DIAGRAMMES

PIERRE AGERON

Sémantique catégorique des constructeurs de types d'ordre supérieur

Diagrammes, tome 22 (1989), p. 31-44

<http://www.numdam.org/item?id=DIA_1989__22__31_0>

© Université Paris 7, UER math., 1989, tous droits réservés.

L'accès aux archives de la revue « Diagrammes » implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/legal.php). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

DIAGRAMMES VOLUME 22 , 1989

ACTES DES JOURNEES E.L.I.T.

(UNIV. PARIS 7, 27 JUIN-2 JUILLET 1988)

SEMANTIQUE CATEGORIQUE DES CONSTRUCTEURS DE TYPES D'ORDRE SUPERIEUR

Pierre AGERON

ABSTRACT. We describe a new categorical framework for the semantics of polymorphic type theory. Its two main features are:
- variable types are mere maps rather than functors
- the type $\forall X.A(X)$ is the solution of a small inductive universal problem instead that of a big projective one.

For this purpose, we introduce abstract "type syntaxes", in the spirit of Ehresmann's sketches, and study their semantics. We get a quite general result of algebricity (tripleability).

1 - QU'EST-CE QU'UN TYPE DANS UNE CATEGORIE ?

Pourquoi choisir le langage de la théorie des catégories pour parler de types? Un argument de départ peut être le suivant. La notion de type est intuitivement peu différente de celle d'ensemble; or chaque catégorie est une sorte de variante de la théorie des ensembles: il est alors normal de chercher à faire vivre les types dans certaines catégories convenables. La catégorie Ens n'est alors plus qu' une catégorie parmi les autres, sémantiquement adaptée ou non au type ou système de types considéré.

reçu le 04/09/1989

Voici les questions de fond que nous allons discuter:

a) Qu'est-ce qu'un type A dans une catégorie $\mathbb{C}$?
b) Qu'est-ce qu'un terme fonctionnel de type A=>B ?

L'idée la plus simple qui vienne à l'esprit, c'est de considérer que les types sont des *objets* de $\mathbb{C}$, un terme de type A=>B étant alors une *flèche* de A vers B.
Ce point de vue impose de se donner arbitrairement un certain nombre de types de base, à partir desquels les autres types sont construits par produits, exponentielles, sommes;... On retrouve ici les limitations d'expressivité du lambda-calcul typé ordinaire de Curry-Church, dans lequel il n'est pas possible de manipuler les types de données courants (booléen, entier, liste, etc.)- à moins de les considérer comme des types atomiques et d'introduire pour chacun d'eux de nouveaux schémas non-logiques.

C'est précisément l'intérêt du système F, ou lambda-calcul typé du deuxième ordre de Girard ([1]), que d'échapper à l'atomicité, tout en restant purement logique (c'est la version fonctionnelle du calcul propositionnel intuitionniste du deuxième ordre).

Dans ce système, comme dans d'autres voisins, il n'y a plus de types de base: tous les types y sont construits, conformément à leur "géométrie" intrinsèque.
Ceci est rendu possible par la considération de types variables, c'est-à-dire de types où figurent librement des variables de type : X,Y,... Pour alléger, les types que nous écrirons comporteront au plus une variable libre. Si A(X) est un tel type, on peut construire le type $\forall X.A(X)$, dans lequel X est muette. L'étude sémantique devra statuer sur le point de savoir

c) Quelle est la construction qui fait passer d'un type variable A(X) au type clos $\forall X.A(X)$?

Dans un premier temps, il est extrêmement tentant d'admettre que les types variables A(X) sont certains *foncteurs* $A : \mathbb{C} \to \mathbb{C}$, qu'un terme fonctionnel de type A(X)=>B(X) est une *transformation naturelle* de A vers B, et que le type clos $\forall X.A(X)$ est la *limite projective* $\underset{X\in\mathbb{C}}{\text{Limproj}}\ A(X)$ (certains ont pensé aussi, mais ceci semble moins naturel, au *produit* $\prod_{X\in\mathbb{C}} A(X)$).

Or cette belle simplicité ne tient pas la route. D'abord, imposer la covariance de tous les types variables est excessif: il faut pouvoir séparer occurrences négatives et positives d'une même variable par rapport au constructeur =>. Quant aux "gros" produits, leur existence est en général fort problématique.
Certains auteurs ont cependant pu établir une sémantique sur ces bases "fonctorielles", au prix d'astuces peu convaincantes: travailler avec des catégories internes (Seely, Pitts), supposer que tous les morphismes sont des monomorphismes, ou jouer avec deux sortes de morphismes (Girard, Coquand, Lamarche, Taylor).

Dans un deuxième temps est venue (Freyd, Girard, Scott, Scedrov) l'idée de dédoubler chaque variable de type X en X^g et X^d, donc d'opter pour une représentation "schizophrène" du type variable A(X) par un *bifoncteur* $A : \mathbb{C}^{op} \times \mathbb{C} \to \mathbb{C}$. Les termes fonctionnels devenaient alors des *transformations dinaturelles*, et le type clos $\forall X.A(X)$ était la *fin projective* $\int A(X,X)$ du bifoncteur A.
Mais on se heurta rapidement à une difficulté grave: contrairement aux transformations naturelles, les transformations dinaturelles ne se composent pas. Il devenait en général impossible de substituer un terme dans un autre. Ajoutons aussi qu'un tel formalisme ruine tout espoir d'algébricité, de présentation équationnelle, etc.: quoi qu'on y fasse, $\mathbb{C}^{op}$ n'est pas "$\mathbb{C}$ à la puissance op"!

Très récemment, Freyd, constatant que les notions classiques se révélaient inadéquates, a été amené, pour interpréter le système F, à définir de nouvelles notions de "théorie générale des catégories": celles de *structeur*, de *transformation structurelle*, et de *liaison projective* d'un structeur. Ses premiers résultats sont très intéressants ([2]).

Un point commun à tous ces formalismes est de définir le type $\forall X.A(X)$ comme la solution d'un "gros" problème universel de nature projective. Or il y a une sorte de principe en théorie des catégories qui veut qu'un tel problème soit équivalent à un "petit" problème universel de nature inductive. Par exemple, l'objet initial d'une catégorie peut être vu indifféremment comme la limite projective de "tout" ou la limite inductive de "rien".

Autre exemple: la dualité de Stone-Weierstrass permet de ramener, du point de vue des modèles, l'étude des petites esquisses inductives à celle de grosses esquisses projectives. Ce principe justifie l'idée, proposée en [3], d'aborder la sémantique des types par le versant inductif, d'autant plus que celui-ci est le seul à avoir une représentation informatique. Il reste à espérer un jour un énoncé précis faisant la jonction entre les deux points de vue (le théorème I-6 de [2], établissant un lien entre "liaison projective" et "algèbre initiale", est un pas important dans cette direction).

Dans [3], l'idée de départ était celle, défendue depuis longtemps par l'"école d'Ehresmann" (voir par exemple [6]), qui consiste à prendre au sérieux la structure interne des types. Autrement dit, on cesse d'assimiler un type clos à un seul objet de $\mathbb{C}$, pour le voir plutôt comme un diagramme dans $\mathbb{C}$, figurant tous les domaines auxiliaires, fonctions de construction, fonctions d'accès, etc. nécessaires à sa définition. Evidemment, les différents objets et flèches de ce diagramme ne sont pas indépendants, et il nous faut trouver un moyen de spécifier formellement les liens qui existent entre eux.

Si ces liens étaient tous de simples limites projectives ou inductives, on pourrait utiliser des esquisses. Mais pour spécifier aussi des propriétés universelles plus complexes, car à plusieurs étages (comme celles définissant une exponentielle ou un objet des entiers naturels), il nous faut recourir à une notion d'esquisse généralisée, comme celle de *trame* ([4]), ou celle, plus souple encore, et qui s'en inspire directement, de *canevas*.

2 - CANEVAS, LOGIQUES ET SEMANTIQUES

Un *canevas* $\mathbb{T}$ est une syntaxe géométrique abstraite, qui consiste en un compographe (son *support*) et un certain nombre de *spécifications* (comparer avec esquisse = support + cônes, voir [6]). Ces spécifications définissent une propriété "universelle" (en un sens très général) exprimée elle-même dans le langage des canevas. On note Cpgraph la catégorie des petits compographes.

DEFINITION DES CANEVAS ET HOMOMORPHISMES DE CANEVAS

a) $\mathbb{T} = (\underline{\mathbb{T}} , O)$ est un canevas d'ordre 0 ssi

· $\underline{\mathbb{T}}$ est un compographe

b) $H = (\mathbb{T}', \underline{H}, \mathbb{T})$ est un homomorphisme de canevas d'ordre 0 ssi

· $\mathbb{T}'$, $\mathbb{T}$ sont des canevas d'ordre 0

· $\underline{H} : \underline{\mathbb{T}}' \to \underline{\mathbb{T}}$ est un foncteur

c) $\mathbb{T} = (\underline{\mathbb{T}}, \mathscr{K}(\mathbb{T}))$ est un canevas d'ordre $\leq n+1$ ssi

· $\underline{\mathbb{T}}$ est un compographe (appelé support de $\mathbb{T}$)

· $\mathscr{K}(\mathbb{T})$ est une classe de quintuplets (dits spécifications)

$q = (F':\underline{\mathbb{I}}\to\underline{\mathbb{T}},\ H:\mathbb{I}\to\mathbb{J},\ F:\underline{\mathbb{J}}\to\underline{\mathbb{T}},\ J:\mathbb{J}\to\mathbb{K},\ J':\mathbb{J}\to\mathbb{K})$

où H,J,J' sont des homomorphismes de canevas d'ordre $\leq n$

F,F' sont des foncteurs

$F\cdot\underline{H} = F'$ et $\underline{J}\cdot\underline{H} = \underline{J}'\cdot\underline{H}$

d) $H = (\mathbb{T}', \underline{H}, \mathbb{T})$ est un homomorphisme de canevas d'ordre $\leq n+1$ ssi

· $\mathbb{I}'$, $\mathbb{T}$ sont des canevas d'ordre $\leq n+1$

· $\underline{H} : \underline{\mathbb{T}}' \to \underline{\mathbb{T}}$ est un foncteur tel que

$(F',H,F,J,J') \in \mathscr{K}(\mathbb{T}') \Rightarrow (\underline{H}\cdot F', H', \underline{H}\cdot F, J, J') \in \mathscr{K}(\mathbb{T})$.

Les canevas petits (i.e. ceux pour lesquels "tout est petit") (resp. les homomorphismes entre canevas petits) sont clairement les objets (resp. les flèches) d'une catégorie localement petite que nous noterons Can. On peut identifier Cpgraph à la sous-catégorie pleine de Can dont les objets sont les canevas d'ordre 0.

DEFINITION DES MODELES

$M = (\mathbb{T}, \underline{M}, \mathbb{C})$ est un modèle (de $\mathbb{T}$ dans $\mathbb{C}$) ssi

1) $\mathbb{T}$ est un canevas

2) $\mathbb{C}$ est une catégorie

3) $\underline{M} : \mathbb{T} \to \mathbb{C}$ est un foncteur

4) quel que soit $(F', H, F, J, J') \in \mathcal{P}(\mathbb{T})$,

- $M \cdot F' \underset{def}{=} (\mathbb{I}, \underline{M} \cdot F', \mathbb{C})$ est un modèle
- $M \cdot F \underset{def}{=} (\mathbb{J}, \underline{M} \cdot F, \mathbb{C})$ est un modèle
- pour tout modèle $N = (\mathbb{J}, \underline{N}, \mathbb{C})$ tel que

$$N \cdot H \underset{def}{=} (\mathbb{J}, \underline{N} \cdot \underline{H}, \mathbb{C}) = M \cdot F' ,$$

il existe un unique modèle $R = (\mathbb{K}, \underline{R}, \mathbb{C})$ vérifiant

$$\underline{R} \cdot \underline{J} = \underline{M} \cdot \underline{F} \quad \text{et} \quad \underline{R} \cdot \underline{J}' = \underline{N} .$$

La classe des modèles de $\mathbb{T}$ dans $\mathbb{C}$ sera notée $\mathrm{Mod}(\mathbb{T}, \mathbb{C})$. A noter que nous ne cherchons pas à définir une notion d'homomorphisme entre modèles.

Si $H : \mathbb{I} \to \mathbb{J}$ est un homomorphisme de canevas et $\mathbb{C}$ une catégorie, on dispose de l'application

$$\mathrm{Mod}(H, \mathbb{C}) : \mathrm{Mod}(\mathbb{J}, \mathbb{C}) \to \mathrm{Mod}(\mathbb{I}, \mathbb{C})$$

qui au modèle $M = (\mathbb{J}, \underline{M}, \mathbb{C})$ associe $M \cdot H \underset{def}{=} (\mathbb{I}, \underline{M} \cdot \underline{H}, \mathbb{C})$ dont on vérifie que c'est bien un modèle.

DEFINITION DES SYNTAXES DE TYPE

Une syntaxe de type est un triplet d'homomorphismes de canevas $\sigma = (H:\mathbb{I}\to\mathbb{J}, J:\mathbb{J}\to\mathbb{K}, J':\mathbb{J}\to\mathbb{K})$ avec $J\cdot H = J'\cdot H$.

Si $\mathbb{T}$ est un canevas et σ une syntaxe de type, on pose $\mathscr{X}_\sigma(\mathbb{T}) = \{ (F',H,F,J,J') \in \mathscr{X}(\mathbb{T}) \ / \ (H,J,J') = \sigma \}$.

DEFINITION DES LOGIQUES

Une logique Σ est un ensemble de syntaxes de type.

Si Σ est une logique et $\mathbb{T}$ un canevas, on dit que $\mathbb{T}$ est subordonné à Σ ssi $\mathscr{X}(\mathbb{T}) = \bigcup_{\sigma\in\Sigma} \mathscr{X}_\sigma(\mathbb{T})$.

On note $Can(\Sigma)$ la sous-catégorie pleine de Can dont les objets sont les canevas subordonnés à Σ. Par exemple, $Can(\mathbb{O})$ s'identifie à Cpgraph.

On dit qu'une logique Σ est héréditaire ssi:
$\forall \sigma = (H:\mathbb{I}\to\mathbb{J}, J:\mathbb{J}\to\mathbb{K}, J':\mathbb{J}\to\mathbb{K}) \in \Sigma$, $\mathbb{I}$, $\mathbb{J}$, $\mathbb{K}$ sont subordonnés à Σ (il suffit de le demander pour $\mathbb{K}$).

Plus particulièrement, si α est un cardinal régulier, on dit que Σ est α-héréditaire ssi $\forall \sigma = (H:\mathbb{I}\to\mathbb{J}, J:\mathbb{J}\to\mathbb{K}, J':\mathbb{J}\to\mathbb{K}) \in \Sigma$, $\mathbb{I}$, $\mathbb{J}$, $\mathbb{K}$ sont des objets α-présentables de $Can(\Sigma)$.

DEFINITION DES SEMANTIQUES CATEGORIQUES D'UNE LOGIQUE ET DES HOMOMORPHISMES DE SEMANTIQUES

($\mathbb{C}$, $(a_\sigma)_{\sigma \in \Sigma}$) est une sémantique catégorique pour la logique Σ ssi

1) $\mathbb{C}$ est une catégorie

2) pour tout $\sigma = (H:\mathbb{I} \to \mathbb{J}, J:\mathbb{J} \to \mathbb{I}, J':\mathbb{J} \to \mathbb{I}) \in \Sigma$,

a_σ est une application de $Mod(\mathbb{I},\mathbb{C})$ dans $Mod(\mathbb{J},\mathbb{C})$

(dite type dans $\mathbb{C}$ de syntaxe H) telle que

a) $Mod(H,\mathbb{C}) \circ a_\sigma = Id_{Mod(\mathbb{I},\mathbb{C})}$

b) $\forall M' \in Mod(\mathbb{I},\mathbb{C})$, $\forall N \in Mod(\mathbb{J},\mathbb{C})$,

$N \cdot H = M' \Rightarrow \exists ! R \in Mod(\mathbb{K},\mathbb{C}) / R \cdot J = a_\sigma(M')$ et $R \cdot J' = N$.

Soient ($\mathbb{C}$, $(a_\sigma)_{\sigma \in \Sigma}$) et ($\mathbb{C}'$, $(a'_\sigma)_{\sigma \in \Sigma}$) deux sémantiques catégoriques d'une même logique Σ. Un homomorphisme de la première vers la seconde est un foncteur $G: \mathbb{C} \to \mathbb{C}'$ tel que

$\forall \sigma = (H:\mathbb{I} \to \mathbb{J}, J:\mathbb{J} \to \mathbb{I}, J':\mathbb{J} \to \mathbb{I}) \in \Sigma$, $\forall M \in Mod(\mathbb{I},\mathbb{C})$,

$G \cdot M = (\mathbb{I}, \underline{G} \cdot \underline{M}, \mathbb{C}')$ est un modèle,

$G \cdot a_\sigma(M) = (\mathbb{J}, \underline{G} \cdot \underline{a_\sigma(M)}, \mathbb{C}')$ est un modèle,

et $a'_\sigma(G \cdot M) = G \cdot a_\sigma(M)$.

Les sémantiques catégoriques petites (resp. les homomorphismes entre ces sémantiques) d'une logique Σ sont clairement les objets (resp. les flèches) d'une catégorie localement petite que nous noterons $Sém(\Sigma)$. Par exemple, $Sém(\mathbb{O})$ s'identifie à Cat.

DEFINITION DE DIVERS FONCTEURS D'OUBLI

On dispose d'un certain nombre de foncteurs d'oubli intéressants:

1) U_Σ : Sém(Σ) -> Cat
qui envoie la sémantique ($\mathcal{C}$, $(a_\sigma)_{\sigma\in\Sigma}$) sur la catégorie $\mathcal{C}$

Si $\Sigma'\subset\Sigma$, on a plus généralement un foncteur d'oubli $U_{\Sigma,\Sigma'}$: Sém(Σ) -> Sém(Σ') . Pour $\Sigma' = \emptyset$, on retrouve le foncteur U_Σ .

On a également un foncteur d'oubli canonique B_Σ: Sém(Σ) -> Graph, en composant U_Σ avec l'oubli B_0: Cat -> Graph.

2) V_Σ : Sém(Σ) -> Can(Σ)
qui envoie la sémantique ($\mathcal{C}$, $(a_\sigma)_{\sigma\in\Sigma}$) sur le canevas ($\mathcal{C}$, { (M',H,a (M'),J,J') / = (H:$\mathcal{I}$->$\mathcal{J}$, J:$\mathcal{J}$->$\mathcal{C}$, J':$\mathcal{J}$->$\mathcal{I}$) $\in\Sigma$ et M' $\in$ Mod($\mathcal{I}$,$\mathcal{C}$) })

Tout l'intérêt du foncteur V_Σ réside dans le fait que pour tout canevas $\mathcal{T}$ subordonné à Σ , les homomorphismes de $\mathcal{T}$ dans $V_\Sigma((\mathcal{C}, (a_\sigma)_{\sigma\in\Sigma}))$ s'identifient aux modèles de $\mathcal{T}$ dans $\mathcal{C}$; autrement dit, V_Σ coreprésente les foncteurs Mod(-,$\mathcal{C}$) : Can(Σ)^{op} -> Ens.

Pour $\Sigma = \emptyset$, V_Σ n'est autre que le foncteur d'oubli canonique V_0 : Cat -> Cpgraph.

3 - QUELQUES EXEMPLES

On se reportera aux planches des syntaxes pages 11 et 12.

exemple 0. Soit $\Sigma_I = \{ \sigma_{un}, \sigma_{prod}, \sigma_{exp} \}$: Σ_I est appelée *logique intuitionniste du premier ordre.* Ses sémantiques sont exactement les catégories cartésiennes fermées.

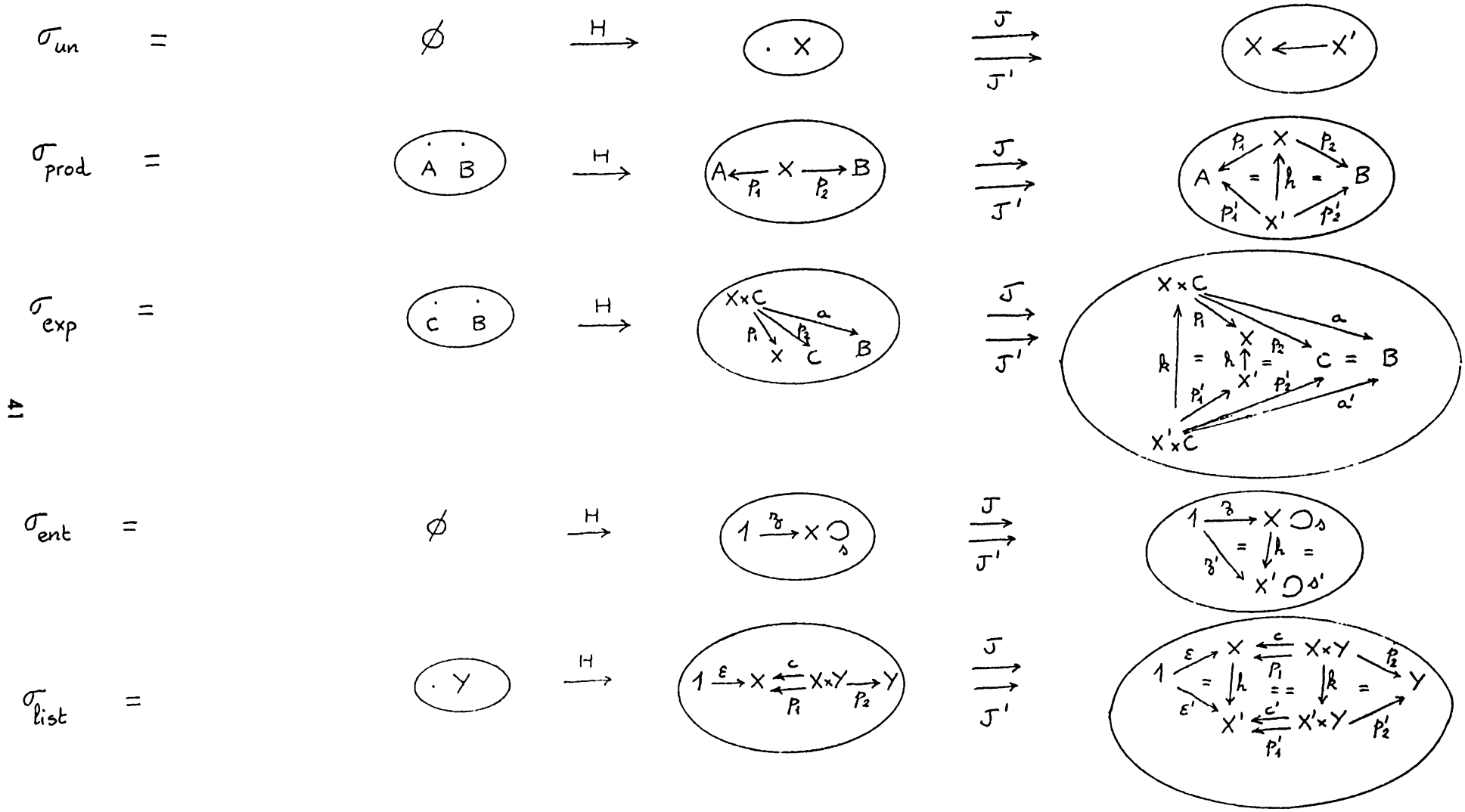
A tout type A du système F, on peut, par un procédé de décodage systématique (partiellement décrit dans [3]), associer une certaine "syntaxe de type pointée"
$\sigma_A = (H:\mathbb{H}\to\mathbb{J} , J:\mathbb{J}\to\mathbb{K} , J':\mathbb{J}\to\mathbb{J} , \psi:1\to\mathbb{J})$
avec les intentions suivantes:
• $\mathbb{J}$ décrit la structure interne du type A;
• H (resp. ψ) pointe dans $\mathbb{J}$ les emplacements des variables libres (resp. du domaine principal) de A;
• J et J' traduisent la propriété universelle attendue.

On dit que A est sémantisable dans une catégorie $\mathbb{C}$ ssi $\mathbb{C}$ est (sous-jacente à) une sémantique de $\Sigma_I \cup \{\sigma_A\}$.

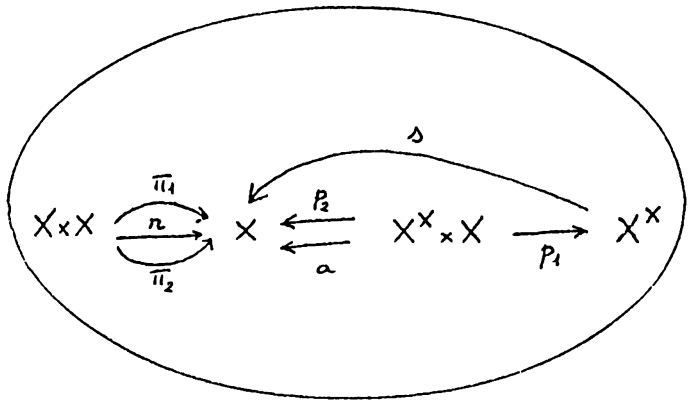
exemple 1. Au type clos ent = $\forall X.X\Rightarrow((X\Rightarrow X)\Rightarrow X)$, on associe σ_{ent} représentée page 11. Ce type est sémantisable dans $\mathbb{C}$ ssi $\mathbb{C}$ est une catégorie cartésienne fermée avec objet des entiers naturels.

exemple 2. Au type à une variable libre list(Y) = $\forall X.X\Rightarrow((X\Rightarrow(Y\Rightarrow X))\Rightarrow X)$, on associe σ_{list} représentée page 11. Ce type s'interprète alors comme "type des listes d'entités de type Y". En "branchant" convenablement σ_{ent} sur σ_{list}, on dessine facilement la syntaxe géométrique du type clos list(ent).

exemple 3. Au type clos (type du λ-calcul pur) lambda = $\forall X.(X\Rightarrow(X\Rightarrow X))\Rightarrow(((X\Rightarrow X)\Rightarrow X)\Rightarrow X)$, on associe σ_{lambda} représentée page 12. Ce type n'est pas un type naturel au sens de [3], le problème universel dont il est solution est donc plus compliqué que dans les deux exemples précédents (problèmes de co/contravariance).

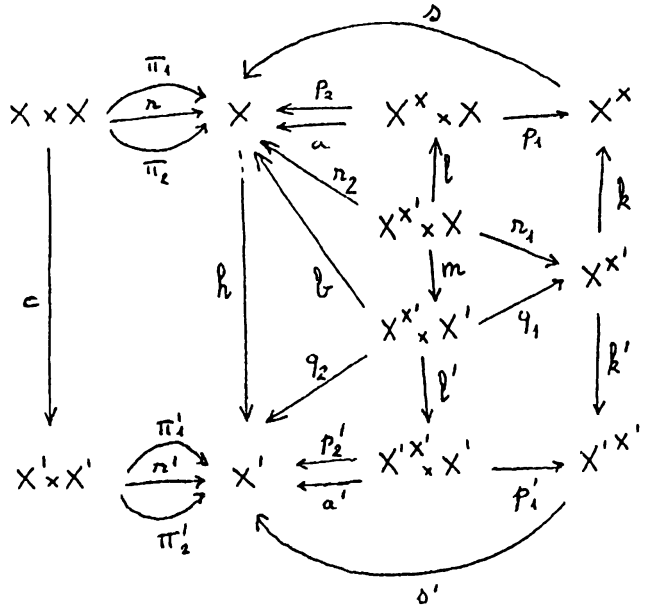

σun = ∅ H · X J J' X ← X'
σprod = A B H A ← P1 X P2 → B J J' P1 P2 A = h = B P1' P2' X X'
σexp = C B H X×C a P1 P2 X C B J J' X×C a P1 P2 X = h = C = B h P1' P2' X' a' X'×C
σent = ∅ H 1 → X ↻ J J' 1 → X ↻ = h = X' ↻
σlist = · Y H 1 → X ⇄ X×Y → Y J J' ε c P1 P2 1 = h = k = Y ε' c' X' X'×Y P1' P2'

$\sigma_{\text{lambda}} = \phi$



H





avec les équations :

$p_1 l = k r_1$ $\quad p_2 l = r_2$

$p'_1 l' = k' q_1$ $\quad p'_2 l' = q_2$

$a l = b m$ $\quad a' l' = h b$

$q_1 m = r_1$ $\quad q_2 m = h r_2$

$h s k = s' k'$ $\quad h r = r' c$

$h \pi_1 = \pi'_1 c$ $\quad h \pi_2 = \pi'_2 c$

4 - THEOREME D'ALGEBRICITE

On démontre (voir [5]):

théorème. Soient α un cardinal régulier et Σ une logique α-héréditaire. Alors la catégorie Sém(Σ) est localement α-présentable, et le foncteur d'oubli canonique V_Σ : Sém(Σ) -> Can(Σ) admet un adjoint à gauche et est monadique de rang α.

remarque 1. On en déduit en particulier que la structure de catégorie à choix de $\mathcal{G}$-limites projectives et $\mathcal{G}'$-limites inductives est algébrique (monadique) par rapport à celle de $(\mathcal{G},\mathcal{G}')$-esquisse (ce seul cas particulier semblant déjà nouveau).

remarque 2. Le foncteur U_{Σ} admet aussi un adjoint à gauche; mais contrairement à V_{Σ}, il n'est pas nécessairement monadique. A ce sujet, il faut noter que si l'algébricité sur les catégories (ou même sur les graphes) est une situation remarquable sur le plan mathématique, elle n'est pas très utile pour les applications informatiques: en effet, si les tests de commutativité sont un obstacle au calcul, il n'en va pas de même de la spécification de types déjà construits (cf les combinateurs catégoriques de Lambek et leur utilisation dans la Categorical Abstract Machine de Cousineau, Curien et Mauny).

BIBLIOGRAPHIE

[1] J.-Y. Girard, Proofs and types, Cambridge University Press, 1989
(version originale française dactylographiée sous le titre Lambda-calcul typé)

[2] P.J. Freyd, Structural polymorphism, draft (23 jan.89)

[3] P. Ageron, Sémantique catégorique des types: comprendre le système F, Diagrammes 19, 1988

[4] C. Lair, Trames et sémantiques catégoriques des systèmes de trames, Diagrammes 18, 1987

[5] P. Ageron, Logiques, catégories et esquisses, thèse de doctorat, à soutenir

[6] L. Coppey et C. Lair, Leçons de théorie des esquisses (I) *Diagrammes* 12, 1984 et (II) *Diagrammes* 19, 1988