Structures mathématiques du langage

Alain Lecomte
Université Paris 8 - Vincennes-Saint-Denis
Master LFT

1 Du calcul de Lambek aux catégories

La théorie des catégories a été inventée [1] ([MacLane 1971]) pour apporter un nouveau type de fondement aux mathématiques, qui soit plus général que ne l'est la théorie des ensembles. La théorie des ensembles est basée sur la relation d'appartenance ($\in$) entre *éléments* et entités appelées *ensembles*. La notion d'ensemble apparaît très naturelle. Cependant, vue d'une manière trop intuitive, elle conduit à des antinomies fatales (dont celle de Russell (1902), fameuse, de l'ensemble des ensembles qui ne s'appartiennent pas à eux-mêmes). Pour les éviter, il a fallu formuler une théorie axiomatique dont la version la plus connue et la plus utilisée est due à Zermelo-Fraenkel (on parle de théorie ZF), qui axiomatise la relation d'appartenance. On peut toutefois avoir envie de se débarrasser de cette relation, comme de la notion *d'élément*, de façon à obtenir une théorie dont les objets se comporteraient tout à fait comme des ensembles, même s'ils ne sont pas définis à partir d'éléments. On obtient alors[2] un type très général de structure abstraite que l'on appelle un *topos* et comme ce type de structure est beaucoup plus général que la structure ensembliste, on gagne en abstraction et en puissance fondatrice. Pour formuler ce que l'on entend par un topos, il faut d'abord toutefois passer par la théorie des catégories, qui donne, avant d'atteindre les topos, des types de structures associés à des théories plus simples que celle des ensembles, comme le sont les diverses logiques propositionnelles (classique et intuitionniste notamment). Il est intéressant de regarder si les grammaires, en particulier catégorielles, trouvent leur place dans cette gamme de structures.
On notera par ailleurs que, les catégories se situant à un niveau d'abstraction plus élevé que les ensembles et que les différentes structures algébriques que l'on peut définir à partir d'eux (monoïdes, groupes, anneaux, corps, ensembles partiellement ordonnés etc.), il deviendra possible grâce à elles d'exprimer de façon générale les genres de constructions auxquelles on procède régulièrement pour toutes ces structures (construire une sous-structure, un produit, une somme etc.). On "factorise" en quelque sorte le travail en définissant des notions de *produit*, de *somme*, de *limite* etc. qui sont ensuite applicables à chacune de ces structures. On peut même envisager des constructions nouvelles.
[Lambek(1988)] établit le rapport entre les catégories au sens des grammaires dites "catégorielles" et les catégories au sens de la théorie des catégories. On pourrait dire entre une logique *catégoriale* et une logique *catégorique*. Il faut pour cela opérer juste un léger changement de point de vue sur les dérivations syntaxiques. Au lieu de se baser sur les types (formules) et de dire qu'il existe une relation de déduction entre les types comme vu initialement dans le chapitre sur le calcul de Lambek d'origine, on considère chaque relation de déduction comme une *flèche*, c'est-à-dire en termes catégoriques comme un *morphisme*. Les types ou formules font alors figure *d'objets*. Mais avant d'en venir là, il faut voir quelques

[1] A vrai dire, si la théorie des catégories a été inventée par Eilenberg et McLane en 1945, c'est Lawvere qui, en 1964, a perçu qu'elle pouvait servir de fondement aux mathématiques.
[2] A la suite des travaux de A. Grothendieck, souvent présenté comme le plus grand mathématicien du XXème siècle.

notions de la théorie des catégories.

2 Introduction à la théorie des catégories

2.1 Notions de base

2.1.1 Catégories et foncteurs

Pour introduire la notion de catégorie, nous sommes plus ou moins obligés de nous appuyer sur celle d'ensemble, même si par la suite, la théorie des catégories est utilisée comme fondation, y compris de la théorie des ensembles. Donc, au début, nous prenons comme exemples des ensembles, pouvant être munis d'une structure (monoïde, groupe, anneau, corps, espace vectoriel etc.). Nous savons que, dans tous ces cas, se détache, outre la notion de structure, celle d'homomorphisme. Par exemple, si $(G, \star)$ et $(G', \bullet)$ sont des groupes, un homorphisme de groupe sera une application f de G vers G' telle que :

- $f(e_G) = e'_G$ (e_X étant l'élement neutre du groupe X)
- $f(x \star x') = f(x) \bullet f(x')$

f sera un isomorphisme s'il existe un homorphisme f' de G' vers G tel que :

$$f \circ f' = I_G \quad et \quad f' \circ f = I_{G'}$$

où I_X est l'identité de X (qui est bien sûr un isomorphisme).
De façon générale, une **catégorie** est une collection d'*objets* et de *morphismes*, un morphisme étant en général représenté par une *flèche* et un objet par un *sommet*, ce qui fait qu'en première approximation, une catégorie n'est rien d'autre qu'un graphe. Simplement, parmi les flèches, il doit nécessairement y en avoir une particulière pour chaque objet A, qu'on appelle la flèche *identité* $1_A : A \to A$. Les flèches peuvent d'autre part se composer, ce qui fait que pour tout couple formé d'une flèche $f : A \to B$ et d'une flèche $g : B \to C$, il existe une flèche $gf : A \to C$. Tout cela de telle sorte que :

- pour tous A et B et pour tout $f : A \to B$: $f1_A = f = 1_B f$
- pour toutes $f : A \to B$, $g : B \to C$ et $h : C \to D$: $(hg)f = h(gf)$

A titre d'exemples de catégories :

- la catégorie **Ens**, dont les objets sont les ensembles et les flèches les applications entre ensembles,
- la catégorie **Mon**, dont les objets sont les monoïdes et les flèches les homomorphismes de monoïdes,
- la catégorie **PreOrd**, dont les objets sont les ensembles pré-ordonnés et dont les flèches sont les applications monotones

On peut aussi noter que certains ensembles bien connus sont eux-mêmes des mini-catégories :

- un simple ensemble A est une catégorie : ses objets sont ses éléments et les seules flèches sont les "boucles" 1_a pour tout $a \in A$,
- un monoïde est une catégorie : elle a un seul objet et les flèches sont les éléments du monoïde. La flèche identité est l'élément neutre et l'opération de composition est la composition des flèches[3],
- un ensemble pré-ordonné est une catégorie : les objets sont ses éléments et les flèches sont les couples (a, b) tels que $a \preceq b$

Parmi les catégories, on peut situer **Cat**, dont les objets sont justement... les catégories (!). Les flèches de cette catégorie un peu spéciale sont les *foncteurs*.

Définition 1 *Un* **foncteur** $F : \mathcal{A} \to \mathcal{B}$ *est un morphisme de graphes, c'est-à-dire une application définie sur les objets et les morphismes de* $\mathcal{A}$ *de sorte que pour toute flèche* $f : A \to B$*, on ait :* $F(f) : F(A) \to F(B)$*, avec :*

[3] Soit par exemple un monoïde libre sur $\mathcal{A}$, à chaque lettre a correspond une flèche $a : \bullet \to \bullet$, $\bullet$ étant le seul objet de la catégorie. Un mot est alors la composition des flèches associées aux lettres qui le composent.

- $F(1_A) = 1_{F(A)}$
- $F(gf) = F(g)F(f)$

De nombreux objets mathématiques peuvent être vus comme des foncteurs. Par exemple on peut définir *List* comme un foncteur de **Ens** dans **Ens**. A chaque ensemble S (donc chaque *objet* de **Ens**), *List* associe un autre ensemble : $List(S)$ qui s'interprète comme l'ensemble des listes d'éléments de S. Maintenant, à chaque application $f : S \to S'$ (*morphisme*), *List* associe $List(f) : List(S) \to List(S')$ qui associe à toute liste d'éléments de S la liste d'éléments de S' qui est : $[f(s_1), f(s_2), ..., f(s_n)]$. Dans un langage comme LISP, $List(f)$ est noté maplist(f).
On peut, de plus, dire que chaque objet $List(S)$ est doté d'une concaténation associative "$\frown$", et d'une liste vide [ ] (élément neutre pour "$\frown$"). C'est donc un monoïde, donc un objet de la catégorie **Mon**. *List* est donc un foncteur de **Ens** dans **Mon**. $List(f)$ est un homomorphisme de monoïdes, ce qui se traduit par le fait que :

$$\begin{array}{lcl} List(f)([\,]) & = & [\,] \\ List(f)(L_1^\frown L_2) & = & List(f)(L_1)^\frown List(f)(L_2) \end{array}$$

pour assurer que ces équations définissent bien maplist de manière unique, on rajoute :

$$List(f)[s] \quad = \quad [f(s)]$$

2.1.2 Transformations naturelles

Définition 2 *Etant donnés deux foncteurs F et G de $\mathcal{A}$ dans $\mathcal{B}$, une transformation naturelle τ de F dans G est une famille de flèches $\tau(A) : F(A) \to G(A)$ dans $\mathcal{B}$, une flèche pour chaque objet A de $\mathcal{A}$, telle que le carré suivant commute pour toute flèche $f : A \to B$ dans $\mathcal{A}$:*

$$\begin{tikzcd} F(A) \arrow[r, "\tau(A)"] \arrow[d, "F(f)"'] & G(A) \arrow[d, "G(f)"] \\ F(B) \arrow[r, "\tau(B)"] & G(B) \end{tikzcd}$$

autrement dit telle que :

$$G(f)\tau(A) = \tau(B)F(f)$$

Quand des objets mathématiques usuels sont vus comme des catégories, les morphismes entre ces objets sont justement les transformations naturelles. Par exemple, tout ensemble peut être vu comme un foncteur de $\{\bullet\}$ dans **Ens**, où $\{\bullet\}$ est la catégorie avec un seul objet et le morphisme identité, il s'agit, pour un objet A de **Ens**, du foncteur F_A tel que $F_A(\bullet) = A$ et pour 1 : $F_A(1) = 1_A$. Si f est une application $A \to B$, f est aussi la transformation naturelle de F_A dans F_B qui, étant donné le seul objet $\bullet$, est telle que $f : F_A(\bullet) \to F_B(\bullet)$.
D'autre part, la notion de transformation naturelle nous permet de faire des constructions intéressantes. Soit en effet deux catégories $\mathcal{A}$ et $\mathcal{B}$, qu'est-ce que $\mathcal{B}^{\mathcal{A}}$? Par analogie avec ce qui se passe pour des ensembles, on a envie de dire qu'il s'agit de l'ensemble des "fonctions" de $\mathcal{A}$ dans $\mathcal{B}$. Mais comme ce que nous voulons obtenir, c'est une catégorie, il faut à la fois définir des objets et des flèches. Pour les objets, il est évident qu'il s'agira des foncteurs de $\mathcal{A}$ dans $\mathcal{B}$. Quant aux flèches, il faut que ce soit des "applications" de foncteurs sur d'autres, donc il s'agit des *transformations naturelles* entre foncteurs.
Un *isomorphisme* entre catégories $\mathcal{A}$ et $\mathcal{B}$ est un foncteur qui possède un inverse, autrement dit un foncteur F de $\mathcal{A}$ dans $\mathcal{B}$ tel qu'il existe G de $\mathcal{B}$ dans $\mathcal{A}$ avec : $GF = 1_{\mathcal{A}}$ et $FG = 1_{\mathcal{B}}$.
(le foncteur identité $1_{\mathcal{A}}$ est le foncteur qui laisse les objets et les morphismes inchangés).
Ces définitions permettent de démontrer les isomorphismes suivants :

– $\mathcal{C}^{\mathcal{A}\times\mathcal{B}} \cong (\mathcal{C}^{\mathcal{B}})^{\mathcal{A}}$
– $(\mathcal{A}\times\mathcal{B})^{\mathcal{C}} \cong \mathcal{A}^{\mathcal{C}} \times \mathcal{B}^{\mathcal{C}}$
Nous pouvons ainsi utiliser des catégories de foncteurs.
A titre d'exemples de foncteurs particuliers, on peut envisager les foncteurs Hom.

Définition 3 *Si A et B sont deux objets d'une catégorie $\mathcal{A}$, on note $Hom_{\mathcal{A}}(A, B)$ la classe de toutes les flèches $A \to B$. Si c'est un ensemble, pour chaque A et B, alors la catégorie $\mathcal{A}$ est dite localement petite.*

Exemples de foncteurs : (NB : $\mathcal{A}^{op}$ est la catégorie $\mathcal{A}$ où toutes les flèches ont été inversées)
Si $\mathcal{A}$ est une catégorie localement petite, alors il existe un foncteur Hom : $\mathcal{A}^{op} \times \mathcal{A} \to$**Ens**. Pour une flèche $(g, h) : (A, B) \to (A', B')$ de $\mathcal{A}^{op} \times \mathcal{A}$, où : $g : A' \to A$ et $h : B \to B'$ dans $\mathcal{A}$, Hom(g, h) envoie $f \in$Hom(A, B) sur $hfg \in$Hom(A', B').

$$\begin{tikzcd} A' \arrow[r, "Hom(g,h)(f)"] \arrow[d, "g"'] & B \\ A \arrow[r, "f"] & B \arrow[u, "h"'] \end{tikzcd}$$

On obtient aussi un foncteur Hom* : $\mathcal{A}^{op} \to$**Ens**$^{\mathcal{A}}$, ainsi qu'un foncteur Hom$^*_{op}$: $\mathcal{A} \to$**Ens**$^{\mathcal{A}^{op}}$. Par exemple, pour ce dernier, si on reprend le diagramme précédent, Hom$^*_{\mathcal{A}^{op}}$ est le foncteur tel que pour toute $h : B \to B'$ dans $\mathcal{A}$, Hom$^*_{\mathcal{A}^{op}}(h)$ est le foncteur de $\mathcal{A}^{op}$ dans **Ens** qui envoie la flèche g sur l'ensemble Hom(g, h).

Définition 4 *Une sous-catégorie $\mathcal{C}$ d'une catégorie $\mathcal{B}$ est une catégorie dont les objets et les flèches sont contenus dans ceux de $\mathcal{B}$ et qui est fermée sous les opérations de source, but, identité et composition. $\mathcal{C}$ est dite pleine si pour tous objets C, C' de $\mathcal{C}$, $Hom_{\mathcal{C}}(C, C')$ = $Hom_{\mathcal{B}}(C, C')$*

Pour des transformations naturelles, on écrit Nat(F, G) au lieu de Hom(F, G). Autrement dit, Nat(F, G) est l'ensemble des transformations naturelles de F vers G (si c'est bien un ensemble).

2.1.3 Limites

L'un des concepts centraux de la théorie des catégories est fourni par la notion de *limite*. Elle permet la construction de nombreuses sortes d'objets, dans chaque catégorie, qui sont toujours définis à un isomorphisme près. Un premier exemple est fourni par les objets dits terminaux.

Définition 5 *Un objet 1 d'une catégorie $\mathcal{A}$ est dit être un* **objet terminal** *si pour chaque objet A de $\mathcal{A}$, il existe une flèche unique $\bigcirc_A : A \to 1$.*

On voit facilement que la notion d'objet terminal est définie à un isomorphisme près : en effet, soit 1' un autre objet terminal, alors comme 1 et 1' sont en particulier des objets, d'après la définition, il existe une et une seule flèche $(\bigcirc_1)$ de 1 vers 1' et une et une seule flèche, $(\bigcirc_{1'})$, de 1' vers 1. La composition de ces deux flèches donne :

$$\bigcirc_1 \circ \bigcirc_{1'} : 1' \to 1' \qquad \bigcirc_{1'} \circ \bigcirc_1 : 1 \to 1$$

Comme la définition implique également l'unicité de toute flèche de 1 dans 1, comme de $1'$ dans $1'$, et donc son égalité avec, respectivement, 1_1 et $1_{1'}$, on obtient :

$$\bigcirc_1 \circ \bigcirc_{1'} = 1_{1'} \qquad \bigcirc_{1'} \circ \bigcirc_1 = 1_1$$

d'où l'isomorphisme.
Exemple : dans la catégorie **Ens**, tout singleton est terminal, et ils sont tous isomorphes entre eux. Noter

qu'en effet, il existe une et une seule application d'un ensemble quelconque vers un singleton $\{a\}$ (l'application constante qui associe a à chaque élément de l'ensemble de départ, si celui-ci n'est pas vide. S'il est vide, on sait qu'il existe une et une seule application de $\emptyset$ dans n'importe quel ensemble, donc en particulier aussi dans $\{a\}$).

Définition 6 *Un objet 0 d'une catégorie $\mathcal{A}$ est dit être un* **objet initial** *si pour chaque objet A de $\mathcal{A}$, il existe une flèche unique $0 \to A$. C'est bien sûr l'objet terminal de $\mathcal{A}^{op}$.*

Exemple : dans **Ens**, $\emptyset$ est initial.
Un élément d'un ensemble S (dans la catégorie **Ens**) peut être vu comme une flèche $1 \to S$. On appelle une telle flèche : une **constante** de S.
Une construction très fréquente, pratiquée pour chaque structure algébrique (voire topologique), est celle de produit (de telle sorte que le produit - cartésien - de deux ensembles soit un ensemble, le produit de deux groupes soit un groupe, celui de deux espaces topologiques soit un espace topologique etc.). En fait, il y a un lien entre les produits en général et la notion d'objet terminal.

Définition 7 *Un* **produit** *de deux objets A et B consiste en la donnée d'un objet noté $A \times B$ et de deux flèches, appelées* **projections** *$\pi_1 : A \times B \to A$ et $\pi_2 : A \times B \to B$, tels que pour tout objet C et tout couple de flèches $f : C \to A$ et $g : C \to B$, il existe exactement une flèche $< f, g >: C \to A \times B$ telle que le diagramme suivant commute.*

$$\begin{tikzcd}
& C \arrow[dl, "f"'] \arrow[d, dashed, "{< f, g >}"] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B
\end{tikzcd}$$

Etant donnés A et B deux objets d'une catégorie $\mathcal{A}$, considérons la catégorie des paires de flèches $< f : C \to A, g : C \to B >$, la définition précédente fait que le produit, donné par la paire $< \pi : A \times B \to A, \pi' : A \times B \to B >$ est terminal dans cette catégorie.
Par dualité (cf. la relation existante entre objet terminal et objet initial), on passe de la notion de produit à celle de *co-produit*. On obtient :

Définition 8 *Un* **coproduit** *de deux objets A et B consiste en la donnée d'un objet noté $A + B$ et de deux flèches, appelées* **injections** *$\iota_1 : A \to A + B$ et $\iota_2 : B \to A + B$, tels que pour tout objet C et tout couple de flèches $f : A \to C$ et $g : B \to C$, il existe exactement une flèche $[f, g] : A + B \to C$ telle que le diagramme suivant commute.*

$$\begin{tikzcd}
A \arrow[r, "\iota_1"'] \arrow[dr, "f"'] & A + B \arrow[d, dashed, "{[f, g]}"] & B \arrow[l, "\iota_2"] \arrow[dl, "g"] \\
& C &
\end{tikzcd}$$

Exemple :
- dans **Ens**, le coproduit correspond à la somme disjointe.

La somme disjointe $A \oplus B$ est en effet munie de deux applications, les injections canoniques :

$$\iota_1 : A \to A \oplus B \qquad \iota_2 : B \to A \oplus B$$

et il est vrai que pour tout couple de flèches $f : A \to C$, $g : B \to C$, on peut définir $[f, g]$ de $A \oplus B$ dans C de manière que $[f, g]\iota_1 = f$ et $[f, g]\iota_2 = g$, il suffit de prendre la fonction définie par :

$$\begin{array}{rcl} \forall x \in A \oplus B, [f, g](x) & = & f(x) \; si \; x \in A \\ & = & g(x) \; si \; x \in B \end{array}$$

Cette fonction est parfaitement définie puisqu'il n'y a pas d'élément commun aux deux membres d'une somme disjointe.

Définition 9 *Une flèche $e : X \to A$ est dite être un* **égaliseur** *de deux flèches $f : A \to B$ et $g : A \to B$ si :*

1. $fe = ge$
2. *pour toute flèche $e' : X' \to A$ telle que $fe' = ge'$, il existe une unique flèche $k : X \to X'$ telle que $ek = e'$*

$$\begin{tikzcd} X \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\ X' \arrow[u, dashed, "k"] \arrow[ur, "e'"'] & & \end{tikzcd}$$

Exemple :

– dans **Ens**, étant donnés deux applications f et g de A dans B, soit X le sous-ensemble de A où f et g sont égales. La fonction d'inclusion $e : X \to A$ est un égaliseur.

La construction duale (de co-égaliseur) fournit l'équivalent de la notion de relation d'équivalence.

On définit également, de façon générale, la notion de *limite* et on montre que les notions de produit et de co-produit se ramènent à elle.

Définition 10 *Soit $\mathcal{A}$ une catégorie, et* **D** *un diagramme quelconque sur $\mathcal{A}$, dont les objets sont notés D_i. un* **cône** *pour* **D** *est un objet X de $\mathcal{A}$ et une famille de flèches $f_i : X \to D_i$ tels que le diagramme suivant commute, pour tous i et j.*

$$\begin{tikzcd} & X \arrow[dl, "f_i"'] \arrow[dr, "f_j"] & \\ D_i \arrow[rr, "g"'] & & D_j \end{tikzcd}$$

Définition 11 *Un cône est une* **limite** *si, de plus, pour chaque autre cône $\{f'_i : X' \to D_i\}$, il existe une flèche unique $k : X' \to X$ telle que le diagramme suivant commute, pour tout D_i.*

$$\begin{tikzcd} X' \arrow[rr, dashed, "k"] \arrow[dr, "f'_i"'] & & X \arrow[dl, "f_i"] \\ & D_i & \end{tikzcd}$$

Exemples :

- si, dans $\mathcal{A}$, le diagramme se limite aux deux objets : A et B, un cône pour ce diagramme est la donnée d'un diagramme :

$$\begin{tikzcd} A & X \arrow[l, "f"'] \arrow[r, "g"] & B \end{tikzcd}$$

  Si ce cône est une limite, alors on retrouve exactement la notion de produit.
- si **D** est le diagramme :

$$\begin{tikzcd} & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd}$$

  alors un cône pour **D** est un P muni des flèches f', h, g' tel que le diagramme suivant commute :

$$\begin{tikzcd} P \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, "h"] & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd}$$

  Si un tel cône est une limite, on obtient un **pullback**. Les pullbacks donneront lieu à la notion importante de *produit fibré*.

La notion duale de celle de cône est celle de **cocône**, et de limite : **co-limite**.

2.1.4 Adjonction

Il est usuel d'avoir dans de nombreuses structures, une situation où deux foncteurs sont, en un certain sens, "inverses" l'un de l'autre. Tout le monde a appris sur les bancs de l'école que, dès que nous avons la notion de *multiplication*, nous avons celle de *division*, avec évidemment, sur $\mathbb{R}^+$, la relation :

$$a \times b \leq c \Leftrightarrow b \leq \frac{c}{a}$$

La multiplication par a (notons-la m_a) et la division par a (notons-la d_a) entretiennent donc une relation évidente, qu'on peut formuler comme suit :

$$m_a(b) \leq c \Leftrightarrow b \leq d_a(c)$$

De la même façon, le calcul de Lambek admet les équivalences :

$$A \bullet B \to C \Leftrightarrow A \to C/B \Leftrightarrow B \to A \setminus C$$

Considérons $F_A(B) = A \bullet B$ et $G_A(B) = A \setminus B$, nous avons alors :

$$F_A(B) \to C \Leftrightarrow B \to G_A(C)$$

Dans les deux cas, nous avons une structure commune, que l'on nomme *adjonction* (ou aussi parfois, *correspondance de Galois*). Par le biais de l'adjonction, on retrouvera les notions définies plus haut.

Définition 12 *Une* **adjonction** *entre des catégories $\mathcal{C}$ et $\mathcal{D}$ consiste en :*

- *deux foncteurs $F : \mathcal{C} \to \mathcal{D}$ et $G : \mathcal{D} \to \mathcal{C}$*
- *une transformation naturelle $\eta : I_{\mathcal{C}} \to (G \circ F)$*

tels que pour tout objet X de $\mathcal{C}$ et toute flèche f de $\mathcal{C}$: $f : X \to G(Y)$, il existe une et une seule flèche de $\mathcal{D}$: $f^* : F(X) \to Y$ telle que le diagramme suivant commute :

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[dr, "f"'] & G(F(X)) \arrow[d, "G(f^*)"] \\ & G(Y) \end{tikzcd}$$

Si nous nous plaçons donc dans une seule catégorie, dont les objets sont les types du calcul de Lambek et les morphismes les déductions entre types, alors en définissant :

- pour chaque type A, soit $F_A : B \mapsto F_A(B) = A \bullet B$
- pour chaque type A, soit $G_A : B \mapsto G_A(B) = A \setminus B$

la définition de l'adjoint conduit à :
- pour toute $f : X \to A \setminus Y$, il existe une flèche f^* unique, $f^* : A \bullet X \to Y$ telle que $f = G_A(f^*) \circ \eta_X$, où, à X, η_X associe $A \setminus (A \bullet X)$.
Ce qu'on écrira de la manière résumée suivante :

$$\frac{f : X \to A \setminus Y}{f^* : A \bullet X \to Y}$$

qui est exactement une des règles mentionnées ci-dessus. Nous y reviendrons plus loin.
La transformation naturelle η est appelée *unité*.
La construction ci-dessus possède sa duale :

$$\begin{tikzcd} F(G(Y)) \arrow[r, "\epsilon_Y"] & Y \\ F(X) \arrow[u, dashed, "F(g^*)"] \arrow[ur, "g"'] & \end{tikzcd}$$

où ϵ est la co-unité. Il est possible de prouver que l'existence d'une unité entraîne celle d'une co-unité, et réciproquement, ce qui donne deux points de vue possibles pour regarder l'adjonction.
Du point de vue du calcul de Lambek, ϵ_Y est la transformation naturelle qui, à toute formule $A \bullet (A \setminus Y)$ associe Y.
La règle d'inférence déduite est la réciproque de la précédente :

$$\frac{g : A \bullet X \to Y}{g^* : X \to A \setminus Y}$$

Exercice : Soit $\mathcal{C}$ =**Ens** et $\mathcal{D}$ =**Mon**. On prend pour η, la famille d'applications qui, à chaque ensemble S, associe l'ensemble $List(S)$ des listes de S, pour F la fonction $List$ et pour G le foncteur d'oubli (c'est-à-dire le foncteur trivial de **Mon** dans **Ens** qui "oublie" la structure de monoïde).

- Démontrer qu'on obtient une adjonction.
- Démontrer que, en prenant $(\mathbb{N}, +, 0)$ comme objet de **Mon** et 1 la fonction constante de S dans $\mathbb{N}$ qui, à tout élément de S associe 1, la fonction unique 1^* définie par l'adjonction n'est autre que la fonction *longueur* !

2.1.5 Monades

On a vu qu'on pouvait toujours considérer un ensemble pré-ordonné $(A, \leq)$ comme une catégorie $\mathcal{A}$. Un foncteur entre deux telles catégories, $(A, \leq)$ et $(B, \leq)$, est alors une application monotone : si $a \leq a'$ alors $F(a) \leq F(a')$. Soit un tel foncteur F, et soit G dans l'autre direction, qui est son adjoint à droite (autrement dit, pour tous $a \in A$ et $b \in B$, $F(a) \leq b$ si et seulement si $a \leq G(b)$), alors on peut voir facilement que pour tout $a, a' \in A$,

1. $a \leq GF(a)$
2. $GFGF(a) \leq GF(a)$
3. si $a \leq a'$ alors $GF(a) \leq GF(a')$

(1) : on a $F(a) \leq F(a)$ donc $a \leq GF(a)$.
(2) : $GF(a) \leq GF(a)$ donc en faisant migrer le G de droite vers la gauche (il devient alors un F) : $FGF(a) \leq F(a)$, et en appliquant G aux deux termes : $GFGF(a) \leq GF(a)$.
GF est alors typiquement ce que l'on appelle une *fermeture*. Les points précédents se traduisent en :

1. $a \subset \overline{a}$
2. $\overline{\overline{a}} \subset \overline{a}$
3. si $a \subset a'$ alors $\overline{a} \subset \overline{a'}$

De même FG permet de définir une opération d'*intérieur*. NB : il est clair que (1) et (2) ci-dessus entraînent $GFGF(a) \cong GF(a)$ (on ne mettrait "=" que si la relation $\leq$ était bien une relation d'ordre).
Si on appelle *fermés* de A les éléments a tels que $GF(a) \cong a$ et *ouverts* de B les éléments b de B tels que $FG(b) \cong b$, alors F et G déterminent une équivalence entre l'ensemble préordonné $\mathcal{A}_0$ des fermés de A et l'ensemble préordonné $\mathcal{B}_0$ des ouverts de B, selon le diagramme suivant :

$$\begin{tikzcd}
\mathcal{A} \arrow[r, shift left, "F"] & \mathcal{B} \arrow[l, shift left, "G"] \\
\mathcal{A}_0 \arrow[u, "inclusion"] \arrow[r, "equivalence"] & \mathcal{B}_0 \arrow[u, "inclusion"']
\end{tikzcd}$$

Ces notions de fermeture et d'intérieur se rencontrent souvent en mathématiques, notamment bien sûr en topologie. Notons $T = GF$. T est appelé une *monade*.

Définition 13 *Une* **monade** *sur une catégorie $\mathcal{A}$ est un triplet (T, η, μ), où T est un foncteur $\mathcal{A} \to \mathcal{A}$ et où :*

– *η est une transformation naturelle $1_{\mathcal{A}} \to T$*
– *μ est une transformation naturelle $T^2 \to T$*

de telle sorte que les diagrammes suivants commutent :

$$\begin{tikzcd}
T \arrow[r, "\eta T"] \arrow[dr, "="'] & T^2 \arrow[d, "\mu"] & T \arrow[l, "T\eta"'] \arrow[dl, "="] \\
& T &
\end{tikzcd}
\qquad
\begin{tikzcd}
T^3 \arrow[r, "T\mu"] \arrow[d, "\mu T"] & T^2 \arrow[d, "\mu"] \\
T^2 \arrow[r, "\mu"'] & T
\end{tikzcd}$$

autrement dit satisfaisant les équations :

$$\mu \circ T\eta = 1_T = \mu \circ \eta T, \qquad \mu \circ \mu T = \mu \circ T\mu$$

Appliquée au cas des ensembles pré-ordonnés, T est une application monotone de A dans A, il y a pour chaque élément a, un morphisme η vers $T(a)$, autrement dit $a \leq T(a)$, et pour chaque objet a, un morphisme de $TT(a)$ dans $T(a)$, autrement dit $TT(a) \leq T(a)$. Les équations disent qu'en partant de a, η donne un $T(a)$ et que μ appliqué à $TT(a)$ redonne bien le $T(a)$ initial, et qu'en partant de $TT(a)$, on obtient le même $T(a)$ en suivant l'un ou l'autre des deux chemins indiqués.
Noter que cette notion de *monade* ressemble beaucoup à, voire généralise, la notion de monoïde.
Un monoïde M peut en effet être vu comme un ensemble M muni de deux fonctions :

$$\eta : 1 \to M \qquad \mu : M \times M \to M$$

telles que les diagrammes suivants commutent :

$$\begin{tikzcd}
1 \times M \arrow[r, "\eta \times 1"] \arrow[dr, "\lambda"'] & M \times M \arrow[d, "\mu"] & M \times 1 \arrow[l, "1 \times \eta"'] \arrow[dl, "\rho"] \\
 & M &
\end{tikzcd}
\qquad
\begin{tikzcd}
M \times M \times M \arrow[r, "1 \times \mu"] \arrow[d, "\mu \times 1"] & M \times M \arrow[d, "\mu"] \\
M \times M \arrow[r, "\mu"'] & M
\end{tikzcd}$$

1 est interprété comme le singleton $\{\emptyset\}$, donc η est la fonction qui à $\emptyset$ associe le mot vide ϵ. Dans $1 \times \mu$, 1 est interprété comme la fonction identité $M \to M$. λ (resp. ρ) est la fonction (bijective) de $1 \times M$ (resp. $M \times 1$) dans M qui, à tout couple $(\emptyset, x)$ (resp. $(x, \emptyset)$) associe x. La commutation des diagrammes s'interprète comme :

$$\epsilon x = x \qquad x\epsilon = x$$

$$(xy)z = x(yz)$$

autrement dit comme l'existence d'un élément neutre et l'associativité.
Pour ces raisons, dans une monade, la transformation η est appelée *l'unité* et μ la *multiplication*.

2.1.6 Exemples de monades

Un exemple typique de monade sur la catégorie **Ens** est le foncteur $\wp$. Il consiste à prendre pour unité la famille de flèches $\eta_A : A \to \wp(A)$ définies par : $\forall a \in A, \eta_A(a) = \{a\}$, et pour multiplication, la famille de flèches $\mu_A : \wp(\wp(A)) \to \wp(A)$ définie par :

$$\forall \mathcal{X} \in \wp(\wp(A)), \mu_A(\mathcal{X}) = \bigcup \{X ; X \in \mathcal{X}\}$$

Un autre exemple est fourni par le foncteur *List*, qui, à tout ensemble associe l'ensemble des listes de ses éléments, avec pour unité la famille de fonctions η_A qui sont les fonctions définies sur A associant à chaque élément $a \in A$ l'élément $[a] \in List(A)$, et pour multiplication la famille de fonctions μ_A qui sont les fonctions définies sur $List(List(A))$ associant à chaque élément (par exemple une liste de listes $[[a, b], [c, d], [e]]$) la liste obtenue en mettant à plat les élements de la liste de listes (par exemple $[a, b, c, d, e]$). On voit sur cet exemple que la notion de monade permet de définir d'emblée une opération généralisée, ici la concaténation des listes. La condition d'associativité de la monade assure l'associativité de cette opération.

2.1.7 Rôle des monades

La notion de monade est importante car, si on part d'une catégorie de structures algébriques quelconques **Alg** (par exemple **Mon**, **Grp** etc.) et que l'on définit une monade par l'intermédiaire d'un foncteur de **Ens** dans **Alg** (comme par exemple le foncteur *List* ci-dessus), alors on peut reconstruire la catégorie à partir de cette monade. (Cela revient aussi, étant donnée une monade T sur **Ens**, à se demander

s'il existe une catégorie **Alg** telle que cette monade soit la composée d'adjoints F et G avec F de **Ens** dans **Alg**, et G de **Alg** dans **Ens** (en général le foncteur d'oubli). On peut montrer que c'est le cas mais que cette catégorie n'est pas unique).
Cette reconstruction passe par l'intermédiaire de deux nouveaux types de catégories : les catégories de Eilenberg-Moore et celles de Kleisli.
Nous ne rentrerons pas ici dans le détail de ces constructions. Sachons néanmoins que pour résoudre le problème posé ci-dessus, étant donnée une monade (T, η, μ) sur la catégorie $\mathcal{A}$, on appelle *résolution* de cette monade un quadruplet $(\mathcal{B}, F, G, \epsilon)$ formé d'une catégorie $\mathcal{B}$, de deux foncteurs $F : \mathcal{A} \to \mathcal{B}$ et $G : \mathcal{B} \to \mathcal{A}$ tels que $GF = T$, avec adjonctions η et ϵ, ϵ étant tel que $G\epsilon F = \mu$. Ces résolutions forment elles-mêmes une catégorie, dont les flèches $(\mathcal{B}, F, G, \epsilon) \to (\mathcal{B}', F', G', \epsilon')$ sont des foncteurs $\Phi : \mathcal{B} \to \mathcal{B}'$ tels que $\Phi F = F'$, $G'\Phi = G$ et $\Phi\epsilon = \epsilon'\Phi$. La catégorie de Eilenberg-Moore donnera alors un objet terminal de cette catégorie, alors que la catégorie de Kleisli en donnera un objet initial.

2.1.8 Catégories de Kleisli

Nous présenterons les catégories de Kleisli notamment parce qu'elles ont connu une utilisation intéressante en informatique théorique et en sémantique des langues naturelles.
Soit une monade (T, η, μ) sur la catégorie $\mathcal{A}$. Nous allons modifier cette catégorie de la manière suivante. Chaque flèche $A \to A'$ est remplacée par une flèche $A \to T(A')$. Cela implique que l'on redéfinisse la composition des flèches. Etant donnée deux flèches $f : A \to T(A')$ et $g : A' \to T(A'')$, on notera qu'il est possible de trouver un chemin de A vers $T(A'')$ de la manière suivante :

$$A \xrightarrow{\quad f \quad} T(A') \xrightarrow{Tg} TT(A'') \xrightarrow{\mu_{A''}} T(A'')$$

On définira donc $g \star f$ par :

$$g \star f = \mu_{A''}(Tg)f$$

On remarquera en particulier que :

$$f \star \eta_A = \mu_{A''}(Tf)\eta_A = \mu_{A''}\eta T_{A'} f = 1_T(A')f = f$$

et que :

$$\eta_{A'} \star f = f$$

de sorte que η_A serve de flèche identité.
Nous pouvons reprendre l'exemple de l'ensemble des parties d'un ensemble. Dans la catégorie de Kleisli associée, les flèches $A \to B$ sont en réalité les flèches $A \to \wp(B)$ de **Ens**. Elles correspondent à des applications multivaluées, ou relations entre ensembles. Etant données une flèche $f : A \to \wp(B)$ et une flèche $g : B \to \wp(C)$, on définit leur composée par :

$$g \star f(a) = (\mu_C \wp(g)f)(a) = \bigcup\{g(b); b \in f(a)\}$$

Noter que en associant la relation binaire R_f entre A et B à l'application $f : A \to \wp(B)$, de sorte que aR_fb si et seulement si $b \in f(a)$, on a :

$$\begin{array}{rcl} aR_{g\star f}c & \Leftrightarrow & c \in (g \star f)(a) \\ & \Leftrightarrow & \exists b \in B(b \in f(a) \wedge c \in g(b)) \\ & \Leftrightarrow & \exists b \in B(aR_fb) \wedge (bR_gc)) \\ & \Leftrightarrow & aR_gR_fc \end{array}$$

La catégorie de Kleisli pour la monade $\wp$ sur **Ens** est donc la catégorie dont les objets sont les ensembles et les flèches les relations binaires entre ensembles.

Chung-chieh Shan ([Shan 01]) part directement de la catégorie de Kleisli associée à une monade (au lieu de, simplement, une "monade" comme il le dit, mais il suit en cela [Wadler 92]), qui peut aussi se définir directement par la donnée d'un triplet $(T, \eta, \star)$ avec :

$$\eta : X \to TX \qquad\qquad \star : (TX, X \to T(Y)) \to T(Y)$$

avec les égalités qui conviennent assurant unité et associativité. η sert à injecter les objets de la catégorie de départ (par exemple **Ens**) dans son image par l'endofoncteur T, et $\star$ permet d'appliquer un morphisme de la catégorie transformée à un objet de celle-ci. Cette formulation permet de retrouver la précédente (celle où l'on définissait non pas *l'application* d'un morphisme à un argument, mais la *composition* des morphismes) pour autant que l'on dispose du processus de λ-abstraction, ce qui sous-entend (voir plus loin) que l'on soit dans une Catégorie Cartésienne Fermée. Avec cette réserve, on obtient, comme équivalents des conditions à vérifier (neutralité et associativité) :

$$\begin{array}{cl} \eta_A(a) \star k = k(a) & \forall a : A, k : A \to T(B) \\ m \star \eta_A = m & \forall m : T(A) \\ (m \star k) \star l = m \star \lambda v.(k(v) \star l) & \forall m : T(A), k : A \to T(B), l : B \to T(C) \end{array}$$

[Shan 01] développe l'exemple de la monade $\wp$. Nous savons que pour tout ensemble A, nous définissons $T(A) = \wp(A)$, ou, ce qui revient au même, pour tout type α, le type $T(\alpha) = \alpha \to t$ (où t est le type des valeurs de vérité), alors en ce cas, l'injection η donne : $\eta_\alpha(a) = \{a\} : T(\alpha)$ pour tout $a : \alpha$ et la composition $\star$ est donnée par : $m \star k = \bigcup\{k(a); a \in m\}$.
Considérons un individu de type $T(e)$, dans **Ens**, c'est un ensemble d'individus, dans $T(\mathbf{Ens})$ c'est un nouvel individu... mais un individu composite (ou "non-déterministe") : à l'ensemble $\{Pierre, Marie\}$ correspond l'individu qu'on aurait envie de noter $Pierre \oplus Marie$[4], qui est de type $T(e)$.
On a évidemment la même chose pour tous les types. Par exemple si on considère une entité de type propriété, soit une entité de type $e \to t$, on peut aussi la transporter par le foncteur T, on obtiendra une entité de type $(e \to t) \to t$, qui est un ensemble de propriétés, pouvant aussi être vu comme une "propriété indéterminée", cf. $\lambda x.love(x, Mary) \oplus \lambda x.hate(x, Mary)$.
Comme exemple de composition par $\star$, on a le cas suivant : soit $k : e \to T(e \to t)$, définie par :

$$\forall a : e, k(a) = \{\lambda x.love(x, a), \lambda x.hate(x, a)\} : T(e \to t)$$

soit $m = \{Pierre, Mary\} : T(e)$, alors on a :

$$\begin{array}{rcl} m \star k & = & \bigcup\{\lambda x.love(x, a), \lambda x.hate(x, a); a \in \{Pierre, Mary\}\} \\ & = & \{\lambda x.love(x, Pierre), \lambda x.love(x, Mary), \\ & & \lambda x.hate(x, Pierre), \lambda x.hate(x, Mary)\} : T(e \to t) \end{array}$$

(le non-déterminisme s'est propagé).
Si maintenant, on injecte un individu dans T de manière à garder un individu déterministique, autrement dit si on utilise l'injection η, alors l'individu *Pierre* : e devient $\{Pierre\} : T(e)$ et on ne peut lui appliquer notre fonction précédente k là encore qu'au moyen de $\star$, on obtiendra :

$$\eta(Pierre) \star k = \{\lambda x.love(x, Pierre), \lambda x.hate(x, Pierre)\} : T(e \to t)$$

De la même façon, quand on a une fonction déterministe $h : e \to e \to t$ comme :

$$h = \lambda a.\lambda x.love(a, x)$$

[4]Ceci donne une idée pour le traitement des pluriels.

on peut la transformer en une fonction non-déterministe, de type $e \rightarrow T(e \rightarrow t)$. Pour cela, on utilise $\eta : (e \rightarrow t) \rightarrow T(e \rightarrow t)$, et on fait la composition $\eta \circ h$. On a :

$$(\eta \circ h)(a) = \eta(h(a)) : T(e \rightarrow t)$$

et donc :

$$\eta \circ h = \lambda a.\eta(h(a)) : e \rightarrow T(e \rightarrow t)$$

Noter alors que, si on exprime l'application ordinaire d'une fonction à son argument (comme cela se passe dans la grammaire de Montague) par une fonction *App* (analogue à un *apply*) :

$$App : (A \rightarrow B) \rightarrow (A \rightarrow B)$$

telle que :

$$App(f)(x) = f(x) \qquad \forall f : A \rightarrow B, x : A$$

alors on peut essayer de *lifter* cette fonction pour la transformer en App_T :

$$App_T : T(A \rightarrow B) \rightarrow (T(A) \rightarrow T(B))$$

(par exemple, transformer *App* servant à appliquer une fonction de type $e \rightarrow t$ à une entité de type e en une App_T servant à appliquer une fonction non-déterministe f de type $T(e \rightarrow t)$ à un individu non déterministe x de type $M(e)$). Soit $f : T(A \rightarrow B)$ et $x : T(A)$, alors, pour obtenir un résultat de type $T(B)$, il faut appliquer η à un certain résultat de type B. Supposons $b : A$ et $a : A \rightarrow B$, alors $a(b) : B$ et donc $\eta(a(b)) : T(B)$. Nous obtenons donc $\lambda b.\eta(a(b)) : A \rightarrow T(B)$, c'est typiquement une flèche de la catégorie de Kleisli, dont nous savons comment elle peut s'appliquer à un x de type $T(A)$, nous avons :

$$x \star \lambda b.\eta(a(b)) : T(B)$$

et donc :

$$\lambda a.x \star \lambda b.\eta(a(b)) : (A \rightarrow B) \rightarrow T(B)$$

qu'on peut donc composer avec f :

$$f \star [\lambda a.x \star \lambda b.\eta(a(b))] : T(B)$$

D'où la définition :

$$App_T(f)(x) = f \star [\lambda a.(x \star (\lambda b.\eta(a(b)))]$$

Appliqué au cas particulier du foncteur $\wp$, on en tire :

$$App_\wp(f)(x) = \{a(b); a \in f, b \in x\} : \wp(B)$$

quelle que soit le sous-ensemble f de fonctions de A dans B et le sous-ensemble x de A. Chung-chieh Shan fait ici le lien avec la sémantique des questions de Hamblin, pour laquelle le sens d'une interrogative est donné comme l'ensemble des réponses alternatives possibles à la question. Autrement dit, une question est une fonction de A dans $T(B)$ où A est l'ensemble des propositions (cf. *quelqu'un a ouvert la porte*) et $T(B)$ l'ensemble des ensembles des propositions (cf. *Pierre a ouvert la porte*, *Marie a ouvert la porte*, *Jacques a ouvert la porte* etc.). Une phrase affirmative a alors sa signification qui est donnée par η.

2.2 Systèmes déductifs

2.2.1 Calculs propositionnels

Un système déductif est une catégorie dont les flèches sont interprétées comme des *preuves*, et les objets comme des *formules*. Plus précisément, c'est un graphe avec une flèche particulière 1_A pour chaque objet A (la preuve de A à partir de A) et une opération de composition entre flèches, qu'on peut noter :

$$\frac{f : A \to B \quad g : B \to C}{gf : A \to C}$$

et qu'on peut interpréter comme une *règle d'inférence*.
Un **calcul conjonctif** est alors un système déductif centré sur $\top$ (la "vérité") et $\wedge$ doté des règles suivantes :

1. $\bigcirc_A : A \to \top$
2. $\pi_{A,B} : A \wedge B \to A$
3. $\pi'_{A,B} : A \wedge B \to B$
4. $\dfrac{f : C \to A \quad g : C \to B}{< f, g >: C \to A \wedge B}$

On voit immédiatement que pour avoir un tel calcul, il suffit d'avoir une catégorie qui possède un objet terminal ($\top$) et un produit (ici noté $\wedge$). Une telle catégorie est dite *cartésienne*.
Un **calcul propositionnel intuitionniste positif** sera alors un calcul conjonctif avec un opérateur additionnel *si*, noté $\Leftarrow$, et avec les règles suivantes :

$$\epsilon(B) : (B \Leftarrow C) \wedge C \to B$$

$$\frac{h : A \wedge C \to B}{h^* : A \to B \Leftarrow C}$$

C'est-à-dire les règles d'élimination et d'introduction de l'implication.
Or, là aussi, on peut faire le raisonnement suivant : soit F_C le foncteur qui à toute formule A associe la formule $A \wedge C$. On a vu que l'adjonction revenait à établir une bijection entre toutes les preuves $F_C A \to B$ et toutes les preuves $A \to G_C B$, pour un certain G_C qui est alors l'adjoint à droite de F. Alors, $B \Leftarrow C$ est défini comme $G_C B$. L'implication est l'adjointe à droite de la conjonction.
Autrement dit, le calcul propositionnel intuitionniste positif est un calcul conjonctif où on fait l'hypothèse de l'existence d'une adjonction. C'est ce que l'on appelle une **catégorie cartésienne fermée**. Comme on l'a vu, cela va de pair avec l'hypothèse de l'existence de transformations naturelles $\eta : 1 \to G_C F_C$ et $\epsilon : F_C G_C \to 1$, autrement dit de familles de flèches $\eta(A) : A \to G_C F_C(A)$ et $\epsilon(B) : F_C G_C(B) \to B$, soit, avec nos notations logiques :

- $\eta(A) : A \to (A \wedge C) \Leftarrow C$
- $\epsilon(B) : (B \Leftarrow C) \wedge C \to B$

ϵ est ainsi la règle du *modus ponens*.
On note $\epsilon_{B,C}$ la flèche $\epsilon(B)$ ci-dessus, et $\eta_{A,C}$ la flèche η.
On a vu d'autre part comment associer, dans le cas d'adjonction, à toute flèche $f : F_C A \to B$ une flèche $f^* : A \to G_C B$, c'est-à-dire à toute preuve $f : A \wedge C \to B$, une preuve $f^* : A \to B \Leftarrow C$. Cela donne la règle d'inférence :

$$\frac{h : A \wedge C \to B}{h^* : A \to B \Leftarrow C}$$

Soit une preuve $g : D \to A$. Puisque $\epsilon_{D,B} : (D \Leftarrow B) \wedge B \to D$, on peut faire la composition et on obtient : $g\epsilon_{D,B} : (D \Leftarrow B) \wedge B \to A$, puis on obtient par adjonction : $(g\epsilon_{D,B})^* : (D \Leftarrow B) \to (A \Leftarrow B)$ (en principe, c'est une flèche $GFG(D) \to G(A)$, mais $GFG(D)$ est équivalent à $G(D)$). D'où la règle d'inférence :

$$\frac{g : D \to A}{(g\epsilon_{D,B})^* : (D \Leftarrow B) \to (A \Leftarrow B)}$$

Soit une preuve $g : \top \to B \Leftarrow A$. Elle se compose avec $\bigcirc_A$, de manière à donner une preuve $g\bigcirc_A : A \to B \Leftarrow A$. On peut alors construire le diagramme suivant :

$$\begin{tikzcd}
& A \arrow[dl, "g\bigcirc_A"'] \arrow[d, dashed, "< g\bigcirc_A , 1_A >"] \arrow[dr, "1_A"] & \\
B \Leftarrow A & (B \Leftarrow A) \wedge A \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A
\end{tikzcd}$$

La flèche en pointillés peut être continuée alors par $\epsilon_{B,A}$, et on voit qu'on obtient une preuve de $A \to B$. On pose : $g^\flat = \epsilon_{B,A} < g\bigcirc_A, 1_A >$. D'où la règle d'inférence :

$$\frac{g : \top \to B \Leftarrow A}{g^\flat : A \to B}$$

Similairement, si on part d'une flèche $f : A \to B$, alors elle peut se composer avec la flèche de projection $\pi_2 : \top \wedge A \to A$. On obtient : $f\pi_{2,1,A} : \top \wedge A \to B$, dont on peut prendre l'étoile, ce qui donnera : $(f\pi_{2,1,A})^* : \top \to B \Leftarrow A$. On notera $\lceil f \rceil$ cette flèche. On obtiendra la règle :

$$\frac{f : A \to B}{\lceil f \rceil : \top \to B \Leftarrow A}$$

Un **calcul propositionnel intuitionniste** s'obtient en rajoutant la constante "faux" : $\bot$ et le connecteur $\vee$. $\bot$ est considéré comme objet initial, et le $\vee$ comme le coproduit. Autrement dit, on a les flèches :

1. $\Box_A : \bot \to A$,
2. $\kappa_{A,B} : A \to A \vee B$,
3. $\kappa'_{A,B} : B \to A \vee B$

ainsi que la règle d'inférence :

$$\frac{f : A \to C \quad g : B \to C}{[f, g] : A \vee B \to C}$$

Autrement dit il s'agit d'une catégorie cartésienne fermée, avec, en plus, un coproduit, c'est ce que l'on appelle donc une **catégorie bicartésienne fermée**.
L'importance fondamentale des catégories cartésiennes fermées réside en ce qu'elles nous permettent de retrouver le λ-calcul, en tant que *langage interne* (c'est-à-dire langage apte à décrire les flèches existantes).

2.2.2 Théorème de la déduction

On pourrait se contenter de la règle d'inférence qui fait passer de h à h^* vue plus haut, mais on exige plus. Nous allons en effet nous occuper de preuves faites avec des *hypothèses*. Une hypothèse est une flèche $x : \top \to A$. A partir d'un système déductif $\mathcal{L}$, on forme un nouveau système déductif en ajoutant à $\mathcal{L}$ cette nouvelle flèche. On appelle $\mathcal{L}(x)$ le nouveau système obtenu. L'idée est maintenant de s'intéresser à un tel nouveau système. Les preuves de ce nouveau système dépendent en principe désormais de cette nouvelle flèche, on les écrit donc $\phi(x) : B \to C$.

Théorème 1 (Théorème de la déduction) *Dans un calcul propositionnel, à toute preuve $\phi(x) : B \to C$ obtenue à partir de l'hypothèse $x : \top \to A$ correspond une preuve $f : A \wedge B \to C$ dans $\mathcal{L}$ qui ne dépend pas de x. On note $f = \kappa_{x \in A}\phi(x)$ cette nouvelle preuve.*

Démonstration : la démonstration se fait par récurrence sur la longueur de la preuve. Prenons par exemple le calcul intuitionniste positif. Toute preuve $\phi(x) : B \to C$ à partir de $x : \top \to A$ possède une des formes suivantes :

(i) $k : B \to C$, preuve dans $\mathcal{L}$,

(ii) $x : \top \to A$ avec $B = \top$ et $C = A$,

(iii) $< \psi(x), \chi(x) >$ avec $\psi(x) : B \to C', \chi(x) : B \to C'', C = C' \wedge C''$,

(iv) $\chi(x)\psi(x)$ avec $\psi(x) : B \to D, \chi(x) : D \to C$,

(v) $(\psi(x))^*$ avec $\psi(x) : B \wedge C' \to C'', C = C'' \Rightarrow C'$

On procède en énumérant ces cas.

(i) une flèche $A \wedge B \to C$ s'obtient en composant k avec la deuxième projection du produit, donc $\kappa_{x \in A}k = k\pi'_{A,B}$,

(ii) $\kappa_{x \in A}x = \pi_{A,\top}$,

(iii) $\kappa_{x \in A} < \psi(x), \chi(x) > = < \kappa_{x \in A}\psi(x), \kappa_{x \in A}\chi(x) >$,

(iv) $\kappa_{x \in A}\chi(x)\psi(x) = \kappa_{x \in A}\chi(x) < \pi_{A,B}, \kappa_{x \in A}\psi(x) >$,

(v) $\kappa_{x \in A}(\psi(x))^* = (\kappa_{x \in A}\psi(x)\alpha_{A,B,C'})^*$

(où $\alpha_{A,B,C'}$ est la preuve de l'associativité : $(A \wedge B) \wedge C' \to (A \wedge B) \wedge C'$.
les cas (i) et (ii) sont les cas de base et comme on le constate, ils ne font pas intervenir x.

2.3 λ-calcul dans les catégories cartésiennes fermées

Habituellement, on note A^B ce que nous avons noté $A \Leftarrow B$, dans les CCF. D'après la définition d'une CCF, à partir d'une flèche $h : C \times B \to A$, nous avons une flèche $h^* : C \to A^B$, ce qui nous permet d'obtenir, grâce au produit, une flèche $h^*\pi_{CB} : C \times B \to A^B$, d'où par produit, une flèche $< h^*\pi_{CB}, \pi'_{CB} >: C \times B \to A^B \times B$. Comme nous utilisons le fait que $F_A : B \mapsto A \times B$ et $G_A : B \mapsto B^A$ sont adjointes l'une de l'autre, nous avons la co-unité $\epsilon_{AB} : A^B \times B \to A$, avec la propriété :

$$\epsilon_{AB} < h^*\pi_{CB}, \pi'_{CB} >= h$$

h^* est la *curryfication* de h. Quant à ϵ_{AB}, c'est la flèche de $A^B \times B$ dans B qui correspond à *l'évaluation*. En termes ensemblistes, ce serait la fonction *apply* qui, à toute fonction $f : B \to A$ et à tout élément a de B associe $apply(f, a) = f(a)$. Mais comme nous ne nous référons pas aux ensembles, il faut obtenir l'idée équivalente d'une autre manière. Ce sera en "redécouvrant" (en quelque sorte) le λ-calcul. Avant cela, il nous faut trouver l'équivalent des notions de *fonction* et de *variable*. Pour les variables, nous venons de voir à la section précédente, que nous pouvions faire intervenir dans un système déductif des flèches "indéterminées", qui correspondent à des *hypothèses*. Si 1 représente l'élément neutre (ou ce que nous avons noté $\top$ ci-dessus), nous nous intéressons à des flèches $x : 1 \to A$ et nous engendrons à partir d'une telle flèche toutes les autres qui vont faire que malgré cet ajoût, la structure de catégorie (cartésienne, ou cartésienne fermée) est conservée. Si on partait d'une catégorie $\mathcal{A}$, avec l'ajoût d'une flèche x,

on obtient alors la catégorie $\mathcal{A}[x]$. Les flèches dans $\mathcal{A}[x]$ dépendent de x, on les appelle des *polynômes* en x. La relation "$=_x$", qui sera utilisée ci-dessous est la relation d'équivalence entre morphismes modulo les équations ajoutées entre polynômes pour que le résultat soit bien une catégorie. Comme en algèbre standard, étant donné un polynôme formel, on se demande s'il peut être réalisé par un morphisme (dans le cas standard par une fonction f).
Cela prend la forme d'un théorème de **complétude fonctionnelle**

Théorème 2 *Pour tout polynome $\phi(x) : B \to C$ en $x : 1 \to A$, il existe une flèche unique $f : A \times B \to C$ telle que $f < x\bigcirc_B, 1_B >=_x \phi(x)$*

$$\begin{tikzcd}
& B \arrow[dl, "x\bigcirc_B"'] \arrow[d, "< x\bigcirc_B , 1_B >"] \arrow[dr, "1_B"] & & B \arrow[dd, "\phi(x)"] \\
A & A \times B \arrow[l, "\pi"] \arrow[r, "\pi'"'] \arrow[d, "f"'] & B & \\
& C & & C
\end{tikzcd}$$

Remarque : ceci est comme le théorème de la déduction (se souvenir que $A \times B = A \wedge B$), avec en plus une forme explicite de la décomposition, qui possède le corollaire que, si on prend $B = 1$, alors on obtient : toute preuve $\phi(x) : 1 \to C$ obtenue à partir de l'hypothèse $x : 1 \to A$ est équivalente à une preuve obtenue en composant la flèche $1 \to A$ avec une flèche $A \to C$ (puisque $A \times 1 = A \wedge 1 = A \wedge \top = A$). Autrement dit, c'est l'analogue du théorème d'élimination de la coupure.
Comme dans la démonstration du théorème de la déduction, on note ce morphisme f (unique) : $\kappa_{x \in A}\phi(x)$. On prouve par induction sur la longueur de la preuve que :

$$\kappa_{x \in A}\phi(x) < x\bigcirc_B, 1_B >=_x \phi(x)$$

Corollaire 1 *Pour tout polynome $\phi(x) : 1 \to C$ en une indéterminée $x : 1 \to A$ sur une catégorie cartésienne ou cartésienne fermée $\mathcal{A}$, il existe une unique flèche $g : A \to C$ dans $\mathcal{A}$ telle que $gx =_x \phi(x)$. Sur une catégorie cartésienne fermée $\mathcal{A}$, il existe une unique flèche $h : 1 \to C^A$ telle que $\epsilon_{C,A} < h, x >=_x \phi(x)$.*

Démonstration : dans le premier cas, il suffit de prendre $g = \kappa_{x\in A}\phi(x) < 1_A, \bigcirc_A >$. L'adaptation du diagramme précédent donne en effet :

$$\begin{tikzcd}
& 1 \arrow[d, "x"] & & 1 \arrow[ddd, "\phi(x)"] \\
& A \arrow[dl, "1_A"'] \arrow[d, "< 1_A , \bigcirc_A >"] \arrow[dr, "\bigcirc_A"] & & \\
A & A \times 1 \arrow[l, "\pi"] \arrow[r, "\pi'"'] \arrow[d, "f = \kappa_{x\in A}\phi(x)"] & 1 & \\
& C & & C
\end{tikzcd}$$

Pour le deuxième cas, il suffit d'utiliser la règle d'inférence (p. 24) :

$$\frac{f : A \to C}{\lceil f \rceil : 1 \to C^A}$$

et on obtient bien une flèche $1 \to C^A$, c'est la flèche

$$h = \lceil \kappa_{x\in A}\phi(x) < 1_A, \bigcirc_A > \rceil$$

On peut alors prouver que $\kappa_{x\in A}(gx) < 1_A, \bigcirc_A >= g$ et que $h^\flat = g$. On se souvient pour cela de la règle d'inférence :

$$\frac{g : 1 \to C^A}{g^\flat : A \to C}$$

Finalement, on note : $\lambda_{x\in A}\phi(x)$ la flèche h telle que $h^\flat x = \epsilon_{C,A} < h, x >=_x \phi(x)$

$$\begin{tikzcd}
& 1 \arrow[dl, dashed, "h"'] \arrow[d, "< h , x >"] \arrow[dr, "x"] & & 1 \arrow[dd, "\phi(x)"] \\
C^A & C^A \times A \arrow[l, "\pi"] \arrow[r, "\pi'"'] \arrow[d, "\epsilon_{C,A}"'] & A \arrow[dl, "h^\flat"] & \\
& C & & C
\end{tikzcd}$$

Insistons sur l'importance de ce théorème. Il signifie que toute preuve $\phi(x)$ d'une formule de type C dépendant d'une hypothèse x de type A permet de définir une "fonction" (plus généralement un morphisme) de type C^A. C'est ce morphisme qui permet d'interpréter dénotationnellement le λ-terme $\lambda x.\phi(x)$, et de plus nous avons son unicité. *Conclusion* : la notion de *catégorie cartésienne fermée* permet bien de reconstruire le λ-calcul, ou plus précisément : *un* λ-calcul puisque les objets sont ici définis à un isomorphisme près.

3 Calcul de Lambek revisité à la lumière des catégories

Admettons les axiomes introduits dans la définition d'un système déductif. [Lambek(1988)] définit le *calcul syntaxique* (associatif et avec unité) comme un système déductif où la classe des types (objets) contient un type spécial I (qui sera le type de la chaîne vide) et qui est fermé sous $\bullet, /, \backslash$, et où sont imposées les flèches et règles suivantes.

$$\alpha_{A,B,C} : (A \bullet B) \bullet C \to A \bullet (B \bullet C) \qquad \alpha^{-1}_{A,B,C} : A \bullet (B \bullet C) \to (A \bullet B) \bullet C$$

$$\rho_A : A \bullet I \to A \quad \lambda_A : I \bullet A \to A$$

$$\rho^{-1}_A : A \to A \bullet I \quad \lambda^{-1}_A : A \to I \bullet A$$

$$\frac{f : A \bullet B \to C}{f^* : A \to C/B} \qquad \frac{f : A \bullet B \to C}{{}^*f : B \to A \setminus C}$$

$$\frac{g : A \to C/B}{g^+ : A \bullet B \to C} \qquad \frac{g : B \to A \setminus C}{{}^+g : A \bullet B \to C}$$

On doit considérer ici que f^* (resp. ${}^*f$, g^+, ${}^+g$) est l'image par un certain foncteur de f (resp. f, g, g). Il serait rigoureux d'écrire, en notant $\beta_{A,B,C}, \gamma_{A,B,C}, \beta^{-1}_{A,B,C}, \gamma^{-1}_{A,B,C}$ ces foncteurs :

$$f^* = \beta_{A,B,C}(f) \quad {}^*f = \gamma_{A,B,C}(f) \quad g^+ = \beta^{-1}_{A,B,C}(g) \quad {}^+g = \gamma^{-1}_{A,B,C}(g)$$

Il est facile de voir qu'une dérivation dans **L** est désormais représentée par une flèche (morphisme) qui s'exprime à l'aide de celles qui sont ainsi données. Ainsi :

$$\cfrac{\cfrac{1_{A/B} : A/B \to A/B}{1^+_{A/B} : (A/B) \bullet B \to A}}{{}^*(1^+_{A/B}) : B \to (A/B) \setminus A}$$

donne une preuve du théorème de type-raising. Noter que $B \to (A/B) \setminus A$ est le résultat et que ${}^*(A^+_{A/B})$ est le nom de la flèche qui encode la dérivation.
Exercices :
- Donner les preuves de :

$$\begin{array}{l} B \bullet (B \setminus A) \to A \\ I \to A/A \\ (A \setminus B)/C \to A \setminus (B/C) \\ (A/B) \bullet (B/C) \to A/C \\ A/B \to (A/C)/(B/C) \end{array}$$

- Démontrer que l'ensemble des sous-ensembles d'un monoïde est un calcul syntaxique, où $I, \bullet, /$ et $\backslash$ sont définis de la façon suivante :

$$A \bullet B = \{x^\frown y \in M; x \in A \wedge y \in B\} \tag{1}$$
$$C/B = \{x \in M; \forall y \in B\ x^\frown y \in C\} \tag{2}$$
$$A \setminus C = \{y \in M; \forall x \in A\ x \bullet y \in C\} \tag{3}$$
$$I = \{\epsilon\} \tag{4}$$

et en supposant qu'il existe une seule flèche de A vers B quand $A \subset B$ (inclusion).
Remarque : on peut démontrer (théorème de complétude de Buszkowski) que :

Théorème 3 *$A \to B$ est un théorème du calcul de Lambek si et seulement si $V(A) \subset V(B)$ pour toute valuation V, c'est-à-dire toute assignation qui assigne aux types de base des sous-ensembles arbitraires d'un monoïde M et aux autres types les sous-ensembles définis par les égalités ci-dessus.*

On appelle *catégorie monoïdale bi-fermée* : une catégorie $\mathcal{C}$, qui satisfait aux équations d'une catégorie, et qui est telle que :

1. $\bullet, /, \backslash$ sont des bi-foncteurs, respectivement de $\mathcal{C} \times \mathcal{C}$ dans $\mathcal{C}$, de $\mathcal{C} \times \mathcal{C}^{op}$ dans $\mathcal{C}$ et de $\mathcal{C}^{op} \times \mathcal{C}$ dans $\mathcal{C}$;
2. $\rho, \lambda, \alpha, \beta, \gamma$ sont des isomorphismes naturels ;
3. les flèches composées suivantes sont des identités :

$$\begin{array}{lll} A \bullet B \to (A \bullet I) \bullet B & \to & A \bullet (I \bullet B) \to A \bullet B \\ ((A \bullet B) \bullet C) \bullet D & \to & (A \bullet B) \bullet (C \bullet D) \\ & \to & A \bullet (B \bullet (C \bullet D)) \\ & \to & A \bullet ((B \bullet C) \bullet D) \\ & \to & ((A \bullet B) \bullet C) \bullet D \end{array}$$

4 Calcul de Lambek symétrique

4.1 Symétrisation

Dans ce qui suit, nous notons $\otimes$ le produit jusqu'ici noté $\bullet$, afin d'établir une homogénéité de notation avec l'opérateur qui sera nouvellement introduit $\oplus$. Nous avons fait référence plus haut aux valuations V qui permettent de donner une sémantique dénotationnelle aux types syntaxiques (en termes de parties d'un monoïde). La formule qui définit $V(A \otimes B)$ a une "duale". Ce serait la formule :

$$\forall a \forall b (c = ab \Rightarrow a \in A \vee b \in B)$$

Notons donc $A \oplus B$ la formule qui, en termes monoïdaux, aurait cette sémantique.
Il faut aussi un symétrique de **1**. On peut considérer le complémentaire de **1**. Notons-le **0**. Nous pouvons noter que :

$$\begin{aligned} A \oplus \mathbf{0} &= \{c; \forall a \forall b (c = ab \Rightarrow a \in A \vee b \in \mathbf{0})\} \\ &= \{c; \forall a \forall b (c = ab \Rightarrow a \in A \vee b \neq 1)\} \\ &= \{c; \forall a \forall b (c = ab \wedge b = 1 \Rightarrow a \in A)\} \\ &= \{c; \forall a \forall b (c = a \Rightarrow a \in A)\} = A \end{aligned}$$

On peut aussi prouver que :

$$(A \oplus B) \oplus C = A \oplus (B \oplus C)$$

Il reste à trouver des adjoints pour cette opération. Ce seront des opérations $\mathbin{\backslash\!\backslash}$ et $\mathbin{/\!/}$ telles que :

$$C \subset A \oplus B \Leftrightarrow C \mathbin{/\!/} B \subset A \Leftrightarrow A \mathbin{\backslash\!\backslash} C \subset B$$

Il suffit pour cela de définir :

$$C \mathbin{/\!/} B = \{x; \exists b (xb \in C \wedge b \notin B\}$$

$$A \mathbin{\backslash\!\backslash} C = \{x; \exists a (ax \in C \wedge a \notin A\}$$

Il est facile de voir que, si nous définissons une négation par :

$$\neg A = \{b; b \notin A\}$$

on a :

$$A \oplus B = \neg(\neg A \otimes \neg B) \; et \; \mathbf{0} = \neg \mathbf{1}$$

de plus,

$$C /\!\!/ B = \neg(\neg C / \neg B)$$

4.2 Le calcul symétrique

Revenant au calcul syntaxique, nous avons maintenant enrichi ce dernier en lui adjoignant un connecteur 0-aire et trois connecteurs binaires :

$$\mathbf{0}, \oplus, /\!\!/, \backslash\!\!\backslash$$

avec les règles et axiomes :

$$(A \oplus B) \oplus C \leftrightarrow A \oplus (B \oplus C) \qquad\qquad A \oplus \mathbf{0} \leftrightarrow A \leftrightarrow \mathbf{0} \oplus A$$

$$C \rightarrow A \oplus B \; ssi \; C /\!\!/ B \rightarrow A \; ssi \; A \backslash\!\!\backslash C \rightarrow B$$

La logique obtenue est une forme dite faible de la logique bilinéaire.

4.3 Les lois de Grishin

Les sous-ensembles d'un monoïde satisfont de plus des propriétés que la forme faible de la logique bilinéaire ne permet pas de démontrer. C'est la raison pour laquelle V. N. Grishin a suggéré d'introduire des axiomes supplémentaires, en l'occurrence :

$$(A \oplus B)/C \rightarrow A \oplus (B/C) \qquad\qquad C \backslash (B \oplus A) \rightarrow (C \backslash B) \oplus A$$

Exemple : soit $x \in (A \oplus B)/C$ et $c \in C$, alors $xc \in A \oplus B$ donc si $xc = ab$, on a : $a \in A$ ou bien $b \in B$. Supposons que $x = ay$ et que $a \notin A$, alors $xc = ab = (ay)c = a(yc) \in A \oplus B$. Comme $a \notin A$, il résulte que $yc \in B$. Donc quelque soit $c \in C$, $yc \in B$, on en déduit que $y \in B/C$. donc si $x = ay$ et $a \notin A$, on a $y \in B/C$, c'est donc que $x \in A \oplus (B/C)$.
On en déduit aussi :

$$(A \backslash\!\!\backslash D) \otimes C \rightarrow A \backslash\!\!\backslash (D \otimes C) \qquad\qquad C \otimes (D /\!\!/ A) \rightarrow (C \otimes D) /\!\!/ A$$

Ces axiomes permettent de faire communiquer entre eux les opérateurs des deux familles (la famille conjonctive et la famille disjonctive), ce qui s'avère crucial du point de vue linguistique (voir **exercices**).

4.4 Calcul des séquents

Les séquents pour la logique bilinéaire faible sont de la forme $\Gamma \vdash \Delta$, avec $\Gamma = A_1, ..., A_m$ et $\Delta = B_1, B_2, ..., B_n$. Le sens d'un séquent de ce genre est :

$$A_1 \otimes ... \otimes A_m \rightarrow B_1 \oplus ... \oplus B_n$$

Règles structurelles :

$$A \vdash A$$

$$\frac{\Lambda \vdash A \quad \Gamma, A, \Delta \vdash \Theta}{\Gamma, \Lambda, \Delta \vdash \Theta} \qquad\qquad \frac{\Lambda \vdash \Phi, A, \Psi \quad A \vdash \Theta}{\Lambda \vdash \Phi, \Theta, \Psi}$$

Règles logiques :

$$\frac{\Gamma, \Delta \vdash \Theta}{\Gamma, \mathbf{1}, \Delta \vdash \Theta} \qquad \vdash \mathbf{1}$$

$$\frac{\Gamma, A, B, \Delta \vdash \Theta}{\Gamma, A \otimes B, \Delta \vdash \Theta} \qquad A, B \vdash A \otimes B$$

$$B/A \quad A \vdash B \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash B/A}$$

$$\Lambda \vdash \Phi, \top, \Psi$$

$$A \wedge B \vdash A \qquad \frac{\Lambda \vdash \Phi, A, \Psi \quad \Lambda \vdash \Phi, B, \Psi}{\Lambda \vdash \Phi, A \wedge B, \Psi}$$

$+$ toutes les règles qu'on peut obtenir par inversion des flèches et remplacement de $\otimes$ par $\oplus$.
On peut intégrer les deux règles de coupure dans les règles logiques de manière à reformuler $A, B \vdash A \otimes B$ sous la forme :

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$$

dans le but de prouver un théorème d'élimination des coupures. On notera en effet que cette règle (règle classique d'introduction à gauche du $\otimes$) résulte de la preuve suivante :

$$\frac{\Delta \vdash B \qquad \dfrac{\Gamma \vdash A \qquad A, B \vdash A \otimes B}{\Gamma, B \vdash A \otimes B}}{\Gamma, \Delta \vdash A \otimes B}$$

Idem pour obtenir les règles connues concernant $/$ et $\backslash$:

$$\frac{\dfrac{\Theta \vdash A \qquad B/A, A \vdash B}{B/A, \Theta \vdash B} \qquad \Gamma, B, \Gamma' \vdash \Delta}{\Gamma, B/A, \Theta, \Gamma' \vdash \Delta}$$

Le "$\wedge$" (resp. "$\vee$") correspond au & (resp. $\oplus$) de la logique linéaire. Par symétries, on obtient les règles du $\vee$, par exemple, à partir de celle du $\wedge$ à droite, par symétrisation des séquents :

$$\frac{\Gamma, A, \Gamma' \vdash \Delta \qquad \Gamma, B, \Gamma' \vdash \Delta}{\Gamma, A \vee B, \Gamma' \vdash \Delta}$$

mais aussi, par inversion des flèches, à partir de la règle d'introduction à gauche de $/$:

$$\frac{A \vdash \Theta \qquad \Delta \vdash \Gamma, B, \Gamma'}{\Delta \vdash \Gamma, B /\!\!/ A, \Theta, \Gamma'}$$

4.5 Associativité mixte et système $BL2$

On peut rajouter deux autres règles de coupure :

$$\frac{\Lambda \vdash \Phi, A \quad A, \Delta \vdash \Theta}{\Lambda, \Delta \vdash \Phi, \Theta} \qquad \frac{\Lambda \vdash A, \Psi \quad \Gamma, A \vdash \Theta}{\Gamma, \Lambda \vdash \Theta, \Psi}$$

mais alors ces règles correspondent aux *lois d'associativité mixte* :

$$(A \oplus B) \otimes C \to A \oplus (B \otimes C) \qquad A \otimes (B \oplus C) \to (A \otimes B) \oplus C$$

(ce qu'on voit facilement en notant que, si on prend la première, Λ, Δ donne $(\Phi, A), \Delta$ et si on admet l'associativité mixte : $\Phi, (A, \Delta)$ donc Φ, Θ.)
Ces règles permettent alors d'obtenir les converses des lois de Grishin :

$$A \oplus (D/C) \to (A \oplus D)/C \qquad C\backslash(D \oplus A) \to (C\backslash D) \oplus A$$

Lambek ([Lambek(1993)]) rappelle que Szabo a développé un calcul des séquents pour la logique avec règles structurelles restreintes en utilisant une règle de coupure plus générale :

$$\frac{\Lambda \vdash \Phi, A, \Psi \quad \Gamma, A, \Delta \vdash \Theta}{\Gamma, \Lambda, \Delta \vdash \Phi, \Theta, \Psi}$$

mais on remarque l'étrangeté de cette règle, qui nécessite l'échange entre Φ et Γ d'un côté et Ψ et Δ de l'autre. Cette règle n'est pas valide en logique bilinéaire, où les règles de coupure doivent être restreintes aux cas où soit Φ ou Γ est vide, soit Ψ ou Δ est vide.
On décide dorénavant de noter $BL1$ la logique bilinéaire faible sans les lois de Grishin, $BL1(b)$ la logique bilinéaire avec les converses des lois de Grishin et $BL2$ la logique bilinéaire faible avec les lois de Grishin et leurs converses.
Dans $BL2$ avec les quatre règles de coupure, on peut formuler les règles logiques de la manière suivante. Sous cette forme, elles absorbent les coupures (autrement dit, on peut facilement procéder à l'élimination des coupures).

$$\frac{\Gamma, \Delta \vdash \Theta}{\Gamma, \mathbf{1}, \Delta \vdash \Theta} \qquad \vdash \mathbf{1}$$

$$\frac{\Gamma, A, B, \Delta \vdash \Theta}{\Gamma, A \otimes B, \Delta \vdash \Theta} \qquad \frac{\Gamma \vdash \Phi, A, \Psi \quad \Delta \vdash B, \Psi}{\Gamma, \Delta \vdash \Phi, A \otimes B, \Psi}$$

$$\frac{\Lambda \vdash A, \Psi \quad \Gamma, B, \Delta \vdash \Theta}{\Gamma, B/A, \Lambda, \Delta \vdash \Theta, \Psi} \Delta \; ou \; \Psi \; vide \qquad \frac{\Gamma, A \vdash \Phi, B}{\Gamma \vdash \Phi, B/A}$$

$$\frac{\Gamma, A, \Delta \vdash \Theta}{\Gamma, A \wedge B, \Delta \vdash \Theta} \qquad \frac{\Lambda \vdash \Phi, A, \Psi \quad \Lambda \vdash \Phi, B, \Psi}{\Lambda \vdash \Phi, A \wedge B, \Psi}$$

+ règles obtenues à partir des symétries et de l'inversion de flèches.
Exercice : Démontrer que pour la logique bilinéaire ici présentée $\perp$ est un *objet dualisant*, qu'autrement dit, on a :

$$\perp/(A \setminus \perp) = A = (\perp/A) \setminus \perp$$

4.6 Une dérivation utilisant le calcul symétrique

[Kurtonina & Moortgat (2007)] donne une dérivation de *Alice thinks someone left*, avec lecture *de re* dans le calcul symétrique. “Alors que l’assignement de type $s/(sn \backslash s)$ restreint l’analyse à la construction locale de la portée de *someone*, c’est-à-dire limitée à la clause enchâssée, l’assignation du type $(s /\!\!/ s) \backslash\!\!\backslash sn$ admet aussi la construction non locale c’est-à-dire avec portée de *someone* sur la phrase principale”. On obtient la dérivation suivante :

$$
\cfrac{
  \cfrac{
    \cfrac{}{sn \vdash sn}
    \quad
    \cfrac{
      \cfrac{}{s \vdash s}
      \quad
      \cfrac{\cfrac{}{sn \vdash sn} \quad \cfrac{}{s \vdash s}}{(sn \otimes (sn \backslash s) \vdash s}
    }{(sn \otimes (((sn \backslash s)/s) \otimes s)) \vdash s}
  }{(sn \otimes (((sn \backslash s)/s) \otimes (sn \otimes (sn \backslash s))) \vdash s}
  \quad
  \cfrac{
    \cfrac{\cfrac{}{s \vdash s} \quad \cfrac{}{s \vdash s}}{s \vdash (s /\!\!/ s), s}\,[/\!\!/ D]
  }{s \vdash (s /\!\!/ s) \oplus s}\,[\oplus D]
}{
  \cfrac{(sn \otimes (((sn \backslash s)/s) \otimes (sn \otimes (sn \backslash s))) \vdash (s /\!\!/ s) \oplus s}{
    \cfrac{(\mathbf{s} /\!\!/ \mathbf{s}) \backslash\!\!\backslash (sn \otimes (((sn \backslash s)/s) \otimes (sn \otimes (sn \backslash s)))) \vdash s}{
      \cfrac{sn \otimes ((\mathbf{s} /\!\!/ \mathbf{s}) \backslash\!\!\backslash (((sn \backslash s)/s) \otimes (sn \otimes (sn \backslash s)))) \vdash s}{
        \cfrac{sn \otimes (((sn \backslash s)/s) \otimes ((\mathbf{s} /\!\!/ \mathbf{s}) \backslash\!\!\backslash (sn \otimes (sn \backslash s)))) \vdash s}{
          sn \otimes (((sn \backslash s)/s) \otimes (((s /\!\!/ s) \backslash\!\!\backslash sn) \otimes (sn \backslash s))) \vdash s
        }\,G1
      }\,G2
    }\,G2
  }\,[\backslash\!\!\backslash G]
}\,cut
$$

où G1 et G2 sont les lois de Grishin suivantes :

$$
\begin{array}{ll}
(G1) & (A \backslash\!\!\backslash B) \otimes C \vdash A \backslash\!\!\backslash (B \otimes C) \\
(G2) & C \otimes (A \backslash\!\!\backslash B) \vdash A \backslash\!\!\backslash (C \otimes B)
\end{array}
$$

Evidemment, une telle dérivation demande une extension de l’isomorphisme de Curry-Howard, afin que chaque pas dans la dérivation puisse se traduire en une opération de construction de la forme sémantique. Cela sera fait grâce à la sémantique des continuations, et plus particulièrement par l’introduction du $\lambda\mu\tilde{\mu}$-calcul, défini par Curien et Herbelin.

Exercice : Démontrer que $(s /\!\!/ s) \backslash\!\!\backslash sn \vdash s/(sn \backslash s)$ et que $(n \backslash n)/((s /\!\!/ s) \oplus (s/sn)) \vdash (n \backslash n)/(s/sn)$. Montrer que ce deuxième type peut être attribué au pronom *which* de manière à obtenir une dérivation de *movie which John saw on TV*, résolvant ainsi le problème des extractions non périphériques (dont le calcul de Lambek pur ne peut rendre compte).

5 Logique linéaire non commutative

Ce que Lambek appelle “logique bilinéaire” est, au premier regard, la logique linéaire (de Girard) avec la règle structurelle de permutation en moins : on obtient alors la logique linéaire non-commutative, qui a été étudiée par M. Abrusci et P. Ruet. A ceci près que, en logique linéaire non commutative, $\oplus$ est remplacé par $\wp$, lequel ne se trouve pas défini comme co-produit et n’a pas la sémantique dénotationnelle vue plus haut : les opérateurs de la logique linéaire ont une interprétation dans les *espaces cohérents*, ce qui n’est pas du tout la même chose. Attention : $\wp$ est non commutatif. Cela entraîne que, logiquement, on a deux négations, l’une à droite, l’autre à gauche, de manière à ce qu’on puisse définir / et $\backslash$ par :

$$A \backslash B = {}^{\perp}A \wp B \qquad\qquad B/A = B \wp A^{\perp}$$

On en déduit :

$$A^{\perp} = \perp/A \qquad\qquad {}^{\perp}A = A \backslash \perp$$

et :
$$\vdash A^{\perp}, A \qquad\qquad \vdash A, {}^{\perp}A$$
La sorte d'associativité "mixte", introduite par les axiomes de Grishin, est en fait la règle en logique linéaire, puisqu'on peut démontrer, par exemple, le séquent :

$$(A \wp B) \otimes C \vdash A \wp (B \otimes C)$$

6 Prégroupes

[Lambek(1998)] change quelque peu les notations, il note 0 ce que nous avons noté $\perp$ (ce qui n'est pas trop grave puisque nous avons délaissé les additifs) et il note a^r (resp. a^l) ce que nous avons noté ${}^{\perp}a$ (resp. $a^{\perp}$). On vérifiera que l'on a alors :

$$a^{rl} = a = a^{lr}$$

comme conséquence de la propriété d'objet dualisant.
En reprenant le langage des catégories et en considérant la catégorie des ensembles partiellement ordonnés où la flèche n'est autre que la relation d'ordre ($\leq$), on a :

$$a \otimes a^r \leq 0, \quad a^l \otimes a \leq 0$$

et de manière duale :
$$1 \leq a^r \wp a \quad 1 \leq a \wp a^l$$
(que, par soucis de simplicité, Lambek note : $1 \leq a^r + a$ et $1 \leq a + a^l$). On a alors, en définissant a/b comme $a + b^l$ et $b \setminus a$ comme $b^r + a$:

$$(a/b)b = (a + b^l)b \leq a + (b^l b) \leq a + 0 = a$$

grâce aux axiomes de Grishin.
[Lambek(1998)] considère alors que "trop d'opérations apparaissent dans un tel calcul, par rapport à ce qu'exige la linguistique" (!) et que sans doute il serait plus "raisonnable" de procéder aux identifications :

$$a + b = a \otimes b \qquad\qquad 0 = 1$$

Cette hérésie sur le plan de la logique conduit néanmoins à une structure algébrique connue, celle de *prégroupe*.

Définition 14 *Un* **prégroupe** *est un monoïde muni d'une relation d'ordre partielle, dans lequel chaque élément a possède deux adjoints (un adjoint gauche, a^l, et un adjoint droite, a^r) de sorte que : $a^l a \leq 1 \leq aa^l$ et $aa^r \leq 1 \leq a^r a$.*

Les inégalités fondamentales d'un prégroupe s'interprètent alors comme des règles de *réduction* ($a^l a \to 1$ et $aa^r \to 1$) et des règles d'*expansion* ($1 \to aa^l$ et $1 \to a^r a$). Plusieurs applications des prégroupes ont été faites, concernant diverses langues.
Dans la hiérarchie des catégories, les prégroupes correspondent aux catégories bi-fermées compactes. En oubliant l'ordre, on obtient une catégorie monoïdale, qui est aussi une catégorie que l'on trouve à partir des espaces vectoriels munis du produit tensoriel. Cela ouvre alors le champ à la *sémantique distributionnelle*.
Note sur prégroupes et grammaires catégorielles : à première vue, les prégroupes permettent de retrouver les règles du calcul de Lambek. Les réductions donneraient les applications : ainsi $A/B\ B \to A$ se

traduirait en $ab^l\ b \rightarrow a(b^l b) \rightarrow a1 \rightarrow a$, et les expansions les élévations de type (règles *hypothétiques*) : on aurait : $a \rightarrow a1 \rightarrow a(b^l b)$. On peut ainsi facilement voir que les réductions de *livre que Marie lit* sont semblables dans le calcul de Lambek et dans une grammaire de prégoupes :

$$\begin{array}{rccccc} \textit{Lambek}: & n & (n \setminus n)/(s/sn) & sn & (sn \setminus s)/sn \\ \textit{prégroupes}: & n & n^r n\ sn^{ll} s^l & sn & sn^r\ s\ sn^l \end{array}$$

mais il n'est pas possible dans les grammaires de prégroupes de faire la différence entre $(A \otimes B)/C$ et $A \otimes (B/C)$ qui, tous les deux, se traduisent par la même expression : ABC^l.

7 Catégories monoïdales

La notion de prégroupe fait surgir un nouveau type de catégorie : les catégories dites *monoïdales*. Elles ont ceci de particulier que, pour le dire à la façon d'un McLane ([MacLane 1971]), elles viennent "tout équipées d'un produit", que nous noterons $\otimes$.

Définition 15 *Une catégorie monoïdale* $< \mathcal{B}, \otimes, e >$ *est une catégorie* $\mathcal{B}$ *avec un bifoncteur* $\otimes : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ *associatif et avec un élément unité* e*.*

La notion de bifoncteur provient de celle de produit de deux catégories. etant données deux catégories $\mathcal{A}$ et $\mathcal{B}$, on peut définir une catégorie $\mathcal{A} \times \mathcal{B}$ de la manière suivante :

- un objet de $\mathcal{A} \times \mathcal{B}$ est un couple (A, B) formé d'un objet de $\mathcal{A}$ et d'un objet de $\mathcal{B}$
- une flèche $(A, B) \rightarrow (A', B')$ de $\mathcal{A} \times \mathcal{B}$ est un couple de flèches $(A \rightarrow A', B \rightarrow B')$
- la composée de deux flèches est définie par :

$$(f', g') \circ (f, g) = (f' \circ f, g' \circ g)$$

Des foncteurs $P : \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{A}$ et $Q : \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{B}$ sont définis par :

$$P(f, g) = f \qquad\qquad Q(f, g) = g$$

on les appelle les *projections*. Deux foncteurs $U : \mathcal{A} \rightarrow \mathcal{A}'$ et $V : \mathcal{B} \rightarrow \mathcal{B}'$ ont aussi un produit $U \times V : \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{A}' \rightarrow \mathcal{B}'$, défini par :

$$(U \times V)(A, B) = (UA, VB) \qquad\qquad (U \times V)(f, g) = (Uf, Vg)$$

Un foncteur d'une catégorie produit $\mathcal{A} \times \mathcal{B}$ dans une catégorie $\mathcal{C}$ est alors appelé un bi-foncteur. Le produit cartésien de deux catégories est, par exemple, un bifoncteur.
Dire qu'un bifoncteur $\otimes$ de $\mathcal{B} \times \mathcal{B}$ dans $\mathcal{B}$ est associatif se traduit par le fait que pour tous objets A, B, C de $\mathcal{B}$, on a :

$$(A \otimes B) \otimes C = A \otimes (B \otimes C)$$

ce qui revient à[5] :

$$\otimes(\otimes \times Id) = \otimes(Id \times \otimes)$$

$\obslash\oslash$

[5]Noter que $\otimes \times Id$ est le foncteur de $(\mathcal{B} \times \mathcal{B}) \times \mathcal{B}$ dans $\mathcal{B} \times \mathcal{B}$ qui envoie tout $((A, B), C)$ sur $(A \otimes B, C)$, qui donne ensuite par composition avec $\otimes$: $(A \otimes B) \otimes C$, alors que $Id \times \otimes$ envoie $(A, (B, C))$ sur $(A, B \otimes C)$ qui donne, par composition avec $\otimes : A \otimes (B \otimes C)$.

Références

[Kurtonina & Moortgat (2007)] N. Kurtonina & M. Moortgat, Relational Semantics for the Lambek-Grishin Calculus, *Proceedings of MOL'07*,

[Lambek(1988)] Lambek, J. (1988). *Categorial and categorical grammars* in E. Bach, R. Oehrle and D.Wheeler (eds.), *Categorial Grammars and Natural Language Structures* (D. Reidel), pp. 297–317.

[Lambek(1993)] Lambek, J. (1993). *From Categorial Grammar to Bilinear Logic* in P. Schroeder-Heister at Kosta Dosen (eds.), *Substructural Logics*, Oxford Science Publications, pp. 207–238.

[Lambek & Scott] Lambek, J. & Scott, P.J. (1986) *Introduction to higher order categorical logic*, Cambridge Studies in Advanced Mathematics, Cambridge University Press,

[Lambek(1998)] Lambek, J. (1998). *Type Grammar Revisited* in A. Lecomte, F. Lamarche, G. Perrier (eds.), *Logical Aspects of Computational Linguistics* LNCS-LNAI n^0 1582, Springer, 1998, pp. 297–317.

[MacLane 1971] Mc Lane, S. (1971) *Categories for the working mathematician*, Graduate Texts in Mathematics 5, Springer-verlag.

[Shan 01] Shan, Chung-chieh. (2001) Monads for Natural Language Semantics, *Proceedings of the ESSLLI-2001 Student Session, ed. K. Striegnitz, 285–298, 13th European Summer School in Logic, Language and Information*

[Wadler 92] Wadler, P. (1992) The essence of functional programming, in *POPL'92 : Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, pp 1–14, ACM Press