La théorie des types, de Russell aux assistants à la démonstration

Thierry Coquand

COLLÈGE DE FRANCE
1530

Théorie des types

La *théorie des types* a été introduite par Bertrand Russell. Ses travaux comportent deux aspects complémentaires, qui vont correspondre aux deux parties de mon cours :

-le premier aspect est d'utiliser ce formalisme pour une représentation symbolique des mathématiques; ce cours présentera un raffinement de ce formalisme qui est utilisé maintenant dans plusieurs projets de formalisation de preuves mathématiques sur ordinateur.

-le deuxième aspect est d'utiliser ce formalisme pour analyser des questions de nature philosophique, comme le rapport de la logique et les mathématiques, ou le statut des axiomes en mathématiques; un des buts de ce cours sera de montrer comment ce raffinement peut suggérer de nouveaux éléments de réponses à ces questions.

Dans cette leçon inaugurale, je vais essayer de présenter ce sujet dans son contexte historique.

Contribution de Russell en logique

Peano, puis Russell avaient montré qu'il existe un langage mathématique symbolique dans lequel peuvent se traduire toutes les phrases que l'on peut prononcer en mathématiques ; Russell avait de plus montré comment tout raisonnement mathématique se réduisait à des règles de combinaison de ces signes, qu'il a énoncées. Herbrand, 1930

Cette représentation symbolique des raisonnements a joué un rôle crucial; cela a permis :

-programme de Hilbert (non-contradiction comme problème mathématique),

-théorème d'incomplétude de Gödel (reformulation de l'antinomie de Richard)

-théorie de la calculabilité

Logique d'Ordre Supérieur et le Lambda-Calcul

Alonzo Church 1941 *A Formulation of the Simple Theory of Types*

L'idée est d'introduire une stratification des objets mathématiques

Types de base : N (individus) et o (valeurs de vérité)

Types de fonctions $A \rightarrow B$

Russell introduit une telle stratification pour éviter les paradoxes : une fonction $f : A \rightarrow B$ ne peut s'appliquer qu'à un objet de type A.

On évite ainsi le paradoxe de Russell, qui considérait $\neg P(P)$.

Logique d'Ordre Supérieur et le Lambda-Calcul

Cette stratification des objets mathématiques est très naturelle, et s'applique directement en informatique, pour éviter des erreurs en manipulant ensemble des choses qui ne sont pas compatibles, comme des entiers et des chaînes de caractères par exemple.

Alonzo Church introduit aussi une notation pour les fonctions $\lambda_{x:N} x + 1 \ : \ N \to N$, et lois de calcul

$$(\lambda_{x:N} x + 1)(3) = (x + 1)(3/x) = 3 + 1 = 4$$

Bien que ces travaux ont joué un rôle très important en logique, et en informatique, leur influence en mathématique a été peu à peu éclipsée par celle de la théorie des ensembles.

Types et ensembles

It is a pity that a system such as Zermelo-Fraenkel set theory is usually presented in a purely formal way, because the conception behind it is quite straightforwardly based on type theory. (D. Scott)

Version simplifiée de Gödel de *Principia Mathematica*, avec seulement les types

$$T_0 = N \text{ et } T_{n+1} = T_n \to o$$

La théorie des ensembles peut être considérée comme une extension transfinie de cette hiérarchie le long des ordinaux. Dans cette théorie, on n'a pas de notion primitive de fonctions : une fonction est représentée par son graphe, qui est une relation fonctionnelle.

Logique et Mathématiques

Russell utilise ce formalisme pour analyser les rapports entre les mathématiques et la logique.

Il met en évidence trois axiomes, qui n'ont pas de justification logique :

-l'axiome de réductibilité (définitions imprédicatives)

-l'axiome multiplicatif (axiome du choix)

-l'axiome de l'infini

Ces axiomes sont introduits, non pas par leur caractère d'*évidence*, mais par le fait qu'ils ont de nombreuses conséquences (comme les lois en physique). *This axiom (of reducibility) has a purely pragmatic justification; it leads to the desired results, and to no others. But clearly it is not the sort of axiom with which we can rest content.* Russell, 1925.

Logique et Mathématiques

Cette analyse de Russell est remarquable; en particulier, l'axiome de réductibilité, va devenir un sujet principal de la future théorie des démonstrations. Cet axiome apparait dans la théorie des ensembles sous la forme du schéma de compréhension. Il est par exemple utilisé par Zermelo pour justifier le principe d'induction sur les entiers.

Poincaré (1909) est méfiant des définitions imprédicatives, dont le caractère circulaire est pour lui la source des paradoxes; il écrit à propos des axiomes de Zermelo :

Mais s'il a bien fermé sa bergerie, je ne suis pas sûr qu'il n'y ait pas enfermé le loup.

Poincaré insiste sur le caractère premier des principes d'induction, dans des débats très actuels, avec Russell et Zermelo, sur le rôle de la formalisation et des axiomes, publiés dans la *Revue de Métaphysique et de Morale*.

Logique et Mathématiques

En 1925, dans l'introduction de la nouvelle édition des *Principia Mathematica*, Russell, sous l'influence de discussions avec Wittgenstein, suggère de remplacer l'axiome de réductibilité par un *principe d'extensionalité*.

Les lois de l'égalité seront $a \equiv_A a$ et $P(a) \rightarrow P(b)$ si $a \equiv_A b$ et

-$(p \leftrightarrow q) \rightarrow p \equiv_o q$, formulée dans Appendix C

-$(\forall_{x:A} f(x) \equiv_B g(x)) \rightarrow f \equiv_{A \rightarrow B} g$

-un opérateur $\iota(P) : A$ pour $P : A \rightarrow o$ tel que $(\exists!_{x:A} P(x)) \rightarrow P(\iota(P))$

Ce dernier axiome est fondamental en mathématique : il permet de *nommer* un objet si celui-ci est défini de manière unique (appelé ⟪principe d'élément explicite⟫ par Henri Cartan 1943). Il est étudié par Russell dans un article célèbre *On Denoting* 1905.

Égalité en théorie des types

Russell suggère alors de remplacer l'axiome de réductibilité par ces principes.

Contrairement à cet axiome, ces principes d'extensionalité ont une motivation intuitive, que Russell essaie de préciser dans l'introduction de la nouvelle édition des *Principia Mathematica*.

L'idée est essentiellement de *définir* l'égalité de deux objets par induction sur le type de ces objets. *La notion de types joue un rôle crucial pour permettre une telle définition.*

C'est le germe de la notion de ⟪relations logiques⟫, qui sera mise au point par Robin Gandy (sous la direction de Turing), et qui joue maintenant un rôle important pour la sémantique des langages de programmation.

Formalisation des mathématiques

L'intuition de Russell est qu'une meilleure compréhension de la notion de l'égalité permettra de mieux représenter les raisonnements mathématiques, et d'éviter d'avoir à introduire des axiomes non justifiés.

Comme nous allons le voir, ces intuitions ont été confirmées par des travaux récents et de manière peut-être surprenante.

La première étape est un raffinement de la théorie des types, la notion de *types dépendants* pour formaliser les notions mathématiques sur ordinateur dans le but de *vérifier* les preuves.

C'est un développement naturel : une contribution de Russell était une formulation symbolique des preuves, qui permettait une vérification mécanique, et donc pouvait devoir se programmer.

AUTOMATH

Le mathématicien de Bruijn 1967 introduit le système AUTOMATH, pour écrire un programme de vérification des preuves mathématiques.

Il raffine le système de Church, qui présente deux limitations : on ne peut pas former des collections de *structures* et on n'a pas une notation explicite commode pour les *preuves*.

de Bruijn propose une notation pour faire référence à une hypothèse, et la notion de *contexte*

$$x : N,\ h : x > 0,\ y : N$$

Ceci suggère de traiter les *propositions* comme des *types*.

On peut aussi déclarer des types $A : \mathsf{Type},\ x : A$

Contexte

```
a b c : ℕ → ℝ
L : ℝ
a_le_b : a ≤ b
b_le_c : b ≤ c
a_to_L : ∀ ε > 0, ∃ N, ∀ n > N, |a n - L| < ε
c_to_L : seq_converges_to c L
ε : ℝ
ε_pos : ε > 0
ν : ℝ := ε ^ 2
this : ∃ N, ∀ n > N, |c n - L| < ε
⊢ ∃ N, ∀ n > N, |b n - L| < ε
```

```
Ω : Type u_1
Ω' : Type u_2
Ω'' : Type u_3
Ω''' : Type u_4
G : Type u_5
T : Type u_6
mΩ : MeasurableSpace Ω
μ : Measure Ω
mΩ' : MeasurableSpace Ω'
μ' : Measure Ω'
mΩ'' : MeasurableSpace Ω''
μ'' : Measure Ω''
mΩ''' : MeasurableSpace Ω'''
μ''' : Measure Ω'''
hG : MeasurableSpace G
inst✝⁴ : MeasurableSingletonClass G
inst✝³ : AddCommGroup G
inst✝² : MeasurableSub₂ G
inst✝¹ : Fintype G
inst✝ : MeasurableSpace T
```

AUTOMATH

Traitement uniforme des *objets*, des *structures* et des *preuves* mathématiques

$$\lambda_{x:A} t : A \to B \text{ si } t : B \ (x : A)$$

$$f(a) : B \text{ si } f : A \to B \text{ et } a : A$$

Algorithme uniforme pour vérifier la correction d'une assertion $a : A$

Correction de *typage* et vérification d'une *démonstration*

Types dépendants

Indépendamment, cette analogie entre les preuves et les objets était développée systématiquement par Per Martin-Löf, qui l'adapte à une formulation des principes d'induction

N sera un type primitif, avec ⟪constructeurs⟫ $0 : N$ et $x + 1 : N$ si $x : N$

On peut introduire $f : \prod_{x:N} C(x)$ si on a a construit

$$a : C(0) \text{ et } g : \prod_{x:N} C(x) \to C(x+1)$$

avec les lois de calcul $f(0) = a$ et $f(x+1) = g\ x\ f(x)$. Ce sont exactement les lois de calcul pour les fonctions primitives récursives, comme formulées par Hilbert.

Cela confirme les intuitions de Poincaré : le principe d'induction est *primitif* mais formulé de manière structurée avec une dualité introduction/élimination qui est *expliquée* par des lois de calcul.

Types dépendants

Dans cette approche, on représente l'activité mathématique comme un processus uniforme de construction d'objets, qui peuvent être des programmes, des structures mathématiques, ou des preuves. Tout cela est capturé par le jugement de typage

$$a : A$$

Le mathématicien David Mumford décrit ceci comme une unification remarquable de l'*ontologie* (objets mathématiques) et de l'*épistémologie* (preuves)

Types dépendants

Ma thèse (1985), faite à l'INRIA dans le projet FORMEL sous la direction de Gérard Huet, présentait le *calcul des constructions* et était un travail de synthèse des travaux de

-de Bruijn; en particulier le jugement $a : A$ doit être *décidable*, ce qui n'était pas le cas pour les systèmes de types à l'époque (Martin-Löf, NuPrl)

-Girard; avec un sytème imprédicatif qui contient la logique d'ordre supérieur de Church

-Martin-Löf; avec un calcul qui peut être vu simultanément comme un système de preuves et un langage de programmation fonctionnel

Gérard Huet pouvait alors écrire un programme prototype de vérification de correction de typage, et des preuves. Avoir un tel prototype était crucial pour évaluer ce calcul comme langage de preuves mathématiques et de programmation fonctionnelle.

Types dépendants

En particulier, je pouvais reproduire dans ce prototype les dérivations de Frege (1879) dans son livre sur l'*Idéographie*, qui définissait la clôture transitive d'un relation. L'étude systématique des représentations de notions inductives et des algorithmes (Krivine, Colson, Paulin) a conduit à l'ajout des types inductifs comme primitifs dans ce prototype (1989), confirmant ainsi les intuitions de Poincaré.

Grâce à des efforts collaboratifs impressionnants, ce prototype a évolué en plusieurs systèmes interactif de preuves, comme *Coq*, *Agda* et *Lean*, qui combinent dans un même formalisme preuves et programmes et qui sont maintenant utilisés à grande échelle.

Assistants de Démonstration

Utilisation pour écrire des programmes avec leur preuves de correction :

-projets CompCert (compilateurs C) et Iris (logique de séparation)

-*Computer Arithmetic and Formal Proofs: Verifying Floating-point Algorithms with the Coq System*

-*Software Foundations*, un série de livres pour enseigner la vérification formelle et la programmation certifiée.

Assistants de Démonstration

Exemples significatifs en *mathématique* de cette uniformité des preuves et programmes

-le théorème des quatre couleurs (2005); c'est une preuve qui doit considérer un nombre de cas impressionant (de l'ordre de 10^9), et qui utilise des algorithmes sophistiqués,

-le théorème que tout groupe d'ordre impair est résoluble (2012); Gonthier présente un exemple où l'on peut programmer, dans le système de preuves, un solveur SMT pour montrer un résultat sur des matrices à coefficients entiers,

-la preuve (2024) de $BB(5) = 47176870$, qui est le plus grand nombre d'étapes d'une machine de Turing qui termine avec cinq instructions sur deux symboles

Assistants de Démonstration

Ceci montre que l'on peut développer des mathématiques non triviales dans un système *sans axiomes*, uniquement bâti autour de constructions d'objets, qui peuvent être définis par induction.

Tous ces exemples toutefois concernent des objets mathématiques ⟪simples⟫, qui peuvent se décrire de manière combinatoire (graphes et groupes finis, machines de Turing).

Pour représenter des objets plus complexes, comme les nombres réels ou les catégories, le système *Lean* introduit des notions primitives supplémentaires de manière *axiomatique*, comme l'axiome du choix.

Assistants de Démonstration

Avec ces extensions, ce système, grâce aux efforts d'un groupe de mathématiciens enthousiastes, a permis de représenter formellement des exemples non triviaux qui manipulent des objets mathématiques non discrets.

-Vérification formelle d'un résultat de Scholze sur les ensembles condensés (2020),

-Preuve du retournement de la sphère (2022),

-Preuve de la conjecture de Marton, résolue par Gowers, Green, Manners, Tao (2024),

-Buzzard commence un projet de formalisation d'une version moderne de la preuve de Wiles-Taylor du théorème de Fermat.

Assistants de Démonstration

Une des motivations des systèmes interactifs de preuve est la complexité croissante des dérivations en mathématique. Cette complexité, combinée à la longueur des preuves, rend leur vérification par la communauté souvent insuffisante.

Le mathématicien Daniel Litt cite un exemple d'un lemme erroné qui a trois preuves publiées récentes différentes, toutes incorrectes, et aussi d'un *contre-exemple* publié, mais aussi incorrect !

Un avantage de ces systèmes de preuves est leur capacité à faciliter la collaboration : chaque participant peut s'appuyer de manière fiable sur des résultats déjà validés, ce qui renforce la rigueur et l'efficacité des travaux collaboratifs.

Un autre avantage, potentiel, est un complément de vérification pour des systèmes qui suggèrent des preuves, comme *AlphaProof*, qui a pu construire des solutions de problèmes d'Olympiade de Mathématiques 2024 dans *Lean*.

Assistants de Démonstration

Il est vraiment passionnant que ces avantages potentiels, qui ont motivé les projets AUTOMATH et le projet FORMEL à l'INRIA, se mettent ainsi en place.

Je voudrais maintenant expliquer comment l'utilisation de ces assistants de preuve ont permis la formulation d'une nouvelle forme du principe d'extensionalité, et de prolonger les réflexions de Russell (1925) sur ce principe.

Des lois nouvelles de l'égalité

Dans un traitement uniforme des preuves et objets, on peut définir $A \vee B$ comme $A + B$

Les lois d'introduction sont $i(a) : A + B$ si $a : A$ et $j(b) : A + B$ si $b : B$

La loi d'élimination

$$\frac{f : \prod_{a:A} C(i(a)) \qquad g : \prod_{b:B} C(j(b))}{h : \prod_{z:A\vee B} C(z)}$$

est inhabituelle, avec une mention explicite d'un objet preuve $z : A \vee B$.

Martin-Löf (1973) écrit systématiquement les lois logiques dans ce cadre, et, en particulier, les lois pour le type $a \equiv_A b$ qui représente la proposition de l'égalité de a et b.

Des lois nouvelles de l'égalité

La loi d'introduction sera $\mathsf{refl}_a : a \equiv_A a$ et la loi d'élimination

$$\frac{d : C(a, \mathsf{refl}_a)}{f : \prod_{x:A} \prod_{p:a \equiv_A x} C(x, p)}$$

avec $f\ a\ \mathsf{refl}_a = d$.

La loi habituelle est $\forall_{x:A}\ a \equiv_A x\ \rightarrow P(x)$ si $P(a)$.

Cette formulation ajoute donc une mention explicite de la preuve d'égalité $p : a \equiv_A x$.

Cette loi, *suggérée par des considérations purement formelles*, est absolument remarquable.

Elle a de nombreuses conséquences inattendues (comme le fait que tout type a naturellement une structure de groupoïde) mais qui ont longtemps semblées difficiles à comprendre intuitivement.

Types comme ensembles simpliciaux

Pour comprendre le caractère remarquable de cette loi, il faut introduire le modèle de la théorie des types, mis au point par Voevodsky (2005), où un type est interprété par un ensemble simplicial A qui vérifie la condition de Kan, une représentation abstraite de la notion d'espace.

Une famille de types $B(x)$ devient dans cette interprétation une fibration de Kan, au dessus de A. Le type $\sum_{x:A} B(x)$ correspond à l'espace total de cette fibration.

$a \equiv_A x$ sera interprété par l'espace des chemins de source a et de but x.

Une manière équivalente de formuler la nouvelle loi d'égalité est d'introduire le type des chemins C d'origine fixe a, défini comme $\sum_{x:A} a \equiv_A x$ et d'écrire que l'on a un élément de type $(a, \mathrm{refl}_a) \equiv_C (x, p)$ pour chaque $x : A$ et $p : a \equiv_A x$.

Types comme ensembles simpliciaux

La nouvelle loi d'égalité dit donc que le type $\sum_{x:A} a \equiv_A x$ est toujours contractile, ce qui est clair *géométriquement* : un chemin du point a à un point quelconque peut toujours se déformer continûment en un chemin constant en a (le principe du ⟪cordon d'aspirateur⟫).

Cela correspond exactement à la fibration introduite par Jean-Pierre Serre, dans sa thèse (1951). Pour tout espace A et point a, il introduit une nouvelle notion de fibration au dessus de A, dont l'espace total est contractile, et la fibre au dessus de x est l'espace des chemins qui joignent a et x (et donc la fibre au point a est l'espace des lacets en a).

Serre considère l'introduction de cette fibration *comme une de ses découvertes majeures*. Il remarque qu'il est ⟪étrange qu'une construction si simple ait autant de conséquences⟫.

Cette nouvelle loi de l'égalité capture de manière purement logique ce phénomène !

Le principe d'extensionalité

Que devient l'analyse de Russell du principe d'extensionalité dans ce cadre ?

$$(p \leftrightarrow q) \rightarrow p \equiv_o q$$

Les propositions deviennent des types. Ce que suggère cette loi, c'est que, si deux types A et B sont ⟪équivalents⟫ ils doivent être égaux (il y a un chemin de A en B; techniquement il y a une fibration au dessus de l'intervalle $\Delta[1]$ de fibre A en 0 et B en 1). Les ensembles simpliciaux ont une notion naturelle d'équivalence, l'*équivalence d'homotopie*. Ceci suggère donc à Voevodsky une conjecture précise, qui a été résolue par Bousfield (2006).

Si A et B sont homotopiquement équivalents, il existe une fibration $E \rightarrow \Delta[1]$ telle que $E(0) = A$ et $E(1) = B$.

C'est une propriété importante *nouvelle* des ensembles simpliciaux, et cela illustre bien la fécondité de ce nouveau point de vue, application de la *logique* à la théorie de l'*homotopie*

Le principe d'univalence

Voevodsky généralise alors ce résultat en une forme élégante du principe d'extensionalité : la flèche canonique $A \equiv_U B \to A \simeq B$ est elle-même une équivalence !

C'est une application de la théorie de l'homotopie à la logique.

Ce qui est remarquable, c'est que Voevodsky peut alors directement développer *formellement*, dans le système Coq, les conséquences de cette nouvelle forme du principe d'extensionalité.

Il montre ainsi que ce principe entraine le principe d'extensionalité pour les fonctions.

Les définitions sont surprenamment simples; par exemple la définition pour un type A d'être contractile est $\sum_{a:A} \prod_{x:A} a \equiv_A x$

Le principe de description définie

Dans cette approche, le principe de *description définie*, qui dit que l'on peut nommer un objet s'il est défini de manière unique, devient *prouvable* (alors que, dans le système de Church, à la suite de Peano et Russell, on devait introduire une nouvelle opération pour formuler ce principe).

L'existence unique de $x : A$ qui satisfait $B(x)$ peut être définie comme le type qui exprime que $\sum_{x:A} B(x)$ est *contractile*; comme un type contractile a, en particulier, un élément, le principe de description définie devient une tautologie.

Et ceci permet de capturer, comme par magie, des situations qui sont conceptuellement des cas d'unicité, comme unique à isomorphisme unique près, mais qui, en théorie des ensembles, ne peuvent pas être représentées comme des exemples de choix unique.

Par exemple, le résultat qu'un foncteur pleinement fidèle et essentiellement surjectif est une équivalence devient prouvable sans axiome du choix.

Le principe d'univalence

Lors de son développement formel des principes de base des mathématiques, Voevodsky s'attend à devoir utiliser l'axiome du choix, et l'axiome de réductibilité (qui sont valides dans son modèle).

Il est surpris que tout semble prouvable *sans* ces axiomes, juste comme conséquence du principe d'univalence.

Ceci est d'autant plus surprenant que ce formalisme permet d'exprimer directement des phénomènes qui, dans le cadre de la théorie des ensembles, portent sur des objets assez sophistiqués, comme les catégories d'ordre supérieur.

Ceci confirme l'intuition de Russell, qu'une meilleure compréhension de l'égalité va permettre d'éviter l'introduction de certains axiomes.

Justification ?

Le principe d'univalence a donc des conséquences remarquables, mais comment le justifier ?

La justification de Voevodsky du principe d'univalence provient du modèle où les types sont représentés par des ensembles simpliciaux de Kan, et utilise le fait que l'on peut munir cette catégorie d'une structure de catégorie de modèles (suivant Quillen). Toutefois, ce fait repose sur des notions prouvablement non effectives, comme la notion de fibration minimale de Kan, qui utilise une forme forte de l'axiome du choix.

Avec une telle justification, il est difficile de considérer que ce principe a un caractère intuitif. Ceci relance la question du statut de ce principe : doit-on l' accepter comme non évident, par le fait qu'il a de multiples conséquences remarquables et conceptuellement satisfaisantes ?

Justification ?

Un essai naturel est de partir de l'intuition qu'une telle justification doit être un raffinement de la notion d'ensemble introduite par Bishop (1967), qui est proche de l'idée de définir l'égalité par induction sur le type.

Dans cette approche, un ensemble est une collection d'objets munie d'une relation d'équivalence. Si on suit le traitement uniforme des propositions et types, cette relation d'équivalence doit être vue comme une famille de types. Ces types ont eux-mêmes une notion d'égalité, représentée par famille des types, et ainsi de suite. On peut alors comprendre la condition de Kan comme une généralisation dans ce cadre de la notion de relation d'équivalence.

En suivant cette intuition, nous avons pu construire (2013-2015) une justification *effective* de l'univalence. Ce travail s'inspire aussi de notions venant de la sémantique des langages de programmation, comme la notion d'ensembles nominaux, et la technique des relations logiques.

Justification ?

Un aspect inattendu de cette approche, mis à jour par Christian Sattler (2015), est le suivant. La justification de Voevodsky du principe d'univalence reposait sur l'existence d'une structure de modèle de Quillen sur les ensembles simpliciaux. Cette existence était montrée en utilisant l'axiome du choix.

En utilisant cette justification directe de l'univalence, on peut construire de manière réciproque une structure de modèle au sens de Quillen, et tous ces arguments sont effectifs.

De plus, ils ont pu être vérifiés formellement.

Quillen avait introduit cette notion comme une axiomatisation élégante de l'homotopie (en capturant les lemmes essentiels qui interviennent pour les fibrations de Serre). Une structure de modèle décrit ainsi une notion abstraite d'homotopie.

Justification ?

Une fois que l'on a cette structure de modèle, une question naturelle est si on peut construire de cette manière une telle structure, au sens de Quillen, à la structure de modèle sur les espaces topologiques, qui capture la notion de types d'homotopie.

Cette question a été résolue positivement par Sattler (2024).

Ceci donne un moyen de décrire de manière effective la notion de type d'homotopie.

Ceci montre aussi que beaucoup de notions abstraites, qui semblaient utiliser de manière essentielle des axiomes non justifiés, comme l'axiome du choix, peuvent en fait être expliquées effectivement.

Justification ?

En résumé, ces travaux des 20 dernières années, pour lesquels les assistants de preuves ont joué un rôle à différents niveaux, confirment les réflexions de Russell (1925) sur le principe d'extensionalité et l'importance de mieux comprendre l'égalité :

-le principe d'indiscernabilité, exprimé avec les notations introduites par AUTOMATH, révèle une analogie avec des idées qui ont révolutionné l'étude de l'homotopie,

-le principe d'extensionalité dans ce contexte devient le principe d'univalence, qui a de nombreuses conséquences, prouvées d'habitude avec l'axiome du choix et l'axiome de réductibilité,

-le principe d'univalence peut être lui justifié de manière effective, et ceci fournit une description effective de la notion de type d'homotopie.

- Cette approche est bien adaptée pour capturer des analogies entre des domaines différents des mathématiques et transporter des résultats et des objets d'un domaine sur un autre,

Conclusion

En conclusion, je voudrais insister sur le point suivant : pour analyser de manière satisfaisante des questions qui semblent purement philosophiques, comme le statut de l'égalité ou des axiomes, il semble nécessaire de faire appel à des concepts mathématiques/informatiques techniques, comme celui de structure de modèle de Quillen, ou de topos d'ordre supérieur, ou celui de paramétricité et relations logiques.

C'est un des aspects que je trouve particulièrement passionnant dans ce domaine de recherche.