Quatrième exposé : introduction à la logique catégorique

David A. Madore

23 novembre 1998

1 Motivations

La logique catégorique étudie à la fois l'aspect syntaxique et l'aspect sémantique de la logique. Dans l'étude de l'aspect sémantique, on cherche des catégories dont les objets et les flèches représentent les objets d'étude du système formel (typiquement, les ensembles). Dans l'étude de l'aspect syntaxique, les objets des catégories sont censées être des propositions et les flèches des démonstrations entre de ces propositions. En ce qui concerne l'étude de l'aspect sémantique, un type de catégories jouera un rôle essentiel, ce sont les *topoi*, originalement introduits par Grothendieck dans l'étude des topologies étale et plate des schémas. Les catégories utilisées dans la considération de la syntaxe ont moins été étudiées, et notamment celles qui servent pour le calcul des prédicats n'ont à ma connaissance pas été convenablement formalisées. On ne pourra donc donner ici qu'un aperçu assez sommaires de ces questions. Les topoi et les faisceaux sur les topologies de Grothendieck et de Lawvere-Tierney feront l'objet d'un exposé à part.

Il existe en logique catégorique un lien surprenant entre la syntaxe et la sémantique, c'est que les topoi sont des cas particuliers des catégories bicartésiennes closes utilisées dans l'étude de la syntaxe. Par ailleurs, lorsqu'on en écrit les objets et les flèches au moyen d'une notation appelée notation lambda, ces catégories bicartésiennes closes présentent une forte parenté avec le lambda-calcul typé. C'est cette ubiquité fondamentale qui constitue le début de ce qu'on appelle l'isomorphisme de Curry-Howard. Celle-ci n'est pas sans application pratique, puisque c'est elle qui sous-tend certains programmes informatiques de vérification formelle des démonstrations et de certification des programmes, comme le programme `Coq` développé par l'INRIA (dont la catégorie possède des propriétés supplémentaires admirables, comme celle d'admettre des constructions inductives positives arbitraires).

Sur le plan de la pure logique, la logique catégorique (des prédicats) est une logique très forte, qui contient strictement la logique du premier ordre, du second ordre, et même d'ordre n pour tout n : c'est une logique à types, dans laquelle on dispose du produit, coproduit et de l'exponentielle de deux types quelconques. C'est une logique naturellement intuitionniste (c'est-à-dire qu'elle ne vérifie pas le principe du *tertium non datur* $\neg\neg A \Rightarrow A$), et on peut même décrire de façon naturelle dans les mêmes lignes des logiques plus faibles comme la logique linéaire, ou encore des logiques plus complexes comme la logique modale.

Sur le plan épistémologique, certains y voient une façon de libérer les mathématiques de la tyrannie du "paradis Cantorien", la théorie des ensembles, à laquelle on reproche de trop se concentrer sur les codages particuliers des objets d'étude, plutôt que sur leurs propriétés. (*Aus diesem Paradies, daß Cantor uns geschaffen, sollen wir endlich vertrieben werden.*)

2 Catégories cartésiennes et cartésiennes closes

On veut considérer une catégorie dont les objets soient des propositions et les flèches $A \to B$ des démonstrations de B partant de A. On pourrait se contenter de considérer "brutalement" les telles démonstrations, et définir la composée de deux flèches $A \to B$ et $B \to C$ comme la démonstration de C à partir de A obtenue en mettant bout à bout les deux démonstrations. Un tel procédé n'a qu'un intérêt limité : par exemple, aucune flèche sauf les identités ne sera inversible, et aucun objet ne vérifie aucune propriété universelle intéressante.

Considérons le cas du ET logique : il est clair que proposer une démonstration de $A \wedge B$ à partir de S c'est "la même chose" que proposer une démonstration de A à partir de S ainsi qu'une démonstration de B à partir de S. C'est-à-dire que non seulement on demande une règle de logique permettant de former une démonstration $(u, v)\colon S \vdash A \wedge B$ à partir de $u\colon S \vdash A$ et $v\colon S \vdash B$, mais on demande aussi d'identifier la démonstration u avec la démonstration composée de (u, v) par la démonstration triviale $p_1\colon A \wedge B \vdash A$, et de même pour v. Et on demande encore que (u, v) soit la seule démonstration de $A \wedge B$ à partir de S ayant cette propriété. Ce sont là des identifications raisonables entre démonstrations. Les ayant fait, on remarque alors que $A \wedge B$ vérifie précisément la propriété universelle d'un produit de A et B : on a un isomorphisme naturel $\mathrm{Hom}(S, A \wedge B) \cong \mathrm{Hom}(S, A) \times \mathrm{Hom}(S, B)$.

Il en va de même du VRAI : une démonstration $S \vdash \top$ n'apporte rien d'informatif, et on souhaite donc toutes les identifier. Lorsqu'on le fait, le $\top$ devient un objet universel dans la catégorie étudiée. Celle-ci admet donc des produits finis quelconques.

Ceci nous conduit à définir une catégorie cartésienne comme une catégorie admettant un objet terminal et des produits binaires (i.e. des produits finis quelconques). En fait, ceci n'est pas très satisfaisant. Plutôt que de supposer l'existence de ces produits, on préférera se les donner. Soit :

Définition 2.1. *On appelle* catégorie cartésienne *une catégorie* $\mathbf{C}$ *munie d'un foncteur* $\top\colon \mathbf{1} \to \mathbf{C}$ *(identifié à un objet de* $\mathbf{C}$*) et d'un foncteur* $\wedge\colon \mathbf{C}^2 \to \mathbf{C}$ *qui définissent respectivement un objet universel (i.e. terminal)* $\top$ *de* $\mathbf{C}$ *et un produit* $A \wedge B$ *pour tous objets* A, B *de* $\mathbf{C}$ *(pour être encore plus pédant, ce sont donc des adjoints à droite respectivement du foncteur trivial* $\mathbf{C} \to \mathbf{1}$ *et du foncteur diagonal* $\mathbf{C} \to \mathbf{C}^2$*).*

La propriété universelle de l'implication est plus subtile. Il faut la voir dans le théorème de déduction : il revient au même de se donner une démonstration $A \wedge B \vdash C$ qu'une démonstration $A \vdash B \Rightarrow C$ (si on veut démontrer $B \Rightarrow C$, on suppose que B est vrai et on démontre C). On recheche donc un isomorphisme naturel

$$\mathrm{Hom}(A \wedge B, C) \cong \mathrm{Hom}(A, B \Rightarrow C)$$

Ou encore : l'objet $B \Rightarrow C$ est un objet qui représente le foncteur $A \mapsto \mathrm{Hom}(A \wedge B, C)$. (En particulier, une démonstration de $B \Rightarrow C$ à partir de $\top$, i.e. de rien du tout, correspond naturellement à une démonstration de C à partir de B : c'est pourquoi $B \Rightarrow C$ s'appelle souvent un objet "Hom interne".) Cette propriété est une adjonction de foncteurs : le foncteur $\Rightarrow$ est adjoint à droit du foncteur $\wedge$ (par rapport à la deuxième variable, c'est-à-dire que $B \Rightarrow$ est adjoint à droite de $\wedge B$).

On est donc conduit à la définition suivante :

Définition 2.2. *On appelle catégorie* cartésienne close *une catégorie cartésienne* $\mathbf{C}$ *munie d'un foncteur* $\Rightarrow : \mathbf{C}^2 \to \mathbf{C}$ *adjoint à droite par rapport à la deuxième variable du foncteur* $\wedge$.

(Le terme "close" vient de ce que $B \Rightarrow C$ peut être considéré comme un produit de copies de C indicées non pas par un certain ensemble mais par l'objet B.)

On rappelle certaines identités valables dans toute catégorie cartésienne close : $(A \wedge B) \wedge C \cong A \wedge (B \wedge C)$ (naturellement en A, B et C, et de plus cet objet représente un produit triple), $A \wedge B \cong B \wedge A$ (naturellement en A et B), $A \wedge \top \cong A$ (naturellement en A), $A \Rightarrow \top \cong \top$ (naturellement en A), $\top \Rightarrow A \cong A$ (naturellement en A), $A \Rightarrow B \wedge C \cong (A \Rightarrow B) \wedge (A \Rightarrow C)$ (naturellement en A, B, C), $A \wedge B \Rightarrow C \cong A \Rightarrow B \Rightarrow C$ (isomorphisme de curryfication/décurryfication, naturel en A, B, C).

On rappelle que d'un ensemble partiellement ordonné on peut faire une catégorie en décrétant qu'il y a une unique flèche $A \to B$ si et seulement si $A \leq B$. Dans ce cas, un objet terminal est précisément la même chose qu'un plus grand élément. Le produit binaire $A \wedge B$ est la même chose que l'inf de l'ensemble $\{A, B\}$ (i.e. le produit est représentable si et seulement si l'inf existe, et dans ce cas ils coïncident). Un ensemble partiellement ordonné est donc une catégorie cartésienne si et seulement si il a un plus grand élément et que toute partie à deux éléments (donc toute partie finie) admet une borne inférieure. Dans ces conditions, $B \Rightarrow C$ est alors (s'il existe) la borne supérieure des A tels que $\inf(A, B) \leq C$.

Parmi les catégories cartésiennes closes, on en signalera une particulièrement importante : c'est celle qui se déduit de l'ensemble partiellement ordonné $\{0, 1\}$: elle a deux objets, $1 = \top$ et $0 = \bot$ avec pour seule flèche hormis les identités une flèche $0 \to 1$. On la notera $\vec{\mathbf{2}}$. Les tables des opérations $\wedge$ et $\Rightarrow$ sont les tables habituelles.

3 Foncteurs logiques et constructions coüniverselles

La définition d'un morphisme entre catégories cartésiennes ou cartésiennes closes ne fait aucun doute, c'est un foncteur qui préserve les opérations définies sur la catégorie :

Définition 3.1. *Soient* $\mathbf{C}$ *et* $\mathbf{D}$ *des catégories cartésiennes (resp. cartésiennes closes). On appelle* foncteur logique (strict) *un foncteur* $F\colon \mathbf{C} \to \mathbf{D}$ *qui commute aux foncteurs* $\top$ *et* $\wedge$ *(resp.* $\top$*,* $\wedge$ *et* $\Rightarrow$*). Avec ces morphismes, les catégories cartésiennes (resp. cartésiennes closes) forment une catégorie notée* $\mathbf{Cart}$ *(resp.* $\mathbf{CartC}$*).*

La raison du mot "strict" est que, pour faire les choses vraiment correctement, il vaudrait mieux se contenter de demander que le foncteur commute aux connecteurs à isomorphisme près, c'est-à-dire se donner un foncteur $F\colon \mathbf{C} \to \mathbf{D}$ ainsi que des isomorphismes naturels $F(\top) \cong \top$, $F(A \wedge B) \cong F(A) \wedge F(B)$, et éventuellement $F(A \Rightarrow B) \cong F(A) \Rightarrow F(B)$, la composition étant alors donnée de façon relativement claire. Dans ce cas, on est obligé de considérer aussi les morphismes entre tels foncteurs (transformations naturelles), et on obtient finalement une 2-catégorie des catégories cartésiennes (resp. des catégories cartésiennes closes), notée $\mathfrak{Cart}$ (resp. $\mathfrak{CartC}$). La catégorie définie ci-dessus en est une strictification (note : je ne sais pas du tout ce que c'est mais je suis sûr que ça doit être ça). En l'occurrence, on n'a pas besoin de s'encombrer de tels détails car la strictification marche très bien dans le contexte présent.

L'idée intuitive est qu'un foncteur logique apporte éventuellement de nouvelles propositions à la catégorie cartésienne close, éventuellement de nouvelles démonstrations (axiomes), donc éventuellement des identifications nouvelles entre les propositions, mais que tout ce qui était vrai dans $\mathbf{C}$ reste vrai dans $\mathbf{D}$. On dit encore que la catégorie $\mathbf{D}$ (avec le foncteur F) est un *modèle* de la catégorie $\mathbf{C}$. Lorsque la catégorie $\mathbf{D}$ est $\vec{\mathbf{2}}$, on parlera d'un choix binaire de valeurs de vérité (on notera que lorsque $F(A) = 1$ et qu'on a une flèche $A \to B$ dans $\mathbf{C}$ alors on a aussi $F(B) = 1$).

Si un foncteur logique apporte de nouvelles propositions et de nouveaux axiomes, on peut s'attendre à ce que l'objet initial de la catégorie $\mathbf{Cart}$ ou $\mathbf{CartC}$ ne possède aucun axiome et ne possède que les propositions qui sont nécessaires. C'est effectivement le cas. D'ailleurs, cet objet initial est bien peu intéressant puisqu'il n'a qu'un objet, à savoir $\top$, et une seule flèche, l'identité sur $\top$. Nettement plus intéressantes sont les catégories obtenues en rajoutant librement des variables propositionnelles, que nous décrivons maintenant.

Définition 3.2. *Soit* $n \in \mathbb{N}$*. On note* $\mathbf{Cart}(P_1, \ldots, P_n)$ *(resp.* $\mathbf{CartC}(P_1, \ldots, P_n)$*) la catégorie dont les objets sont les catégories cartésiennes (resp. cartésiennes closes) munies de* n *objets* $P_1, \ldots, P_n$*, et dont les flèches sont les morphismes logiques entre telles catégories qui préservent* $P_1, \ldots, P_n$*.*

L'objet initial de $\mathbf{Cart}(P_1, \ldots, P_n)$ (resp. $\mathbf{CartC}(P_1, \ldots, P_n)$) est la catégorie cartésienne (resp. cartésienne close) librement engendrée par les variables propositionnelles $P_1, \ldots, P_n$, sans aucune relation entre elles, i.e. sans aucun axiome les liant.

Théorème 3.3. *La catégorie* $\mathbf{Cart}(P_1, \ldots, P_n)$ *(resp.* $\mathbf{CartC}(P_1, \ldots, P_n)$*) admet un objet initial noté* $\mathbf{cart}[P_1, \ldots, P_n]$ *(resp.* $\mathbf{cartC}[P_1, \ldots, P_n]$*). On l'appelle catégorie cartésienne (resp. cartésienne close) librement engendrée par les variables propositionnelles* $P_1, \ldots, P_n$*.*

(La démonstration est formelle ou bien découle de théorèmes archi-généraux, et que personne n'a jamais énoncé, sur la libre génération des structures. On invite le lecteur à s'en convaincre en répétant un nombre suffisant de fois les mots magiques “Sétrive Yahl”.)

Pour faire les choses vraiment correctement, il faudrait plutôt considérer un objet 2-initial de la catégorie $\mathfrak{Cart}(P_1, \dots, P_n)$ (resp. $\mathfrak{CartC}(P_1, \dots, P_n)$). Il faut cependant noter qu'en faisant ça on perd la notion précise d'égalité entre objets et on ne garde vraiment qu'une existence ou non d'isomorphismes.

On peut non seulement fabriquer des catégories cartésiennes (resp. cartésiennes closes) librement générées à partir de variables propositionnelles, mais aussi rajouter librement des variables propositionnelles, ou bien des axiomes à une catégorie cartésienne (resp. cartésienne close) existant.

Théorème 3.4. *Soit $\mathbf{C}$ une catégorie cartésienne (resp. cartésienne close), $n, k \in \mathbb{N}$, et $Q_1, \dots, Q_k$ des objets de $\mathbf{C}$. On note $\mathbf{Cart}(\mathbf{C}, P_1, \dots, P_n, x_1 : Q_1, \dots, x_k : Q_k)$ (resp. $\mathbf{CartC}(\dots)$) la catégorie dont les objets sont les foncteurs logiques $F\colon \mathbf{C} \to \mathbf{D}$, la catégorie $\mathbf{D}$ étant munie d'objets $P_1, \dots, P_n$ et de flèches $x_1, \dots, x_k$, avec $x_i\colon \top \to F(Q_i)$, les morphismes étant les foncteurs logiques $\mathbf{D} \to \mathbf{D}'$ qui préservent toutes ces données. Alors cette catégorie a un objet initial, noté $\mathbf{C}[P_1, \dots, P_n, x_1 : Q_1, \dots, x_k : Q_k]$.*

Bien sûr, on peut rajouter librement un axiome $x\colon A \to B$ puisque cela revient exactement à rajouter $x\colon \top \to A \Rightarrow B$, en vertu du théorème de déduction. Par ailleurs, il est trivial (?) que les objets de $\mathbf{C}[x_1 : Q_1, \dots, x_k : Q_k]$ sont les mêmes que les objets de $\mathbf{C}$.

On a le résultat important suivant :

Théorème 3.5 (complétude fonctionnelle). *Soit $\mathbf{C}$ une catégorie cartésienne (resp. cartésienne close), et A un type de $\mathbf{C}$. Soit f' une flèche $B \to C$ de la catégorie $\mathbf{C}[x : A]$. Alors il existe une unique flèche $f\colon A \wedge B \to C$ (ou, ce qui revient au même, dans le cas cartésien clos, $B \to A \Rightarrow C$) de $\mathbf{C}$ telle que $f' = f(x)$, avec l'abus de notation évident.*

(C'est intuitivement évident : postuler un nouvel axiome x qui affirme A permet de démontrer précisément les choses qu'on sait montrer à partir de A.)

Ainsi, la catégorie $\mathbf{C}[x : A]$ peut se construire très facilement comme catégorie coslice sur A, c'est-à-dire comme catégorie des flèches $A \to B$ dans $\mathbf{C}$. Cette observation peut d'ailleurs, retrospectivement, servir à démontrer le théorème (3.4), auquel cas le théorème de complétude fonctionnelle est trivial.

(En fait, les catégories **Cart** et **CartC** admettent tous les coproduits finis. C'est clair... si on le veut vraiment.)

4 Notations des flèches et lambda-calcul

Plaçons-nous dans une catégorie cartésienne close $\mathbf{C}$. Les objets de $\mathbf{C}$ seront appelés les *types*, et une flèche $f\colon A_1 \wedge \dots \wedge A_n \to B$ sera appelé un *terme de type B (à variables libres de types $A_1, \dots, A_n$)*. On remarque qu'on peut toujours rajouter des variables libres à un type, en composant par des flèches canoniques $A \to \top$. Par ailleurs, on notera par des lettres comme $x, y, z, \dots$ les termes identiques $A \to A$ de type A ayant une variable libre dans A.

Si g est un terme de type $B \Rightarrow C$ et f un terme de type B (ayant certaines variables libres non précisées), soit $g\colon Y \to B \Rightarrow C$ et $f\colon X \to B$, on peut construire un terme $g^*\colon Y \times B \to C$ et en composant en déduire un terme $Y \times X \to C$, donc de type C ayant les variables libres de f et g réunies, qu'on note gf et qu'on appelle *application* de g à f.

Attention : l'application n'est pas associative : lorsqu'on écrit hgf, c'est $(hg)f$ qu'on entend et non pas $h(gf)$. Au passage, signalons que quand on écrira $A \Rightarrow B \Rightarrow C$ c'est $A \Rightarrow (B \Rightarrow C)$ qu'on entend et non pas $(A \Rightarrow B) \Rightarrow C$.

En appliquant le théorème de déduction dans l'autre sens, on peut au contraire retirer des variables libres : si $f\colon X \wedge A \to B$ est un terme de type B ayant au moins une variable libre de type A (plus éventuellement d'autres non précisées), et qu'on a noté par x le terme identique de type A, on note $\lambda[x : A].f$ le terme $X \to A \Rightarrow B$ de type $A \Rightarrow B$ n'ayant plus la variable x de libre, obtenu par application du théorème de déduction. Cette opération lambda s'appelle l'*abstraction* (de la variable x).

Prenons un exemple très simple : si A est un type, alors x peut désigner le terme identique $A \to A$ (de type A ayant x comme variable libre de type A), et alors $\lambda[x : A].x$ désigne un terme de type $A \Rightarrow A$ sans variable libre (i.e. une flèche $\top \to A \Rightarrow A$), qu'on appelle abusivement l'identité sur A.

Un peu plus compliqué : si f est un terme de type $A \Rightarrow B$, alors $\lambda[x : A].fx$ est précisément égal à f. En effet, fx désigne le terme de type B à variable libre x de type A obtenu par application du théorème de déduction dans un sens à partir de f, et $\lambda[x : A].fx$ applique le théorème de déduction dans l'autre sens, donc on retombe sur f.

On vérifiera ainsi par exemple que les termes $k_{A,B} = \lambda[x : A].\lambda[y : B].x$ et $s_{A,B,C} = \lambda[x : A \Rightarrow B \Rightarrow C].\lambda[y : A \Rightarrow B].\lambda[z : A].xz(yz)$ sont des termes clos (i.e. sans variables libres) de types respectifs $A \Rightarrow B \Rightarrow A$ et $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$.

Par ailleurs, si u et v sont des termes de types respectifs A et B on note (u, v) le terme évident de type $A \wedge B$ (dont les variables libres sont celles de u et celles de v). On se permettra aussi, un peu abusivement, des notations telles que $\lambda[(u, v) : A \wedge B]. \cdots$, pour désigner la chose évidente : l'opération qui consiste à passer de $A \wedge B \Rightarrow X$ à $A \Rightarrow B \Rightarrow X$ s'appelle la curryfication. On note enfin $\star$ le terme unique de type $\top$ (et qui a toutes les variables libres qu'on voudra bien lui prêter).

Il faut préciser que dans l'écriture d'un terme λ, on omet parfois abusivement l'indication du type de la variable abstraite. Par exemple, on dira que $\lambda x.x$ est un terme de type $A \Rightarrow A$ (A étant un type non précisé), que $\lambda x.\lambda y.x$ (aussi noté $\lambda xy.x$) est un terme de type $A \Rightarrow B \Rightarrow A$, et que $\lambda xyz.xy(xz)$ est un terme de type $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$. Les informaticiens connaissent des algorithmes (algorithmiquement complexes) permettant d'effectuer la résolution des types : on pourra par exemple utiliser le langage `caml` à cet effet.

Il faut dire à ce stade-là que la catégorie cartésienne close maternelle pour toutes ces opérations est la catégorie $\mathbf{cartC}[P_1, \ldots, P_n]$ librement construite sur n types sans aucune information sur eux. Bien sûr, un terme de cette catégorie détermine un terme sur n'importe quelle catégorie cartésienne close munie de n types, d'après la propriété coüniverselle de $\mathbf{cartC}[P_1, \dots, P_n]$. En fait, tout terme de $\mathbf{cartC}[P_1, \dots, P_n]$ peut être écrit avec les opérations que nous avons définies. Pour cette raison, on les appelle les termes du lambda-calcul libre sur les n types $P_1, \dots, P_n$.

Considérons maintenant les termes suivants :

$$\begin{aligned}
i = \lambda x.x \quad & : A \Rightarrow A \\
k = \lambda xy.x \quad & : A \Rightarrow B \Rightarrow A \\
s = \lambda xyz.xz(yz) \quad & : (A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C \\
\star \quad & : \top \\
\pi_1 = \lambda(x,y).x \quad & : A \wedge B \Rightarrow A \\
\pi_2 = \lambda(x,y).y \quad & : A \wedge B \Rightarrow B \\
p = \lambda xy.(x,y) \quad & : A \Rightarrow B \Rightarrow A \wedge B
\end{aligned}$$

On les appelle *axiomes de Hilbert*. Ils possèdent la propriété remarquable que tout terme écrit avec les opérations que nous avons énumérées peut se réécrire avec la simple opération d'application à partir des axiomes de Hilbert et des variables libres ou axiomes de la catégorie.

Pour montrer ce résultat, commençons par considérer un terme $\lambda x.f$. On peut supposer que f est déjà écrit comme application d'axiomes de Hilbert. Soit $f = uv$ l'application la plus extérieure. Alors on a $\lambda x.f = s(\lambda x.u)(\lambda x.v)$ comme on le vérifie facilement. Reste donc le cas où f est soit une variable soit un axiome. Si $f = x$ alors $\lambda x.x = i$. Sinon, ou en fait dès que x n'apparaît pas dans f, on a $\lambda x.f = kf$. En ce qui concerne un terme du type $\lambda(x,y).f$, il suffit de l'écrire $\lambda z.f'$, où f' est obtenu en remplaçant x dans f par $\pi_1 z$ et y par $\pi_2 z$. Enfin, un terme (x,y) s'écrit pxy. Ceci termine la démonstration.

Ainsi, une démonstration peut se faire soit en utilisant les règles du calcul propositionnel et sans aucun axiome que ceux de la catégorie, soit en utilisant les axiomes de Hilbert et pour seule règle la règle d'application, c'est-à-dire le *modus ponens*.

Prenons un exemple. La commutativité du $\wedge$ est un terme de type $A \wedge B \Rightarrow B \wedge A$, qui s'écrit $c = \lambda(x,y).(y,x)$. Autrement dit, la démonstration avec les règles du calcul propositionnel s'écrit “par le théorème de déduction, on peut supposer $A \wedge B$ et on doit montrer $B \wedge A$; puisqu'on a $A \wedge B$, on a A d'une part (appelons x ce résultat), et B d'autre part (appelons y ce résultat) ; on a alors montré B et montré A, donc on a montré $B \wedge A$ (c'est (y,x)), ce qui conclut la démonstration”. Pour réécrire cette démonstration au moyen des axiomes de Hilbert, on l'écrit $c = \lambda z.p(\pi_2 z)(\pi_1 z)$, et il reste à éliminer l'abstraction : on a $c = s(\lambda z.p(\pi_2 z))(\lambda z.\pi_1 z)$, soit $c = s(s(\lambda z.p)(\lambda z.\pi_2 z))\pi_1$, soit encore $c = s(s(kp)\pi_2)\pi_1$. Voici une vérification de ce fait en utilisant le langage `caml` :

```
pleiades david ~ $ ocaml
        Objective Caml version 1.07

# let k = fun x -> fun y -> x;;
val k : 'a -> 'b -> 'a = <fun>
```

```
# let s = fun x -> fun y -> fun z -> x z (y z);;
val s : ('a -> 'b -> 'c) -> ('a -> 'b) -> 'a -> 'c = <fun>
# let pi1 = fun (x,y) -> x;;
val pi1 : 'a * 'b -> 'a = <fun>
# let pi2 = fun (x,y) -> y;;
val pi2 : 'a * 'b -> 'b = <fun>
# let p = fun x -> fun y -> x,y;;
val p : 'a -> 'b -> 'a * 'b = <fun>
# s(s(k p)pi2)pi1;;
- : '_a * '_b -> '_b * '_a = <fun>
```

Ici, si on veut écrire les étapes de la démonstration formelle, ce sont les suivantes :

1	s	$(A \wedge B \Rightarrow A \Rightarrow B \wedge A) \Rightarrow (A \wedge B \Rightarrow A) \Rightarrow A \wedge B \Rightarrow B \wedge A$	axiome
2	s	$(A \wedge B \Rightarrow B \Rightarrow A \Rightarrow B \wedge A) \Rightarrow (A \wedge B \Rightarrow B) \Rightarrow A \wedge B \Rightarrow A \Rightarrow B \wedge A$	axiome
3	k	$(B \Rightarrow A \Rightarrow B \wedge A) \Rightarrow A \wedge B \Rightarrow B \Rightarrow A \Rightarrow B \wedge A$	axiome
4	p	$B \Rightarrow A \Rightarrow B \wedge A$	axiome
5	kp	$A \wedge B \Rightarrow B \Rightarrow A \Rightarrow B \wedge A$	modus ponens 3/4
6	$s(kp)$	$(A \wedge B \Rightarrow B) \Rightarrow A \wedge B \Rightarrow A \Rightarrow B \wedge A$	modus ponens 2/5
7	π_2	$A \wedge B \Rightarrow B$	axiome
8	$s(kp)\pi_2$	$A \wedge B \Rightarrow A \Rightarrow B \wedge A$	modus ponens 6/7
9	$s(s(kp)\pi_2)$	$(A \wedge B \Rightarrow A) \Rightarrow A \wedge B \Rightarrow B \wedge A$	modus ponens 1/8
10	π_1	$A \wedge B \Rightarrow A$	axiome
11	$s(s(kp)\pi_2)\pi_1$	$A \wedge B \Rightarrow B \wedge A$	modus ponens 9/10

On notera au passage que les axiomes de Hilbert permettent, du moins en théorie, d'imaginer un langage de programmation entièrement dépourvu de variables.

Remarque culturelle : On sait construire une catégorie cartésienne close ayant exactement deux types, $\top$ et U, avec $\top \ncong U$, et $U \wedge U = U = U \Rightarrow U$. Cette catégorie permet de définir le lambda-calcul non typé. Le lambda-calcul non typé, à la différence du lambda-calcul librement typé que nous avons introduit, n'est pas décidable.

5 Catégories bicartésiennes closes

Jusqu'ici, on s'est intéressé exclusivement aux connecteurs logiques $\Rightarrow$, $\wedge$ et $\top$. Pour obtenir la logique intutionniste, il faut rajouter encore les connecteurs $\vee$ et $\perp$, dont les propriétés sont duales de $\wedge$ et $\top$, c'est-à-dire qu'ils vérifient des propriétés coüniverselles : $\mathrm{Hom}(A \vee B, C) \cong \mathrm{Hom}(A, C) \times \mathrm{Hom}(B, C)$ (naturellement en A, B, C) et $\mathrm{Hom}(\perp, A) \cong \{\bullet\}$ (naturellement en A). On pose donc :

Définition 5.1. *On appelle* catégorie bicartésienne close *une catégorie cartésienne close* $\mathbf{C}$ *munie d'un foncteur* $\perp\colon \mathbf{1} \to \mathbf{C}$ *(identifié à un objet de* $\mathbf{C}$*) et d'un foncteur* $\vee\colon \mathbf{C}^2 \to \mathbf{C}$ *qui définissent respectivement un objet initial* $\perp$ *de* $\mathbf{C}$ *et un coproduit* $A \vee B$ *pour tous objets* A, B *de* $\mathbf{C}$ *(pour être encore plus pédant, ce sont donc des adjoints à gauche respectivement du foncteur trivial* $\mathbf{C} \to \mathbf{1}$ *et du foncteur diagonal* $\mathbf{C} \to \mathbf{C}^2$*).*

Les résultats exposés sur les catégories cartésiennes closes restent valables, *mutatis mutandis* pour les catégories bicartésiennes closes. Il faut noter que l'ajout libre de nouveaux types dépend de si on considère la catégorie comme cartésienne, cartésienne close, ou bicartésienne close : les catégories $\mathbf{cart}[P_1, \ldots, P_n]$, $\mathbf{cartC}[P_1, \ldots, P_n]$ et $\mathbf{biCartC}[P_1, \ldots, P_n]$ ne sont pas équivalentes. En revanche, l'ajout libre de nouveaux axiomes n'en dépend pas : si $\mathbf{C}$ est bicartésienne close, alors la catégorie $\mathbf{C}[x_1 : Q_1, \ldots, x_k : Q_k]$ ne dépend pas du fait que la catégorie $\mathbf{C}$ soit considérée comme cartésienne, cartésienne close ou cocartésienne close (c'est de toute façon la catégorie coslice sur $Q_1 \wedge \cdots \wedge Q_k$).

On a des isomorphismes dans toute catégorie bicartésienne close, naturels en toutes les variables, entre $(A \Rightarrow C) \wedge (B \Rightarrow C)$ et $A \vee B \Rightarrow C$, entre $\perp \Rightarrow A$ et $\top$, entre $A \wedge (B \vee C)$ et $(A \wedge B) \vee (A \wedge C)$, et entre $A \wedge \perp$ et $\perp$. Par exemple pour le dernier : une flèche $A \wedge \perp \to S$ correspond à une flèche $\perp \to A \Rightarrow S$ par théorème de déduction, et il n'y en a qu'une par définition du $\perp$, donc $A \wedge \perp$ est aussi terminal, donc isomorphe à $\perp$. Dans cette démonstration, l'existence de $\Rightarrow$ (l'adjonction) est essentielle, et ce résultat n'est pas valable dans une catégorie bicartésienne quelconque.

On se gardera du reste des analogies trompeuses : on a bien des flèches naturelles et canoniques $A \vee (B \wedge C) \to (A \vee B) \wedge (A \vee C)$ et $A \vee \top \to \top$, et même des flèches dans l'autre sens (encore naturelles mais nettement moins canoniques :-), mais ces flèches *ne sont pas des isomorphismes en général*. Il suffit pour s'en convaincre de regarder la catégorie des ensembles (qui est bien bicartésienne close, le $\wedge$ étant le produit, le $\vee$ l'union disjointe, le $\top$ le singleton, le $\perp$ l'ensemble vide, et le $\Rightarrow$ l'ensemble des fonctions).

Les notations du lambda-calcul s'étendent encore aux catégories bicartésiennes closes, par exemple avec les notations suivantes. Si x est un terme de type A, on notera $x^{\leftarrow}$ (ou $x^{\leftarrow B}$ s'il est nécessaire de préciser) le terme de type $A \vee B$ qui s'en déduit, et de même pour $x^{\rightarrow}$. Pour définir un terme de type $A \vee B \Rightarrow C$, on note $\lambda[x : A|y : B].(f|g)$, où f et g sont tous deux des termes de type C, ayant x et y respectivement comme variable libre de type A ou B respectivement. Enfin, pour définir un terme de type $\perp \Rightarrow A$, on écrit tout simplement $\lambda[].()$.

Les axiomes de Hilbert se complètent en rajoutant

$$\begin{aligned} \omega_1 = \lambda x.x^{\leftarrow} \quad & : A \Rightarrow A \vee B \\ \omega_2 = \lambda y.y^{\rightarrow} \quad & : B \Rightarrow A \vee B \\ q = \lambda xy.\lambda[u : A|v : B].(xu|yv) \quad & : (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow A \vee B \Rightarrow C \\ F = \lambda[].() \quad & : \perp \Rightarrow A \end{aligned}$$

L'introduction du OU, et surtout du FAUX, dans la catégorie, a des conséquences assez fondamentales. À commencer par celle-ci :

Proposition 5.2. *Dans une catégorie bicartésienne close, pour tout objet A, s'il existe une flèche $A \to \perp$, alors c'est un isomorphisme (et donc A est un objet initial, et donc cette flèche est unique).*

Démonstration. Soit $A \xrightarrow{f} \perp$ cette flèche. La composée $\perp \to A \xrightarrow{f} \perp$ est l'identité car il n'y a qu'une flèche $\perp \to \perp$ (puisque $\perp$ est terminal). Il s'agit de montrer que la composée dans l'autre sens est aussi l'identité. Considérons le diagramme commutatif

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "(1_A, f)"'] & \perp \arrow[r] \arrow[d, equal] & A \\ A \wedge \perp \arrow[r, "\pi_2"] & \perp \arrow[r] & A \wedge \perp \arrow[u, "\pi_1"'] \end{tikzcd}$$

Le carré de droite est commutatif par définition de $(1_A, f)$, et le carré de gauche est commutatif car il n'y a qu'une flèche qui part de $\perp$. On a déjà montré que $A \wedge \perp$ est isomorphe à $\perp$, donc terminal, donc la composée des deux flèches de la ligne du bas est l'identité. Ainsi, la composée des deux flèches de la ligne du haut (celle qui nous intéresse) est la composée de la flèche de gauche et de la flèche de droite. Or par définition de $(1_A, f)$, c'est précisément 1_A. Ceci démontre le résultat. $\square$

De manière intuitive, et quelque peu informelle, ce résultat s'énonce en disant qu'il n'y a qu'une façon d'établir $\perp$, c'est de dire une connerie.

On note $\neg A$ l'objet $A \Rightarrow \perp$ (bien entendu, le $\neg$ est un foncteur). Une flèche $X \to \neg A$ correspond à une flèche $X \wedge A \to \perp$, et on a vu qu'il y en a au plus une : il y a au plus une façon de montrer une négation.

En particulier, si on appelle *catégorie booléienne* une catégorie bicartésienne close dans laquelle la flèche canonique $A \to \neg\neg A$ (définie par $\lambda[x : A].\lambda[y : A \Rightarrow \perp].yx$) est un isomorphisme, on voit que dans une catégorie booléienne tous les ensembles Hom ont au plus un élément, i.e. que la catégorie est en fait un ensemble préordonné. Celui-ci est alors équivalent à un ensemble partiellement ordonné (en quotientant par la relation d'équivalence associée au préordre), et un tel ensemble partiellement ordonné (i.e. un ensemble partiellement ordonné qui en tant que catégorie est une catégorie booléienne) est appelé une algèbre de Boole.

Dans l'étude de la logique catégorique, on est plus intéressé par les *algèbres de Heyting* : une algèbre de Heyting est un ensemble partiellement ordonné qui en tant que catégorie est une catégorie bicartésienne close. De manière équivalente, c'est un treillis distributif dans lequel pour tous éléments B et C l'ensemble des A tels que $A \wedge B \leq C$ admet une borne supérieure, notée $B \Rightarrow C$.

Remarque : Si dans une catégorie bicartésienne close $\mathbf{C}$ il existe un objet A tel que $A \cong \neg A$, alors on a une flèche $A \to A \Rightarrow \perp$, d'où $A \wedge A \to \perp$, et en composant avec la diagonale $A \to A \wedge A$ on en déduit une flèche $A \to \perp$. Or on a vu que ceci implique $A \cong \perp$, et donc $\neg A \cong \top$. Mais alors on a un isomorphisme $\perp \cong \top$, et il est facile de voir que tous les objets sont isomorphes, i.e. que la catégorie est (équivalente à) $\mathbf{1}$. Une telle catégorie est dite *inconsistante*. Dans le cas contraire, elle est dite *consistante*.

6 Passage au calcul des prédicats

Tout d'abord, un mot d'avertissement. Le calcul des prédicats en logique catégorique est un terrain bien moins balisé que le calcul propositionnel. Il n'y a pas d'étude bien approfondie (à ma connaissance) du lambda-calcul qui s'y rattache. Les auteurs se dégonflent et se place dans les topoi pour étudier le problème, ce qui est une solution peu élégante (les topoi sont les catégories adaptées à la sémantique, mais conviennent mal pour faire de la syntaxe). Ce qui suit est donc à prendre avec des précautions.

Comme on l'a déjà signalé dans l'exposé de rappels, il existe plusieurs sortes de calculs des prédicats. En logique du premier ordre, toutes les variables sont de la même espèce. En logique du second ordre, on a des variables d'individus et des variables de prédicats, sur lesquelles on peut aussi quantifier. Autrement dit, on a (au moins) deux *types* de variables, disons U pour les variables d'individus et Ω^{U^k} pour les variables de prédicats, cette notation servant à suggérer qu'elles se comportent comme des fonctions de U^k vers $\Omega = \Omega^{U^0}$ (le type des prédicats à 0 variables, ou encore des valeurs de vérité). Par ailleurs, on a introduit en logique du second ordre l'axiome de compréhension, qui permet de construire des termes de prédicats (de type Ω^{U^k}) à partir de n'importe quelle formule dans laquelle les seules variables libres sont des variables d'individus. On a choisi de noter les utilisations de l'axiome de compréhension avec la notation lambda, qui est à rapprocher de la notation lambda introduite plus haut dans cet exposé.

Le calcul des prédicats catégorique, ou calcul des types, pousse ces idées plus loin. On se permet d'étudier des termes de toute une multitude de types, ces types eux-mêmes formant une catégorie bicartésienne close (ou moins si on le souhaite, mais pourquoi se priver des bonnes choses). On possède donc pour les former tout l'arsenal de notations du lambda-calcul qu'on a déjà vues. Mais on utilisera des notations plus mathématiques pour la formation des types, notant 1 le type terminal, 0 le type initial, $\times$ le produit de deux types, $+$ leur somme, et avec un exposant le hom interne (exponentielle de deux types).

À côté de cela, on a sur certains types eux-mêmes des structures supplémentaires. Par exemple, dans l'étude de l'arithmétique, si on considère un type N pour les naturels, on le supposera muni d'un terme (clos) 0 de type N et d'un terme (clos) s de type N^N (la fonction successeur). Un type particulièrement important est le type Ω des valeurs de vérité, ou, plus précisément, des propositions (prédicats sans variable, les autres types de prédicats s'en déduisant par les mécanismes normaux de construction des types). C'est sur ce type, i.e. sur les termes de ce type, que s'appliquent les conjoncteurs $\wedge$, $\vee$, etc, que nous avons rencontrés. Plus précisément, $\Rightarrow$, $\wedge$ et $\vee$ sont des termes de type $\Omega^{\Omega\times\Omega}$, $\top$ et $\bot$ des termes de type Ω.

Mais si les termes de type Ω doivent correspondre aux propositions, alors les démonstrations sont des termes dont le type est lui-même un terme de type Ω. Autrement dit, on a non seulement une structure de catégorie bicartésienne close sur les types, mais aussi sur les propositions, i.e. sur le type Ω. On a deux catégories bicartésiennes closes en jeu : la catégorie "externe", structurale, et la catégorie "interne", le type Ω muni de sa structure. Au niveau externe, produits et autres opérations universelles seront notées par leurs notations mathématiques ($\times$, $+$, 1, etc.), tandis qu'au niveau interne, ils seront notés par les symboles logiques ($\wedge$, $\vee$, $\top$, etc.).

Comment fait-on pour mettre sur un objet d'une catégorie bicartésienne close une structure de catégorie bicartésienne close ? C'est une façon générale de faire en théorie des catégories : on suppose que pour tout type X l'ensemble $\Omega(X) = \mathrm{Hom}(X, \Omega)$ des termes de type Ω ayant variables libres dans X est muni d'une structure de catégorie bicartésienne close, et que ces structures sont compatibles les unes aux autres par changement de X. Rigoureusement :

Définition 6.1. *Soit* $\mathbf{C}$ *une catégorie bicartésienne close, et* Ω *un type (objet) de* $\mathbf{C}$*. Une structure de catégorie bicartésienne close interne sur l'objet* Ω *est la donnée d'une structure de catégorie bicartésienne close sur chacun des ensembles* $\Omega(X) = \mathrm{Hom}(X, \Omega)$ *(*X *un type de* $\mathbf{C}$*), ainsi que pour toute flèche* $\phi \colon X' \to X$ *d'un foncteur logique (strict)* $\Omega(X) \to \Omega(X')$ *qui sur les objets coïncide avec la flèche de composition (qui envoie* $f \colon X \to \Omega$ *sur* $f\phi \colon X' \to \Omega$*). Ceci se dit encore de la façon suivante : on a un foncteur de la catégorie* $\mathbf{C}$ *vers la catégorie* **BiCartC** *(des catégories bicartésiennes closes) dont le foncteur objet (i.e. la composée du foncteur en question avec le foncteur de* **BiCartC** *vers la catégorie des ensembles qui à une catégorie bicartésienne close associe l'ensemble de ses objets) est représentable par un objet* Ω *de* $\mathbf{C}$*.*

On remarquera qu'on n'exige pas la représentabilité du foncteur flèche, mais seulement du foncteur objet. Ceci est raisonable car on ne s'intéresse jamais à toutes les démonstrations (de n'importe quoi à partir de n'importe quoi). Cependant, il peut être judicieux de demander aussi la représentabilité du foncteur démonstrations de source donnée, i.e. exiger que pour tout X et tout $p \in \Omega(X)$ le foncteur qui à $X' \to X$ fait correspondre l'ensemble des flèches de source (l'image dans $\Omega(X')$ de) p, soit représentable par un objet de $\mathbf{C}$ (au-dessus de X bien sûr). Notamment lorsque p est en fait l'objet $\top$ (au-dessus de $X = 1$ disons), c'est ce qu'on a de plus près du “type des propositions qui sont vraies”.

D'autre part, le fait que Ω soit représentable (sur les objets) implique, d'après le lemme de Yoneda, que les morphismes $\wedge \colon \Omega \times \Omega \to \Omega$ et compagnie sont aussi représentables. Il est donc justifié de les considérer comme des termes de type $\Omega^{\Omega \times \Omega}$ etc.

Il faut noter qu'une démonstration, i.e. une flèche dans la catégorie interne Ω, a deux types de prémisses (dépendances, suppositions, hypothèses, variables libres, comme on voudra) : l'objet X de $\mathbf{C}$ au-dessus duquel elle est définie, et sa source dans $\Omega(X)$. Il se peut du reste fort bien que la source en question soit définie plus globalement que la démonstration elle-même (par exemple, on verra que $\exists(x : X).\top$, qui est une proposition sans variable libre, n'a pas de démonstration sans variable libre (si X est un type sur lequel on n'a aucune donnée) mais qu'il a une démonstration (à savoir $(x, \star)$) ayant une variable libre x de type X).

7 Quantificateurs

Étant donnés un type A et un prédicat p, terme de type Ω ayant variables libres de type $A \times X$ (le X servant à représenter d'autres variables libres éventuelles du prédicat), on cherche à donner un sens à $\forall(t : A).p$ et $\exists(t : A).p$, qui doivent être des termes de type $X \to \Omega$ (autrement dit, la quantifiation doit retirer la variable libre t de type A et ne pas toucher aux autres).

Commençons par la quantification universelle. Si on a une démonstration δ d'un terme de type $\forall(t : A).p$, et qu'on a un terme u de type A, il est sensé d'appliquer la démonstration "au cas particulier $t = u$", pour obtenir une démonstration de p_u (p dans lequel on a substitué u à t, ce qui s'écrit encore $(\lambda[t : A].p)u$ si on préfère). On notera tout simplement δu la démonstration ainsi construite. Réciproquement, si on cherche à construire une démonstration de $\forall(t : A).p$, on procède en supposant donné un t et en démontrant alors p. Autrement dit, donnée une démonstration δ de p (avec la variable libre t), on construit une démonstration notée $\lambda[t : A].\delta$ de $\forall(t : A).p$. Il faut bien noter l'analogie entre l'abstraction lambda de type "fantaisie" (supposons que j'aie une démonstration t d'une assertion A) et l'abstraction de type "instantiation" (supposons que t soit une variable de type A). La différence étant une différence de niveau : dans le second cas A est un type, i.e. un objet de la catégorie externe $\mathbf{C}$ tandis que dans le premier cas A est une proposition, i.e. un terme de type la catégorie interne Ω. De plus, dans le cas où A est un type, non seulement δ pourra dépendre de t mais son type lui-même (qui est un terme p de type Ω) en dépendra en général.

On peut maintenant formuler précisément la propriété universelle de $\forall(t : A).p$. Soit $p\colon X \times A \to \Omega$ un terme de type Ω ayant $X \times A$ pour type des variables libres, la variable libre de type A étant notée t. Alors $\forall(t : A).p$ est un terme de type $X \to \Omega$ (de type Ω ayant X pour type des variables libres) muni d'un morphisme $\varpi\colon (\forall(t : A).p) \to p$ dans la catégorie $\Omega(X \times A)$ (ici, $\forall(t : A).p$, objet de la catégorie $\Omega(X)$, est étendu à $\Omega(X \times A)$ de la façon connue), qui vérifie la propriété universelle suivante. Si $q\colon X \to \Omega$ est un terme de type Ω ayant X *mais pas* A pour type des variables libres, et qu'on se donne une flèche $\delta\colon q \to p$ dans la catégorie $\Omega(X \times A)$ (même commentaire que ci-dessus), alors il existe une unique flèche $\lambda[t : A].\delta\colon q \to \forall(t : A).p$ dans la catégorie $\Omega(X)$ dont la composée avec ϖ donne δ, i.e. telle que $\varpi \circ (\lambda[t : A].\delta) = \delta$.

Cette propriété universelle, ici détaillée lourdement, peut s'énoncer d'une multitude de façons différentes. Par exemple, on peut dire que $\varpi\colon \forall(t : A).p \to p$ est la flèche terminale de type $q \to p$ avec q un terme de type Ω *qui ne dépend pas de* A. C'est-à-dire que démontrer p à partir d'hypothèses q ne faisant pas invervenir t, c'est "la même chose" que démontrer p à partir de ces hypothèses q. Ou encore $\mathrm{Hom}_{\Omega(X\times A)}(q, p) \cong \mathrm{Hom}_{\Omega(X)}(q, \forall(t : A).p)$ pour tout objet q de $\Omega(X)$. On peut encore dire que le foncteur qui à p (objet de $\Omega(X \times A)$) associe $\forall(t : A).p$ (objet de $\Omega(X)$) est adjoint à droite du foncteur qui à q (objet de $\Omega(X)$ associe q transporté à $\Omega(X \times A)$. On peut encore transformer cette assertion en quelque chose de totalement incompréhensible avec des fibrations de catégories ou même (si on veut jouer avec les 2-catégories) des champs dans tous les sens. Il n'en demeure pas moins qu'elle est complètement évidente, et que $\forall(t : A).p$ est un produit catégorique de p indicé par A.

La propriété du $\exists$ et duale, elle est coüniverselle au lieu d'être universelle. C'est-à-dire que tandis que $\forall(t : A).p$ est la plus faible assertion indépendante de t et qui démontre

p, l'assertion $\exists(t : A).p$ est la plus forte assertion indépendante de t que p démontre, et on a $\mathrm{Hom}_{\Omega(X \times A)}(p, q) \cong \mathrm{Hom}_{\Omega(X)}(\exists(t : A).p, q)$. En ce qui concerne les notations, si x est un terme de type A et δ une démonstration de p_x (p dans lequel on a substitué x pour t, i.e. $(\lambda[t : A].p)x$), alors on note (x, δ) la démonstration de $\exists(t : A).p$ qui en découle. Et d'autre part, si δ est une démonstration de q, proposition indépendante de t, à partir de p dépendant de t de type A, on notera $(t?\delta)$ la démonstration de q à partir de $\exists(t : A).p$ obtenue par la propriété universelle (il faut lire : on a supposé qu'il existait un terme de type A vérifiant p, appelons-le t).

Bien entendu, si $\mathbf{C}$ est une catégorie bicartésienne close et Ω un type de $\mathbf{C}$ muni d'une structure de catégorie bicartésienne close, il n'est pas dit d'avance que toutes les quantifications existent (i.e. soient représentables) dans Ω. Lorsque c'est le cas, on dira que Ω est une catégorie bicartésienne close complète interne. Plus précisément, il faut non pas postuler l'existence de telles quantifications mais bien les choisir. Autrement dit :

Définition 7.1. *Une* CBCCCBCCCI *(catégorie bicartésienne close à catégorie bicartésienne close complète interne) est une catégorie bicartésienne close* $\mathbf{C}$ *munie d'un objet* Ω *et d'une structure de catégorie bicartésienne close sur* Ω *ainsi que de foncteurs internes* $\forall$ *et* $\exists$ *qui vérifient les propriétés universelles précisées ci-dessus. Un foncteur logique* $F\colon \mathbf{C} \to \mathbf{C}'$ *entre CBCCCBCCCI est un foncteur qui préserve toutes les données sur les catégories* $\mathbf{C}$ *et* Ω*, i.e. c'est un foncteur logique sur* $\mathbf{C}$ *qui vérifie* $F(\Omega) = \Omega'$*, et dont les applications induites* $\Omega(X) \to \Omega'(F(X))$ *sont munies d'une structure de foncteur logique de façon naturelle en* X*. On note* **CBCCCBCCCI** *la catégorie des CBCCCBCCCI avec pour flèches les foncteurs logiques.*

Un exemple de CBCCCBCCCI est la catégorie des ensembles avec pour Ω la catégorie interne $\vec{\mathbf{2}} = \{0, 1\}$.

De même que **Cart**, **CartC** et **BiCartC**, la catégorie **CBCCCBCCCI** admet toutes sortes de constructions coüniverselles. Elle admet un objet initial, noté **cbcccbccci**, qui a déjà une structure fort compliquée, et on a de plus la possibilité d'ajouter librement des types, des termes d'un type ou des démonstrations d'une proposition. Ces derniers ajouts se font avec des constructions coslice externe et coslice interne respectivement. (Ce qui est remarquable, c'est que le fait d'ajouter librement un terme p au type Ω ne perturbe pas sa nature de catégorie bicartésienne close complète interne.)

8 Récapitulation des notations et exemples

Ayant introduit tout un tas de notations non standard et pénibles, nous insistons sur la lourdeur en les récapitulant et en en donnant des exemples.

- (Élimination du $\Rightarrow$, ou *application*, ou *modus ponens*.) Si v est une démonstration de $P \Rightarrow Q$ et u une démonstration de P, alors vu est une démonstration de Q.

- (Introduction du $\Rightarrow$, ou *abstraction*, ou *fantaisie*, ou “théorème de déduction”.) Si δ est une démonstration de Q qui fait intervenir une hypothèse ξ démonstration de P, alors $\lambda[\xi : P].\delta$ est une démonstration de $P \Rightarrow Q$.
- (Élimination du $\wedge$.) Lorsqu'on démontre R à partir d'une hypothèse de type $P \wedge Q$, qui a été abstraite, on peut écrire cette hypothèse (u, v) et considérer u comme une démonstration de P et v comme une démonstration de Q. La démonstration finale s'écrit alors $\lambda[(u, v) : P \wedge Q]. \cdots$.[1]
- (Introduction du $\wedge$.) Si u est une démonstration de P et v une démonstration de Q, alors (u, v) est une démonstration de $P \wedge Q$.
- (Élimination du $\top$: il n'y en a pas, le $\top$ ne s'élimine pas.)
- (Introduction du $\top$.) $\star$ est une démonstration de $\top$.
- (Élimination du $\vee$, ou division en cas.) Lorsqu'on démontre R à partir d'une hypothèse de type $P \vee Q$, on peut écrire cette hypothèse $u : P|v : Q$ et fournir une démonstration f de R à partir de u démonstration de P et une démonstration g de R à partir de v démonstration de Q, auquel cas la démonstration finale s'écrit $\lambda[u : P|v : Q].(f|g)$.
- (Introduction du $\vee$.) Si u est une démonstration de P alors $u^{\to Q}$ est une démonstration de $P \vee Q$. Si v est une démonstration de Q alors $v^{P\leftarrow}$ est une démonstration de $P \vee Q$.
- (Élimination du $\perp$.) $\lambda[].()^P$ est une démonstration de $\perp \Rightarrow P$.
- (Introduction du $\perp$: il n'y en a pas, le $\perp$ ne s'introduit pas.)
- (Élimination du $\forall$, ou spécialisation.) Si δ est une démonstration de $\forall(t : A).P$ alors δa est une démonstration de $P_a = (\lambda[t : A].P)a$ pour tout terme a de type A.
- (Introduction du $\forall$, ou généralisation.) Si δ est une démonstration de P qui fait intervenir une variable t de type A, alors $\lambda[t : A].\delta$ est une démonstration de $\forall(t : A).P$. Ceci suppose qu'aucune des hypothèses de δ ne fait intervenir t.
- (Élimination du $\exists$, ou choix d'un témoin.) Si δ est une démonstration de Q (ne faisant pas intervenir t) à partir d'une hypothèse affirmant $\exists(t : A).P$, alors $(t?\delta)$ est une démonstration de Q à partir d'une hypothèse affirmant P.
- (Introduction du $\exists$.) Si a est un terme de type A et δ une démonstration de $P_a = (\lambda[t : A].P)a$, alors (a, δ) est une démonstration de $\exists[t : A].P$.

Comme premier exemple, nous démontrons

$$\forall(p : \Omega^{U \times V}).(\exists(x : U).\forall(y : V).p(x, y) \Rightarrow \forall(y : V).\exists(x : U).p(x, y))$$

Pour cela, donnons-nous un p de $\Omega^{U \times V}$. Si δ est une hypothèse qui affirme que $\forall(y : V).p(x, y)$, où x est un terme de type U, et que y est un terme de type V, alors $(x, \delta y)$

[1] Certes, cette règle n'est pas géniale. Il y aurait de meilleures façons de procéder. Mais c'est de peu d'importance somme toute.

est une démonstration de $\exists(x:U).p(x,y)$, donc $\lambda[y:V].(x,\delta y)$ est une démonstration de $\forall(y:V).\exists(x:U).p(x,y)$. Comme x n'est plus libre dans cette assertion, on peut éliminer $\exists$ et dire que $(x?\lambda[y:V].(x,\delta y))$ est une démonstration de la même chose mais à partir de δ hypothèse qui affirme $\exists(x:U).\forall(y:V).p(x,y)$ cette fois. Et finalement, la démonstration voulue est

$$\lambda[p:\Omega^{U\times V}].\lambda[\delta:\exists(x:U).\forall(y:V).p(x,y)].(x?\lambda[y:V].(x,\delta y))$$

Nous montrons maintenant le fait que l'égalité est symétrique. Pour cela il faut d'abord définir l'égalité. Cela se fait en introduisant la notation $a=b$ (où a et b sont deux termes d'un même type U quelconque) comme signifiant $\forall(p:\Omega^U).(pa\Rightarrow pb)$. **Attention :** Cette égalité interne entre termes n'a pas de raison de coïncider avec l'égalité externe (égalité entre flèches de la catégorie). En fait, cela n'a pas vraiment de sens, puisque $a=b$ est un élément de Ω tel que nous l'avons défini, pour l'égalité interne.

On veut donc montrer que

$$\forall(a:U).\forall(b:U).((\forall(p:\Omega^U).(pa\Rightarrow pb))\Rightarrow(\forall(q:\Omega^U).(qb\Rightarrow qa)))$$

Soient donc a de type U, b de type U et δ une hypothèse affirmant que $\forall(p:\Omega^U).(pa\Rightarrow pb)$. Considérons $\lambda[x:U].\forall(q:\Omega^U).(px\Rightarrow pa)$. C'est un terme de type Ω^U. On peut donc considérer $\delta(\lambda[x:U].\forall(q:\Omega^U).(qx\Rightarrow qa))$, qui est une démonstration de $(\forall(q:\Omega^U).(qa\Rightarrow qa))\Rightarrow(\forall(q:\Omega^U).(qb\Rightarrow qa))$. Or on peut trouver facilement une démonstration de $\forall(q:\Omega^U).(qa\Rightarrow qa)$. Par exemple, $\lambda[q:\Omega^U].\lambda[\xi:qa].\xi$ en est une. Finalement, la démonstration recherchée est

$$\begin{gathered}\lambda[a:U].\lambda[b:U].\lambda[\delta:\forall(p:\Omega^U).(pa\Rightarrow pb)].\\(\delta(\lambda[x:U].\forall(q:\Omega^U).(qx\Rightarrow qa))(\lambda[q:\Omega^U].\lambda[\xi:qa].\xi))\end{gathered}$$

Voici la chose écrite en Coq.

```
Welcome to Coq V6.2.2 (September 1998)
No .coqrc or .coqrc.6.2.2 found. Skipping rcfile loading.

Coq < Parameter U:Type.
U is assumed

Coq < Theorem symmetry :
Coq < (a,b:U)(((p:U->Prop)((p a)->(p b)))->
Coq < ((q:U->Prop)((q b)->(q a)))).
1 subgoal

  ============================
   (a,b:U)((p:U->Prop)(p a)->(p b))->(q:U->Prop)(q b)->(q a)
```

```
symmetry < Intro a; Intro b; Intro delta.
1 subgoal

  a : U
  b : U
  delta : (p:U->Prop)(p a)->(p b)
  ============================
   (q:U->Prop)(q b)->(q a)

symmetry < Cut (q:U->Prop)(q a)->(q a).
2 subgoals

  a : U
  b : U
  delta : (p:U->Prop)(p a)->(p b)
  ============================
   ((q:U->Prop)(q a)->(q a))->(q:U->Prop)(q b)->(q a)

subgoal 2 is:
 (q:U->Prop)(q a)->(q a)

symmetry < Exact (delta ([x:U](q:U->Prop)(q x)->(q a))).
1 subgoal

  a : U
  b : U
  delta : (p:U->Prop)(p a)->(p b)
  ============================
   (q:U->Prop)(q a)->(q a)

symmetry < Intro q; Intro xi.
1 subgoal

  a : U
  b : U
  delta : (p:U->Prop)(p a)->(p b)
  q : U->Prop
  xi : (q a)
  ============================
```

```
    (q a)

symmetry < Exact xi.
Subtree proved!

symmetry < Qed.
(Intro a; Intro b; Intro delta).
Cut (q:U->Prop)(q a)->(q a).
Exact (delta [x:U](q:U->Prop)(q x)->(q a)).

(Intro q; Intro xi).
Exact xi.

symmetry is defined

Coq < Print symmetry.
symmetry =
[a,b:U]
 [delta:(p:U->Prop)(p a)->(p b)]
  (delta [x:U](q:U->Prop)(q x)->(q a) [q:U->Prop][xi:(q a)]xi)
     : (a,b:U)((p:U->Prop)(p a)->(p b))->(q:U->Prop)(q b)->(q a)


Coq < Quit.
```
Accessoirement, la réflexivité et la transitivité de l'égalité sont beaucoup plus faciles.