homologie singulière des espaces fibrés. Applications*, Thèse, Paris, 1951.
- [Tu] D. TURNER – *Church's thesis and functional programming*, in «Church's thesis after 70 years», A. Olszewski, J. Wolenski, R. Janusz eds., (2006), 518 – 544.
- [UnFo] THE UNIVALENT FOUNDATIONS PROGRAM – *Homotopy type theory : Univalent foundations*, Institute for Advanced Study, 2013.
- [Voe] V. VOEVODSKY – *Experimental library of univalent formalization of mathematics*, à paraître dans Mathematical Structures in Computer Science, arXiv :1401.0053, 2013.

Thierry COQUAND

Chalmers Tekniska Högskola och
Göteborgs Universitet
412 96 Göteborg, Suède
E-mail : `coquand@chalmers.se`