CAHIERS DE
TOPOLOGIE ET GÉOMÉTRIE DIFFÉRENTIELLE
CATÉGORIQUES

ALAIN PROUTÉ

Expressions indéterminées, constructivisme et axiome du choix

Cahiers de topologie et géométrie différentielle catégoriques, tome 33, n° 3 (1992), p. 279-288

<http://www.numdam.org/item?id=CTGDC_1992__33_3_279_0>

CAHIERS DE TOPOLOGIE
ET GEOMETRIE DIFFERENTIELLE
CATEGORIQUES

VOLUME XXXIII
3ème trimestre 1992

Expressions indéterminées, Constructivisme et Axiome du Choix

Alain Prouté[1]

ABSTRACT : We introduce indeterminate expressions (in which substitution does not respect equality). This gives a convenient treatement of sub–types and quotient types in a construtive mathematical system. The Axiom of Choice is revisited at the light of this concept, and an example stresses the interest of the notion introduced.

Pour qu'un algorithme représente une fonction, il est nécessaire qu'il respecte l'égalité. En effet, un algorithme opère non pas sur des objets mathématiques, mais sur des représentations d'objets mathématiques. Il n'y a donc aucune garantie en général (pour un algorithme quelconque) que deux expressions distinctes représentant le même objet mathématique (i.e. : égales) soient transformées par l'algorithme en des expressions égales.

En fait, dans la plupart des systèmes formels, la syntaxe des algorithmes est telle que l'on peut en général montrer qu'il n'est possible d'écrire que des algorithmes respectant l'égalité. Mais bien sûr, la remarque ci–dessus laisse entrevoir la possibilité d'imaginer des systèmes formels dans lesquels on utiliserait des algorithmes ne respectant pas l'égalité. Le problème est de savoir si cela présente un intérêt. Le but de cet exposé est de démontrer que oui.

Pour pouvoir développer cette idée, nous devons nous placer dans un contexte formel. Nous adoptons un système formel de type "Martin–Löf", auquel nous allons apporter des modifications. Il n'est pas indispensable que le lecteur soit familier avec la théorie de Martin–Löf, car nous allons rappeler (sans entrer dans trop de détails) les principales idées de cette "théorie des types".

[1]Université Paris 7, CNRS URA 212, ap@mathp7.jussieu.fr

Notations : Par "connecteur logique", nous entendons l'un des symboles suivants : $\top$ (vrai), $\bot$ (faux), $\wedge$ (et), $\vee$ (ou), $\Rightarrow$ (implique), $\exists$ (il existe), $\forall$ (pour tout) et $\neg$ (non). Ces connecteurs ne sont pas indépendants les uns des autres, et en particulier, on doit considérer que pour tout énoncé A, l'énoncé $\neg A$ n'est qu'une notation pour $A \Rightarrow \bot$. De même, $\top$ et $\bot$ peuvent être traités comme des cas particuliers de $\wedge$ et $\vee$, puisqu'ils ne sont que les éléments neutres de ces opérations. Par ailleurs, nous utilisons des crochets, comme dans $E[x]$ pour signifier que E est une expression contenant éventuellement des occurences libres de la variable x. Le résultat de la substitution de a à x dans $E[x]$ sera noté $E[a]$.

L'interprétation de Brouwer-Heyting-Kolmogorov (interprétation BHK), est une liste d'*exigences* que l'on peut avoir quant à la signification des connecteurs logiques. En voici la liste.

- Une preuve de $A \wedge B$ est un couple formé d'une preuve de A et d'une preuve de B.
- Une preuve de $A \vee B$ est un couple formé d'un drapeau (valant 0 ou 1), et d'une preuve de A si le drapeau vaut 0, sinon d'une preuve de B si le drapeau vaut 1.
- Une preuve de $A \Rightarrow B$ est un algorithme construisant une preuve de B à partir d'une preuve quelconque de A.
- Une preuve de $\forall x \in E\ \varphi(x)$ est un algorithme construisant une preuve de $\varphi(a)$ à partir de toute expression a représentant un élément de E.
- Une preuve de $\exists x \in E\ \varphi(x)$ est un couple formé d'une expression a représentant un élément de E, et d'une preuve de $\varphi(a)$.

Bien sûr, pour que cette interprétation soit possible, une preuve ne doit pas être confondue avec le texte (expression) qui la représente.

Remarquons qu'une preuve d'une implication ou d'un énoncé universellement quantifié sont des algorithmes, et non pas des fonctions. En effet, il n'entre pas dans nos exigences que ces algorithmes respectent l'égalité des preuves.

Les exigence de l'interprétation BHK sont bien sûr des exigences "constructivistes", puisqu'elles impliquent en particulier que dès que l'on détient

une preuve d'existence (exécutée), on détient une représentation de l'objet dont l'existence est affirmée par cette preuve.

La cause de la non constructivité des mathématiques classiques est le raisonnement par l'absurde. Le principe du raisonnement par l'absurde peut s'exprimer par divers énoncés intuitionnistiquement équivalents. Nous n'en retiendrons qu'un, le principe du tiers exclus, que voici :[2]

$$\forall p \in \Omega \; p \vee (\neg p).$$

Ce principe dit donc que tout énoncé ne peut être que vrai ou faux (Aristote).

Par ailleurs, K. Gödel (1931)[2] a démontré qu'un système assez puissant pour permettre l'arithmétique contient nécessairement des énoncés indécidables. On constate alors facilement que le constructivisme et le tiers exclus sont incompatibles. En effet, soit p un énoncé indécidable. D'après le tiers exclus, on a une preuve de $p \vee (\neg p)$, et d'après la deuxième des exigences de l'interprétation BHK, cette preuve contient soit une preuve de p, soit une preuve de $\neg p$, ce qui est impossible.

On doit donc choisir entre classicisme (acceptation du tiers exclus) et constructivisme (renonciation au tiers exclus et à tout ce qui l'entraîne : raisonnement par l'absurde, double négation, formes trop fortes de l'axiome du choix).

Il y a une façon simple de satisfaire *de facto* aux exigences ci-dessus. Il suffit de décider que les énoncés sont des types. On doit intuitivement comprendre qu'un énoncé, lorsqu'il est vu comme un type, est simplement l'*ensemble des preuves* de cet énoncé. On peut alors traduire l'interprétation BHK en langage mathématique usuel, ce qui donne la *définition* suivante des connecteurs logiques :

$$\begin{array}{ll} A \wedge B = A \times B & A \vee B = A \amalg B \\ A \Rightarrow B = \overline{B^A} & \\ (\forall x \in E \; \varphi(x)) = \overline{\prod}_{x \in E} \varphi(x) & (\exists x \in E \; \varphi(x)) = \coprod_{x \in E} \varphi(x) \\ \top = \{*\} \text{ (ensemble singleton)} & \bot = \emptyset \end{array}$$

où $\overline{B^A}$ représente le type des algorithmes (ne respectant pas nécessairement l'égalité) de A vers B (et non pas celui des fonctions de A vers B, qui est noté

[2] Ω est l'ensemble intuitionniste des "valeurs de vérité", c'est-à-dire le quotient de l'ensemble de tous les énoncés par la relation d'équivalence engendrée par la déductibilité.

B^A), et $\overline{\prod}_{x\in E} A(x)$ le type des algorithmes (ne respectant pas nécessairement l'égalité) décrivant des familles d'éléments des types $A(x)$ quand x varie dans E (et non pas le produit au sens habituel).

Commentons un peu cette "traduction". Il faut se souvenir que l'on a confondu tout énoncé A avec l'ensemble de ses preuves. On peut donc lire la première définition ci–dessus comme suit : "l'ensemble des preuves de $A \wedge B$ est le produit cartésien de l'ensemble des preuves de A et de l'ensemble des preuves de B". Ce n'est qu'une façon d'énoncer la première règle de l'interprétation BHK.

Les autres définitions s'interprètent de même. Nous nous attarderons juste un peu sur celle du connecteur $\exists$. Il faut imaginer qu'un élément de l'union disjointe $\coprod_{x\in E} F(x)$ de la famille d'ensembles $(F(x))_{x\in E}$ est la même chose qu'un couple (i, x), où i est un élément de E (que l'on pourrait appeler "l'indice" de (i, x)), et x un élément dont le type dépend de la valeur de i. En fait, il y a la même différence entre ce concept et celui de produit cartésien, qu'entre la notion d'espace fibré et celle de produit cartésien (fibré trivial) d'espaces topologiques. Les informaticiens ont baptisé cette construction "produit dépendant", pour signifier un ensemble de couples dont le type du deuxième élément dépend de la valeur du premier élément. Le lecteur pourra essayer d'interpréter le produit $\prod_{x\in E} F(x)$ comme un ensemble de "fonctions dépendantes" (en fait l'analogue d'une section d'un espace fibré). C'est pourquoi nous adoptons la notation des fonctions $x \mapsto f(x)$ pour représenter un élément d'un tel produit. L'ensemble $\overline{\prod}_{x\in E} F(x)$ est interprété comme un ensemble d'algorithmes dépendants. Bien sûr, le produit dépendant $\coprod_{x\in E} F(x)$ a deux projections, la première sur E (projection sur la base d'un fibré), la deuxième "dépendante" (projection sur la fibre au-dessus du point qui est la projection sur la base).

Le système de Martin–Löf contient d'autres constructeurs de types que ceux cités plus haut. En gros, il y a un type des entiers $\mathbf{N}$, un schéma de construction de types récursifs (dont $\mathbf{N}$ n'est qu'un cas particulier), et des univers U_1, U_2,... permettant de parler de la classe des ensembles, etc ...

Par contre, il n'y a pas de constructeur de sous–types, ni de constructeur de types quotients. Nous allons introduire ci-après le concept de sous–type, et nous allons voir comment il nous mène à renoncer au respect de l'égalité par la substitution. Ceci ne veut pas dire que nous renonçons à l'interprétation extensionnelle de l'égalité entre fonctions. Nous allons revenir sur cette question un peu plus loin.

Une expression $E[x]$ contenant éventuellement des occurences libres de la variable x est dite *déterminée en* x, si pour deux expressions égales a et b, les expressions $E[a]$ et $E[b]$ sont égales. Si l'on ne sait pas prouver que $E[x]$ est déterminée en x, on dira que $E[x]$ est *indéterminée* en x.

Notez que le fait que l'expression $E[x]$ soit indéterminée en x ne signifie pas qu'il y ait des doutes sur l'existence de $E[x]$. Il n'y a de doute que sur sa "classe d'égalité". Du point de vue informatique, cela signifie que $x \mapsto E[x]$ est un algorithme qui termine effectivement et produit un élément du bon type, mais qui par contre ne respecte pas l'égalité.

On distingue deux sortes de systèmes formels, suivant que la sémantique de l'égalité y est *extensionnelle* ou *intensionnelle*. Le théorème d'incomplétude de Gödel affirme que dans tout système formel assez puissant pour que l'on puisse y faire de l'arithmétique, il existe des énoncés indécidables. C'est le cas bien sûr du système qui nous intéresse. Il en résulte en particulier que l'égalité extensionnelle (c'est–à–dire au sens habituel des mathématiques : $f = g \Leftrightarrow \forall x\ f(x) = g(x)$) entre fonctions de $\mathbf{N}$ vers $\mathbf{N}$ n'est pas testable par algorithme, ou encore que l'égalité vue comme une fonction de $\mathbf{N}^{\mathbf{N}} \times \mathbf{N}^{\mathbf{N}}$ vers $\{\top, \bot\}$ n'est pas programmable.

Il y a donc un décalage entre ce que l'on aimerait que l'égalité soit, et ce que nous sommes capables de calculer. La notion d'égalité que l'on obtient en s'en tenant à des algorithmes testant l'égalité des fonctions (et aussi d'autres types d'objets) se nomme *égalité intensionnelle*. Elle est plus faible que l'égalité extensionnelle, en ce sens que si deux objets sont intensionnellement égaux, alors il sont extensionnellement égaux, mais pas réciproquement. Bien sûr, une définition précise de l'égalité intensionnelle dépend fortement d'une définition précise du système formel.

Toutefois, il y a des types pour lesquels les deux notions d'égalité se confondent. Quand c'est le cas, nous dirons que le type est "normal". Le type $\mathbf{N}$ des entiers est normal.

Du point de vue informatique, un type normal est un type pour lequel il existe un algorithme de normalisation confluent et Noethérien, c'est–à–dire un algorithme transformant toute expression de ce type en une expression normale (forme normale) telle que deux expressions soient (extensionnellement) égales si et seulement si leurs formes normales sont syntaxiquement identiques.

Supposons maintenant que nous effectuons les calculs en "appel par valeur", ce qui signifie que l'argument a d'un terme applicatif (de la forme $f(a)$)

est calculé avant que ne lui soit appliqué la fonction ou l'algorithme f. Alors, si A est normal, f va nécessairement respecter l'égalité. En effet, le calcul de l'argument a pour effet de remplacer l'expression représentant cet argument par sa forme normale, et il en résulte que deux expressions égales de type A seront toujours remplacées par des expressions *syntaxiquement identiques*, avant même que l'algorithme représentant f ne leur soit appliqué. Comme bien sûr un algorithme donne des résultats identiques sur des expressions identiques, les résultats obtenus seront égaux parce qu'identiques. Un des axiomes cruciaux de notre système est donc le suivant (Axiome de détermination) :

Si A est un type normal, x une variable de type A, et $E[x]$ une expression quelconque, alors $E[x]$ est déterminée en x.

- Si A est un type, et $p[x]$ un énoncé contenant d'éventuelles occurrences de la variable x (de type A), alors

$$\{x \in A \mid p[x]\}$$

est un type (appelé un "sous-type" de A)

Bien sûr, il faut comprendre $\{x \in A \mid p[x]\}$ comme l'ensemble des éléments x de A pour lesquels $p[x]$ est vrai.

Comment fait-on pour construire un élément de $\{x \in A \mid p[x]\}$? On commence par construire un élément a de A, puis on prouve $p[a]$. On voit donc que si l'on désire conserver une trace de cette preuve dans la représentation de l'élément obtenu, il faut représenter un élément d'un sous-type de A comme un couple, formé d'un élément de A et d'une preuve que cet élément satisfait l'énoncé caractérisant ce sous-type.

Toutefois, l'inclusion de $\{x \in A \mid p[x]\}$ dans A doit être injective. Il en résulte que si on démontre $p[a]$ à l'aide de deux preuves différentes (disons P et Q), les deux couples

$$(a, P) \qquad \text{et} \qquad (a, Q)$$

représentent le même élément de $\{x \in A \mid p[x]\}$, même si la preuve P n'est pas égale à la preuve Q.

Comme un élément b de $\{x \in A \mid p[x]\}$ est un couple, interprétons les deux projections. $b.0$ est le premier élément du couple, autrement-dit, $b \mapsto b.0$

est l'inclusion canonique de $\{x \in A \mid p[x]\}$ dans A. $b.1$ est une preuve de $p[b.0]$. La discussion précédente montre clairement que l'expression $b.1$ est *indéterminée* en b.

L'axiome du choix est l'énoncé suivant, dans lequel A et B sont des types quelconques, et $R(x, y)$ une relation entre A et B.

$$\forall x \in A\ \exists y \in B\ R(x, y) \Rightarrow \exists f \in B^A\ \forall x \in A\ R(x, f(x)).$$

Martin–Löf *prouve* l'axiome du choix dans son système. Voici sa preuve :

$$p \mapsto (x \mapsto p(x).0, x \mapsto p(x).1).$$

p est une preuve de la prémisse, à savoir $\forall x \in A\ \exists y \in B\ R(x, y)$. Il en résulte donc que $p(x)$ (pour x donné dans A) est un couple formé d'un élément $p(x).0$ de B et d'une preuve $p(x).1$ de $R(x, p(x).0)$. Ceci nous donne la fonction f cherchée : $x \mapsto p(x).0$, et une preuve de $\forall x \in A\ R(x, f(x))$, à savoir $x \mapsto p(x).1$.

Cette preuve n'est par contre pas correcte dans notre système. En effet, la preuve p est un élément de $\overline{\prod}_{x \in A} \ldots$, et par conséquent, l'expression $p(x)$ est indéterminée en x. On ne peut donc pas considérer l'expression $x \mapsto p(x).0$ comme une fonction, mais seulement comme un algorithme. Il est dès lors clair, que dans notre système, les deux énoncés plus faibles suivants sont démontrés :

"Axiome du Choix Indéterminé" (Notez le remplacement de B^A par $\overline{B^A}$) :

$$\forall x \in A\ \exists y \in B\ R(x, y) \Rightarrow \exists f \in \overline{B^A}\ \forall x \in A\ R(x, f(x)).$$

"Axiome du Choix Normal" (On suppose que A est normal) :

$$\forall x \in A\ \exists y \in B\ R(x, y) \Rightarrow \exists f \in B^A\ \forall x \in A\ R(x, f(x)).$$

En particulier, l'axiome des choix dénombrables ($A = \mathbf{N}$) est démontrable dans notre système. On voit donc que l'axiome du choix, à condition de ne pas en utiliser des versions trop générales n'est pas un obstacle à la constructivité.

Nous allons voir un peu plus loin que la forme générale de l'Axiome du Choix donnée plus haut ne peut pas être démontrée dans notre système.

Radu Diaconescu (1975)[1] a démontré (d'une manière non formelle) que l'axiome du choix entraîne le tiers exclus. Nous allons donner sa démonstration dans notre langage formel (sans l'écrire complètement).

Remarquons tout de suite que le théorème de Diaconescu ne peut pas être démontré dans le système de Martin-Löf. En effet, s'il pouvait l'être, le tiers exclus pourrait être démontré dans un système constructif contenant l'arithmétique, ce qui est impossible.

Commençons par introduire deux fonctions α et β de Ω vers Ω, par les formules suivantes :

$$\alpha(x) = x \vee (\neg x \wedge p), \qquad \beta(x) = \neg x \vee (x \wedge p).$$

Il est alors immédiat de prouver :

$$\alpha(\top) = \top, \qquad \beta(\bot) = \top, \qquad p \Rightarrow (\alpha = \beta).$$

Posons $\Gamma = \{g \in \Omega^\Omega \mid g = \alpha \vee g = \beta\}$. Alors $\overline{\alpha} = (\alpha, *)$ et $\overline{\beta} = (\beta, *)$ sont des éléments de Γ, où $*$ désigne des preuves d'énoncés "évidents", comme d'ailleurs dans la suite de cette démonstration. Les règles définissant la sémantique des sous-types donnent une preuve de $\alpha = \beta \Rightarrow \overline{\alpha} = \overline{\beta}$, donc de $p \Rightarrow \overline{\alpha} = \overline{\beta}$.

On peut alors prouver $\forall X \in \Gamma\ \exists x \in \Omega\ (X.0)(x)$, comme ceci :

$$X \mapsto [q \mapsto (\top, *), q \mapsto (\bot, *)](X.1).$$

où $[g, h]$ désigne l'unique fonction ou algorithme de $E \amalg F$ vers G dont les composantes sont g et h (règle d'élimination du constructeur $\amalg$).

Soit maintenant Δ une preuve de notre prémisse (c'est-à-dire de l'axiome du choix), alors Δ appliqué à la preuve ci-dessus donne une preuve de

$$\exists f \in \Omega^\Gamma\ \forall X \in \Gamma\ (X.0)(f(X)),$$

On pose donc

$$f = (\Delta(X \mapsto [q \mapsto (\top, *), q \mapsto (\bot, *)](X.1)).0,$$

$$K = (\Delta(X \mapsto [q \mapsto (\top, *), q \mapsto (\bot, *)](X.1)).1$$

(K est une preuve de $\forall X \in \Gamma\ (X.0)(f(X))$), et on peut conclure que $\alpha(f(\overline{\alpha}))$ (par $K(\overline{\alpha})$) et $\beta(f(\overline{\beta}))$ (par $K(\overline{\beta})$).

Par définition de α on a :

$$(f(\overline{\alpha}) \vee (\neg f(\overline{\alpha}) \wedge p))$$

ce qui entraîne $f(\overline{\alpha}) \vee p$. De même, on déduit $\neg f(\overline{\beta}) \vee p$, et de ces deux énoncés on déduit $\Big(f(\overline{\alpha}) \wedge \neg f(\overline{\beta})\Big) \vee p$.

Il suffit donc pour terminer de montrer que $f(\overline{\alpha}) \wedge \neg f(\overline{\beta})$ entraîne $\neg p$ (i.e. : $p \Rightarrow \bot$). Mais ceci résulte immédiatement de ce que p entraîne $\overline{\alpha} = \overline{\beta}$, et de ce que f *est une fonction* et non pas seulement un algorithme. □

Il est important de remarquer dans la preuve ci–dessus que l'on utilise la forme forte de l'axiome du choix. En effet, le type Γ, bien que ne semblant ne contenir que deux éléments, est très loin d'être un type normal. En fait, sans hypothèse sur p, on ne peut même pas savoir si Γ a deux éléments ou un seul.

Par ailleurs, nous avons déjà remarqué que cette démonstration ne peut pas être faite dans le système de Martin–Löf (même si on y ajoute le type Ω). Ceci montre que notre système apporte quelque chose de plus.

Nous pensons avoir montré que l'introduction des expressions indéterminées est non seulement naturelle, mais rend les systèmes formels basés sur la théorie des types plus proches des mathématiques usuelles.

Références

[1] **R. Diaconescu** (1975) *Axiom of Choice and Complementation.* Proc. Amer. Math. Soc. 51, 176-178.

[2] **K. Gödel** (1931) *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter System I.* Monatsh. Math. Phys. 38, 173-198.

[3] **A. Heyting** (1955) *Les fondements des mathématiques. Intuitionnisme. Théorie de la démonstration.* Gauthier-Villars, Paris.

[4] **P. Martin-Löf** (1982) *Constructive Mathematics and Computer Programming.* dans : Cohen et al. : *Logic, Methodology and Philosophy of Science VI* North-Holland, Amsterdam, 153-175.

Université de Paris 7
Département de mathématiques
2 Place Jussieu
75251 Paris Cedex 05