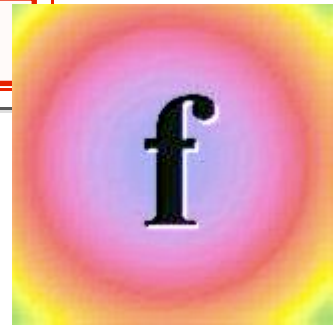


Et si on commençait par les fonctions !

To print higher resolution math symbols, click the Hi-Res button in the printing settings of the MS Word control panel.

Le 28 août 2009, par **Pierre Lescanne**

Professeur d'informatique à l'ENS de Lyon ([page web](#))



Dans cet article on montre qu'au lieu de fonder les mathématiques sur la notion d'ensemble comme le veut la tradition, on peut mettre la notion de fonction au centre de l'édifice.

Du bon usage des flèches

C ET article fait usage de trois types de flèches :

- la flèche vers la droite précédée de sa barre verticale \mapsto pour décrire les fonctions.
- la flèche pointée vers la droite \rightarrow pour la réduction des expressions
- la flèche pointée vers la gauche \leftarrow pour la substitution d'une variable par une expression.

Plutôt que des symboles ésotériques, nous avons choisi ces symboles qui sont d'usage courant en mathématiques. Que le lecteur nous pardonne la gymnastique que cela impose pour s'y retrouver.

Du passé faisons table rase

Dans la théorie des ensembles, une fonction f sur un ensemble A est définie par son graphe, c'est-à-dire par l'ensemble des couples $(a, f(a))$ quand a varie sur A . La définition du concept de fonction ne vient qu'après celui d'ensemble, une fonction étant définie comme un ensemble particulier. On dit aussi qu'elle est définie *en extension*, autrement dit qu'on parcourt (qu'on « s'étend » sur) l'ensemble des couples valeurs-résultats. En fait, on ne dit pas comment la valeur $f(a)$ est obtenue à partir de a , mais on dit simplement qu'elle est associée à a , d'une façon, certes rigoureuse, mais plus ou moins « magique ».

Dans ce qui suit nous allons voir comment on construit les mathématiques à partir de rien ou presque [1]. Traditionnellement depuis Cantor, le point de départ des mathématiques est celui d'*ensemble*. Nous allons, quant à nous, partir du concept de *fonction*, mais pas des fonctions du lycée sur les nombres réels, mais des fonctions qui manipulent ce qu'on connaît déjà.

Mais au fait qu'est-ce qu'on connaît quand on dit que le concept primitif est celui de fonction ?

Mais les « fonctions » parti ! Donc les fonctions que l'on va considérer sont des fonctions sur les fonctions.



Deux petites fonctions bien gentilles

Commençons par une fonction vraiment simple (peut-être trop simple), la *fonction identité* sur a . Vue de façon ensembliste, son graphe est la « diagonale » de l'ensemble produit $A \times A$, autrement dit l'ensemble des couples (a, a) . On aimerait dire que c'est la fonction qui à a associe a , en décrivant comment on obtient le résultat (c'est-à-dire a) à partir de la donnée (c'est-à-dire a), autrement dit on voudrait énoncer que la fonction *identité* c'est $a \mapsto a$. La fonction *identité* n'est pas assez compliquée, passons donc à la fonction *carré* sur \mathbb{N} [2]. Son graphe contient “ $(3,9), (0,0), (2,4), (4,16), (1,1), (12, 144), \dots$ ”, avec l'inconvénient de mettre des points qui laissent planer un doute sur ce qui n'est pas dit. On souhaiterait plutôt affirmer que *carré* est $n \mapsto n \times n$, en faisant l'hypothèse que l'opérateur de multiplication \times a déjà été défini. On dit alors que $a \mapsto a$ et $n \mapsto n \times n$ sont des définitions *en intention*, car elles portent en elles ce qu'on veut dire par les mots « *fonction identité* » et « *fonction carré* », c'est-à-dire un mécanisme qui dévoile comment on obtient (comment on calcule) le résultat à partir de la valeur qu'on a donnée. La fonction s'identifie à ce qu'on a l'« *intention* » de lui faire dire, d'où l'adjectif « *intentionnelle* ». Tout ça est un peu philosophique, mais ça va se clarifier par la suite.

Le mutisme des variables

Nous avons écrit $a \mapsto a$, mais nous aurions tout aussi bien pu écrire $x \mapsto x$ ou $y \mapsto y$, pour la même fonction. De même nous aurions pu écrire $k \mapsto k \times k$ ou $y \mapsto y \times y$ au lieu de $n \mapsto n \times n$. Cela signifie que le nom de la variable n'est pas vraiment important et qu'on peut donc la « renommer ». On dit aussi que la variable x est *liée* dans l'expression $x \mapsto x$ ou qu'elle est rendue *muette*. Le rôle de x est le même dans $x \mapsto x$ que dans $\forall x.(x \in \mathbb{Z}) \Rightarrow x^2 \geq 0$. Dans l'une et l'autre expression x peut-être changé en y sans changer sa signification. On sait en effet que $\forall x.(x \in \mathbb{Z}) \Rightarrow x^2 \geq 0$ a le même sens que $\forall y.(y \in \mathbb{Z}) \Rightarrow y^2 \geq 0$, c'est la même chose avec $x \mapsto x$ qui a le même sens que $y \mapsto y$.

Appliquer une fonction à une valeur

Une fonction est faite pour s'appliquer à des valeurs. Rappelons que les valeurs sont elles-mêmes des fonctions. Si on a une fonction f et une valeur a on va noter $f a$ le résultat de

l'application de f à a [3]. Ainsi l'application de l'identité à b s'écrit $(a \mapsto a) b$ et l'application de l'identité à carré s'écrit $(a \mapsto a) (n \mapsto n \times n)$. On a donc deux constructions de base, l'opérateur \mapsto (que nous appellerons *abstraction*) et l'application [4].



Encore quelques fonctions

Avec notre opérateur \mapsto , on peut définir des fonctions disons assez naïves ou assez « fondamentales » :

- la fonction qui fabrique des fonctions constantes, c'est une fonction à laquelle on donne une valeur, disons c , et qui retourne la fonction identique à c , à savoir $c \mapsto (x \mapsto c)$. Donnons lui son nom traditionnel et appelons la **K**. Notons que le premier \mapsto des deux prend c et produit une fonction.
- la fonction qui permute ses arguments, c'est la fonction qui prend une fonction f et retourne la fonction qui rend $f y x$ quand on lui donne x puis y , autrement dit la fonction : $f \mapsto (x \mapsto (y \mapsto (f y x)))$. Donnons lui son nom traditionnel et appelons la **C**. Quand on applique **C** f à x et y on obtient $f y x$.
- la fonction **deux**, c'est la fonction qui prend une fonction et rend la fonction obtenue en l'appliquant deux fois à son argument, autrement dit c'est $f \mapsto (x \mapsto f (f x))$. Si on applique **deux** à **carré** on obtient la fonction **bicarré**, qui rend la puissance quatre de sa valeur, autrement dit (**deux carré**) a le même effet [5] qu'une fonction que l'on noterait $n \mapsto n \times n \times n \times n$.

On vient de voir qu'on a considéré des fonctions qui retournent des fonctions comme le fait la fonction **K** ou des fonctions qui prennent comme valeur des fonctions comme le fait la fonction **C**. Rien d'étonnant à cela, puisqu'à partir de maintenant, *tout est fonction* ou, comment disent certains, *les fonctions sont les citoyens de première classe* [6]. Notons au passage, un fait remarquable, à savoir que la fonction *identité* est aussi bien l'identité sur A , sur B , sur \mathbb{N} , sur les fonctions de A vers B , etc. Comme elle est définie sur le « comment » plutôt que sur le « quoi » elle peut servir à plusieurs endroits, on dit qu'elle est « **polymorphe** » et c'est très bien ainsi.

Un nouveau calcul

Nous avons bien défini des fonctions, mais nous ne savons pas quoi en faire.

Calculer ou réduire des expressions

On applique une fonction à une valeur, ainsi si on applique la fonction identité à la valeur \mathbf{v} , cela donne $(a \mapsto a) \mathbf{v}$, oui *mézalor* ça n'est pas exactement ce qu'on veut ! On veut que $(a \mapsto a) \mathbf{v}$ « donne » \mathbf{v} . Pour cela, il nous faut un mécanisme de réduction des expressions qui se fonde lui même sur une notion de *substitution*. Nous n'insisterons pas sur ce mécanisme de substitution, nous dirons simplement qu'il doit être subtil, car il faut faire très attention aux variables liées, en évitant de lier une variable qui ne l'était pas ou de renommer une variable liée. Nous supposons que les lecteurs sont assez intelligents pour éviter de tomber dans ces pièges [7]

Notons $\mathbf{t}[x \leftarrow u]$ la substitution de la variable x par l'expression u dans l'expression \mathbf{t} . La réduction de $(x \mapsto \mathbf{t}) \mathbf{t}'$ produit $\mathbf{t}[x \leftarrow \mathbf{t}']$. Ainsi la réduction de $(a \mapsto a) \mathbf{v}$ est bien \mathbf{v} puisque clairement $a[a \leftarrow \mathbf{v}]$ est \mathbf{v} . La réduction se note par une flèche \rightarrow , ainsi on écrit :

$$(x \mapsto \mathbf{t}) \mathbf{t}' \rightarrow \mathbf{t}[x \leftarrow \mathbf{t}'].$$

Des expressions qui ne se réduisent jamais

La plus simple de ces expressions qui ne se réduisent jamais est l'expression $\Omega = (x \mapsto x x) (x \mapsto x x)$. La variable x apparaît liée à deux endroits différents, renommons donc le deuxième x ! Cela donne $(x \mapsto x x) (y \mapsto y y)$. Réduisons cette expression :

$$(x \mapsto x x) (y \mapsto y y) \rightarrow (x x)[x \leftarrow (y \mapsto y y)] \equiv (y \mapsto y y) (y \mapsto y y).$$

Surprise c'est la même à un renommage près ! Mais il faut dire que cette fonction x qui s'applique à elle-même a de quoi surprendre.

Une expression plus « utile » est $\mathbf{Y} = f \mapsto (x \mapsto f (x x)) (x \mapsto f (x x))$. Si on applique \mathbf{Y} à \mathbf{g} , on constate que $\mathbf{Y} \mathbf{g}$ se réduit vers $\mathbf{g} (\mathbf{Y} \mathbf{g})$. Autrement dit si l'on considère que deux expressions qui se réduisent l'une à l'autre sont « égales », on voit que $\mathbf{Y} \mathbf{g}$ est une expression qui est « égale » à $\mathbf{g} (\mathbf{Y} \mathbf{g})$, les mathématiciens disent que $\mathbf{Y} \mathbf{g}$ est un **point fixe** de \mathbf{g} .

Donner un sens aux expressions

Quand on a une expression un peu compliquée, on peut lui donner un sens en la réduisant jusqu'à ce qu'on ne puisse plus le faire. Mais deux questions surgissent :

- *Y a-t-il au plus une forme irréductible ?* La réponse est oui. En effet, quand on a le choix entre deux réductions possibles, si on choisit l'une plutôt que l'autre, cela n'a pas de conséquences, car on pourra toujours « se rattraper » plus tard et retomber vers le réduit de l'autre.
- *La réduction s'arrête-elle toujours ?* La réponse est non [8], comme on vient de le voir.

À partir de là, il y a deux solutions pour donner du sens.

- On ne considère que des expressions pour lesquelles la réduction termine, on peut le faire en restreignant les expressions acceptables, c'est-à-dire en limitant l'usage de l'application et de l'abstraction. On élimine en particulier, les fonctions qui s'appliquent à elle-même [9]
- On donne un sens à toutes les expressions, y compris aux expressions irréductibles. C'est ce qu'à fait **Dana Scott** dans une très belle construction d'un modèle qui évoque la construction de l'ensemble des nombres décrite dans l'article **Et si les nombres pouvaient être infinis à gauche de la virgule plutôt qu'à droite...**

Un calcul sans paires

Jusqu'à maintenant nous n'avons pas parlé de paires, parce nous n'avons pas eu besoin et nous n'en aurons pas besoin. En fait, au lieu d'écrire $f(x, y)$, on écrit $f x y$ et le processus qui consiste à appliquer la fonction g d'abord à x puis à appliquer le résultat $g x$ à y , autrement dit $g x y$, jouera le rôle de $g(x, y)$.

Application, abstraction et variables

Le formalisme que vous venons de définir est particulièrement épuré. Il a deux constructions : l'*abstraction* et l'*application*. Il y a en plus les *variables* pour démarrer la construction des expressions et la réduction pour calculer sur les expressions et leur donner un sens. C'est ce formalisme simple qui peut servir de base aux mathématiques.

Définir les entiers naturels

Von Neumann a le premier proposé de définir les entiers naturels dans la théorie des ensembles en procédant comme suit.

- \emptyset est 0, autrement dit, *vide* représente l'entier zéro.
- $\{\emptyset\}$ est 1, autrement dit l'ensemble qui ne contient que l'ensemble vide représente l'entier 1.
- $\{\{\emptyset\}, \emptyset\}$, autrement dit $\{1, \emptyset\}$ est 2, autrement dit 2 est l'ensemble qui contient les deux ensembles 0 et 1, c'est-à-dire l'ensemble vide et le singleton qui ne contient que vide.
- Plus généralement, pour représenter l'entier $n+1$ comme un ensemble on ajoute le singleton $\{n\}$ fait de l'ensemble n seul à l'ensemble qui représente n . Cet ensemble a exactement $n+1$ éléments.

Pour représenter les entiers dans les fonctions, on part d'une idée similaire, on essaye de trouver une fonction que l'on peut associer naturellement aux entiers. Nous avons déjà vu la représentation de l'entier 2, c'est la fonction **deux** qui associe à la fonction f la répétition deux fois de la fonction f . La fonction **n** est donc la fonction

$$f \mapsto (x \mapsto f(\dots n \text{ fois } \dots (f x)))$$

c'est-à-dire la répétition n fois de la fonction f . Le nombre 5 est donc représenté par la fonction **cinq** $= f \mapsto (x \mapsto f (f (f (f (f x))))))$. Le nombre 0 est représenté par la fonction $f \mapsto (x \mapsto x)$, autrement dit la fonction constante qui prend toujours la valeur *identité*. Le nombre 1 est donc représenté par la fonction $f \mapsto (x \mapsto f x)$.

A partir de là, on peut décrire toute l'arithmétique dans la théorie des fonctions et de là toutes les mathématiques. Ainsi l'addition de deux entiers naturels est définie par la fonction :

$$\mathbf{add} = (p \mapsto (q \mapsto (f \mapsto (x \mapsto (p f) ((q f) x))))).$$

Un peu d'histoire

L'origine de tout cela est un livre dû à **Russell** [10] et **Whitehead** dont une légende dit que personne ne l'a jamais lu tant il est rébarbatif, qui s'appelle les *Principia Mathematica* et qui avait pour but de fonder rigoureusement les mathématiques dans la tourmente de la crise des fondements. Cette recherche a été revitalisée par **Alonzo Church**, pour donner le formalisme décrit ici avec des notations différentes cependant.

A quoi tout cela sert ?

Ce formalisme a plusieurs applications contemporaines.

- Fonder les mathématiques.
- Offrir un cadre rigoureux et efficace pour raisonner, conduisant à la mécanisation du raisonnement sur ordinateur et aux assistants de preuve modernes comme Coq [11].
- Fournir un fondement mathématique aux langages de programmation et par là initier un nouveau style de programmation [12].

En savoir plus

- Ceux qui veulent appréhender les ressorts philosophiques de cette théorie liront le livre de Gilles Dowek, *Les métamorphoses du calcul : une étonnante histoire de mathématiques*, Le Pommier, Essais, (2007). *Grand prix de philosophie de l'Académie Française*.
- Ceux qui veulent comprendre la mathématique de tout cela liront le livre de Jean-Louis Krivine, *Lambda-Calcul, types et modèles*, Masson 1991.
- Ceux qui veulent surtout jouer avec les fonctions liront l'ouvrage ludique de Raymond Smullyan [13], *To Mock a Mockingbird*, ISBN 0192801422.

P.S. :

Ce qu'on vient d'expliquer s'appelle le λ -calcul, mais nous n'avons pas parlé de λ , une notation pour \mapsto .

Notes

[▲1] Bizarrement, cela suppose qu'on connaît des mathématiques. En effet, ce que nous allons présenter sert de base aux mathématiques, mais c'est aussi une part d'un domaine des mathématiques, que l'on appelle la *logique mathématique*.

[▲2] On admet qu'on connaît cet objet \mathbb{N} , on verra plus tard qu'il est fait de fonctions particulières.

[▲3] On note cela habituellement $f(a)$, mais il y a déjà assez de parenthèses comme ça !

[▲4] dont l'opérateur est une *espace* « ».

[▲5] Dans le formalisme que nous décrivons ici on ne peut pas démontrer que deux fonctions qui ont le même effet sont égales.

[▲6] contrairement à la théorie des ensembles où ce sont les ensembles qui sont les citoyens de première classe.

[▲7] De façon assez amusante les premières mises en œuvre de langages de programmation fondés sur ce formalisme sont, elles, tombées dans le piège. Le lecteur pourra s'entraîner avec la substitution :

$$((x \mapsto x) (y \mapsto x))[x \leftarrow y]$$

autrement dit on substitue y à x dans $(x \mapsto x) (y \mapsto x)$.

[▲8] Cela rappelle la matière traitée dans l'article **Est-ce que ça s'arrête ?**

[▲9] On entre alors dans le domaine de la *théorie des types*.

[▲10] Suprenant pour un mathématicien, Russell a reçu le prix Nobel de littérature en 1950 ! Mais à dire vrai il n'a consacré qu'une petite partie de sa vie aux mathématiques.

[▲11] La démonstration du **théorème des quatre couleurs** de Gonthier et **Werner** est une grande grande grande fonction. Voir également **cet article**.

[▲12] ☺ Ce style de programmation s'appelle sans originalité la *programmation fonctionnelle*.

[▲13] Raymond Smullyan est un élève de Church.

► Crédits images

Pour citer cet article : **Pierre Lescanne**, **Et si on commençait par les fonctions !**. *Images des Mathématiques*, CNRS, 2009. En ligne, URL : <http://images.math.cnrs.fr/Et-si-on-commencait-par-les.html>