La logique combinatoire : logique de l'objet quelconque ou logique de l'opérateur ?

Combinatory logic: a logic of any objects whatever or a logic of operators?

Jean-Pierre Ginisti

Édition électronique

URL : <http://journals.openedition.org/msh/2874>

DOI : 10.4000/msh.2874

ISSN : 1950-6821

Éditeur

Centre d'analyse et de mathématique sociales de l'EHESS

Édition imprimée

Date de publication : 1 mars 2003

ISSN : 0987-6936

Référence électronique

Jean-Pierre Ginisti, « La logique combinatoire : logique de l'objet quelconque ou logique de l'opérateur ? », *Mathématiques et sciences humaines* [En ligne], 162 | Été 2003, mis en ligne le 10 février 2006, consulté le 23 juillet 2020. URL : <http://journals.openedition.org/msh/2874> ; DOI : <https://doi.org/10.4000/msh.2874>

LA LOGIQUE COMBINATOIRE : LOGIQUE DE L'OBJET QUELCONQUE OU LOGIQUE DE L'OPÉRATEUR ?

Jean-Pierre GINISTI¹

RÉSUMÉ – *La logique combinatoire est-elle une logique de l'objet quelconque ou une logique de l'opérateur ? (1) Les engagements philosophiques de la logique combinatoire sont expliqués à partir de plusieurs principes, pas toujours explicites dans la littérature (pansémie, formalisation complète, neutralité ontologique, réversibilité, libre application). (2) Des exemples (connus, puis nouveaux) du style combinatoire sont donnés pour identifier le programme de cette logique. Les contributions techniques de l'auteur sont (principalement) : l'élimination des foncteurs propositionnels par des nouveaux combinateurs (des distributeurs d'objets sémantiques), puis par des combinateurs classiques. La curryfication et la théorie de la fonctionnalité sont reconsidérées. Partout, la notion d'objet quelconque s'avère capitale.*

MOTS-CLÉS – Logique combinatoire, Formalisation, Ontologie, Curryfication

SUMMARY – *Combinatory logic: a logic of any objects whatever or a logic of operators? Is combinatory logic a logic of any objects whatever or a logic of operators ? (1) The philosophical commitments of combinatory logic are explained from several principles, not always explicit in literature (pansemy, complete formalisation, ontological neutrality, reversibility, free application). (2) Examples (known, then new) of the combinatory style are given in order to identify the program of this logic. The technical contributions of the author are (mainly) : the elimination of propositional functors by new combinators (distributors of semantical objects), then by classical combinators. The process called "currying" and the theory of functionality are revisited. Everywhere, the notion of "any object whatever" turns out to be capital*

KEYWORDS – Combinatory logic, Formalization, Ontology, Curryfication

Notre problème sera d'identifier ce à quoi s'adresse exactement la logique combinatoire, la nature des objets sur lesquels porte sa théorie. Il y a en effet certaines divergences, au moins apparentes, entre les interprètes : pour les uns, en bref, la logique combinatoire est une logique de l'objet quelconque, pour les autres, une logique de l'opérateur. On peut donc préciser les deux optiques, examiner dans quelle mesure elles s'opposent et tenter d'arbitrer entre elles. Avant examen, on entendra par *opérateur*, de la manière la plus générale, un être qui conduit à un certain résultat quand il porte, selon ce qu'il exige, sur une ou plusieurs données, ses opérandes². D'autre part, un objet (dans

¹ Professeur à l'Université Lyon III Jean Moulin, Faculté de philosophie, 1 rue de l'Université, B.P. 0638, 69239 Lyon.

² En ce sens, nous écrirons «*Opérateur*_i» quand il faudra le distinguer de certains autres. Nous écrirons «*LC*» pour «*Logique combinatoire*». Nous nous accorderons quelque liberté avec l'emploi des

une classe qui le spécifie) est dit *quelconque* si on a égard seulement à des propriétés qui sont celles de tous les autres objets (de cette classe), même si on lui connaît des propriétés singulières. 2 est un nombre premier *quelconque* relativement à la définition de «Nombre premier», 2 n'est pas un nombre premier *quelconque* relativement à sa propriété d'être pair. Un objet (non spécifié, donc *quelconque* en un deuxième sens, capital en LC) est *quelconque* (au premier sens), si dans chaque interprétation (où il appartiendra évidemment à une certaine classe) il est traité en ayant égard seulement à des propriétés qui seront celles de tous les autres objets de cette classe.

Nous commencerons par rappeler comment se présente aujourd'hui la morphologie (c'est-à-dire l'alphabet et les règles de formation) d'un système de logique combinatoire.

1. LA MORPHOLOGIE COMBINATOIRE

Si la LC est restreinte strictement à elle-même, au traitement exclusif de ses propres concepts, on dit qu'elle est *pure*. Si elle s'adjoint des éléments (par exemple des quantificateurs) qui sont destinés à permettre, à retrouver ou à faciliter le traitement des problèmes usuels de la logique, on dit qu'elle est *illative*.

La logique combinatoire pure, sous une forme qu'on peut dire *basale*, a besoin de l'alphabet suivant. Nous le formulerons aussi formellement que possible, pour éviter de préjuger ce à quoi on s'adresse.

- Cinq éléments de première espèce : **I**, **K**, **W**, **C**, **B** (dont nous dirons qu'ils forment l'ensemble E) ;
- Un élément de deuxième espèce, noté $\square$;
- Un élément de troisième espèce, noté $\sqcap$;
- Deux éléments de quatrième espèce, notés $(,)$ ³.

On se donne alors les règles de formation suivantes :

- Un élément de E est un objet ;
- Si x_1 et x_2 sont des objets, $(x_1 \square x_2)$ est un objet ;
- Rien d'autre n'est un objet ;
- Si x_1 et x_2 sont des objets, $x_1 \sqcap x_2$ est un énoncé ;
- Rien d'autre n'est un énoncé.

guillemets qui devraient toujours distinguer la mention de l'usage des symboles. Notre analyse portera davantage sur des concepts que sur des techniques. Nous avons présenté celles-ci par ailleurs, voir [15], [16]. Rappelons que la LC est issue d'un article de Schönfinkel, [29], et doit l'essentiel de son développement à Curry.

³ Cette version est commode, mais non minimale : (i) En remplaçant **I**, **W**, **C**, **B** par **S** (note 4), deux éléments suffiraient en première espèce. (ii) On peut éviter les éléments de quatrième espèce en notant de manière préfixée l'élément de deuxième espèce (voir, ici, en IV). En outre, cette version est basale car elle peut réaliser l'accord (ou un très large accord) entre les logiciens d'obédience combinatoire, mais on se demandera s'il lui manque encore des éléments acceptables.

x_1 et x_2 (ainsi que $x_3, x_4, \dots$) sont des métavariabes. Nous utiliserons aussi x, y, z, w , selon commodité. L'objet est atomique s'il appartient à E . Il est moléculaire dans le cas contraire. Dans une expression de forme $(x_1 \sqsupset x_2)$ nous omettrons l'élément $\sqsupset$ en écrivant $(x_1 x_2)$, comme on écrit $2x$ pour $2 \sqsupset x$. Nous omettrons aussi les éléments $(,)$ qui entourent des objets deux par deux à partir de la gauche dans une expression. Ainsi, $((x_1 x_2) x_3) x_4$ s'écrira $x_1 x_2 x_3 x_4$, $((x_1 x_2)(x_3 x_4))$ s'écrira $x_1 x_2 (x_3 x_4)$. La convention vaudra aussi pour les sous-expressions délimitées par des éléments $(, \sqsupset)$ non évitables ; $(x_1((x_2 x_3) x_4))$ s'écrira $x_1(x_2 x_3 x_4)$. Des mesures comparables d'association à gauche, (AG), seront prises dans les autres systèmes dont nous parlerons.

Pour s'exprimer commodément, on peut dénommer les différents ingrédients, mais cela présente le risque, comme on le verra, de ramener la LC à des approches qui ne sont pas les siennes.

- **I, K, W, C, B** sont des objets dits *combinateurs* ;
- $\sqsupset$ est un opérateur binaire dit *application* (même si *applicateur* conviendrait mieux) $\sqsupset$
- $\sqsupset$ est une relation binaire dite *réductibilité* ;
- $(, \sqsupset)$ sont des signes de ponctuation dits, respectivement, *parenthèse gauche* et *parenthèse droite*.

D'autre part, un objet dont chaque atome est un combinateur sera dit lui-même combinateur (combinateur moléculaire).

Aux combinateurs primitifs sont associées des règles dites *de réduction*,

à savoir :

$$\begin{aligned} \mathbf{I}x_1 &\sqsupset x_1 \\ \mathbf{K}x_1 x_2 &\sqsupset x_1 \\ \mathbf{W}x_1 x_2 &\sqsupset x_1 x_2 x_2 \\ \mathbf{C}x_1 x_2 x_3 &\sqsupset x_1 x_3 x_2 \\ \mathbf{B}x_1 x_2 x_3 &\sqsupset x_1(x_2 x_3) \end{aligned}$$

mais qu'on doit lire, en vertu de (AG) :

$$\begin{aligned} (\mathbf{I}x_1) &\sqsupset x_1 \\ ((\mathbf{K}x_1)x_2) &\sqsupset x_1 \\ ((\mathbf{W}x_1)x_2) &\sqsupset ((x_1 x_2)x_2) \\ (((\mathbf{C}x_1)x_2)x_3) &\sqsupset ((x_1 x_3)x_2) \\ (((\mathbf{B}x_1)x_2)x_3) &\sqsupset (x_1(x_2 x_3)) \quad ^4 \end{aligned}$$

On s'expliquera plus loin sur le sens de ces formules car une partie importante de notre sujet dépend de la manière dont il faut comprendre la colonne de droite. On désignera par **(I)**, **(K)**, etc. chacune de ces règles. Une expression de forme $x \sqsupset y$ se lit « $\sqsupset$ se réduit à $y\sqsupset$ », mais on peut la comprendre « $\sqsupset$ on a l'objet x , alors on a l'objet $y\sqsupset$ », « $\sqsupset$ suffit d'avoir l'objet x pour avoir l'objet $y\sqsupset$ », comme on a $p \rightarrow q$ (toutefois, ' $\rightarrow$ ' n'est pas une implication, puisque x et y ne sont pas des propositions). La relation ' $\sqsupset$ ' admet aussi les règles $\sqsupset$

$$(\sqsupset) \quad x \sqsupset x$$

réflexivité.

$$(\sqsupset) \quad x \sqsupset y : zx \sqsupset zy$$

règle de monotonie droite.

$$(\sqsupset) \quad x \sqsupset y : xz \sqsupset yz$$

règle de monotonie gauche.

$$(\sqsupset) \quad x \sqsupset y, y \sqsupset z : x \sqsupset z$$

règle de transitivité⁵.

- Q est dit *combinaison* (applicative) des x_i .
- On peut établir que tout combinateur X , différent des combinateurs **I**, **K**, **W**, **C**, **B**, est définissable par ces derniers.
- X est dit (quand il y a lieu de le préciser) *combinateur classique*.
- On écrira (X) la règle de réduction du combinateur X , à savoir $X \rightarrow Q$.
- Dans une expression de forme $x \rightarrow y$, y sera dite la *résultante* (de l'expression x).

Toutefois, on peut ajouter dans E une infinité d'éléments notés a_1, a_2 , etc. Ces éléments qui sont dits des *indéterminées* pourront correspondre à des données importées (non originellement combinatoires). La logique combinatoire reste pure car ces éléments n'ont pas de contenus spécifiés. Cette adjonction d'indéterminées constitue un autre langage formel. Il sera nommé LC_2 . On peut alors

1. continuer à définir les combinateurs sur des objets quelconques (qui comprennent aussi désormais a_1, a_2 , etc.)
2. le faire sur des métavariabes d'indéterminées a_i, a_j, a_k , etc.
3. le faire sur les n premières indéterminées a_1, a_2 , etc. puisqu'une indéterminée ne diffère pas d'une autre.

Comme il est de la nature d'une indéterminée de convenir à n'importe quel type d'objet, n'importe quel objet peut en prendre la place sans qu'on ait besoin d'une règle de substitution ; c'est pourquoi des expressions comme $Ka_i a_j$, $Ka_1 a_2$, $Kx_1 x_2$ se rapportent en fait aux mêmes données et conviennent à tous les objets de la morphologie. On ne peut pas opposer notamment une forme de la LC qui s'adresserait à des objets quelconques et une forme de LC qui s'adresserait seulement à des indéterminées. Même formulée sur ces dernières, la LC convient aux premiers, une indéterminée pouvant représenter un combinateur⁷.

2. LA NOTION D'OBJET

Le mot «objet» est évidemment délicat. On pourrait certes l'éviter et dire par exemple à sa place «entité» [2], «terme», «expression combinatoire», «ob», etc., mais on n'échapperait pas au problème d'expliquer ce qui fait de **W** par exemple une entité, ou pourquoi on le pose comme expression combinatoire, etc. De manière générale on peut comprendre le mot «objet» ainsi :

- (1) Comme «représentation», au sens étymologique de «objectum», ce qui est présent à l'esprit (rêvant, imaginant, etc.) et qui ne suppose pas d'existant associé, ce qu'on peut nommer, bien qu'il y ait pléonasme à le faire au sens (1) du mot «objet», un «objet de pensée».

⁷ Pour certains objectifs, il importe de définir les combinateurs sur des indéterminées plutôt que sur des métavariabes d'objet [6], p. 189-190. D'autre part, l'intérêt de le faire sur des indéterminées est de rester sur le plan de la langue-objet. Ici, nous garderons l'expression en métavariabes pour avoir une formulation générale plus directe.

- (2) Comme «chose» (par exemple dans l'expression «Un objet de première nécessité»), en un sens opposé au premier, mais plus actuel, ce qui constitue un existant hors de la représentation, en laissant ouverte la question, posée par les métaphysiciens idéalistes, de savoir si la pensée rigoureuse peut appliquer le concept. Souvent cette chose est conçue comme une «donnée immédiate» qu'on peut laisser non analysée pour l'entreprise. L'objet est une entrée ou une sortie pour un traitement. On suppose ainsi (dans un premier temps) des ensembles élémentaires qui collectent des objets bruts, ou des fonctions qui associent un objet-valeur à un objet-argument.
- (3) Comme «But» («Un objet de ma visite est de vous remercier»).
- (4) Comme «Contenu», ce qui fournit une matière à une activité ou à une production intellectuelle («Un objet d'une note de service»).

Cela ne peut pas être au sens (3) ou (4) que **W** est dit un objet. «**W** est un but de l'analyse», «**W** a un contenu» ne diraient rien sur **W**. D'autre part, le sens (1) à lui seul ne peut pas faire comprendre pourquoi $\square$, $\square$, $(,)$ ne sont pas dits des objets, puisqu'ils constituent aussi des objets de pensée. Le concept même d'énoncé paraît de trop puisqu'un énoncé est un objet de pensée. On pourrait à vrai dire ranger tous les éléments dans **E**. On obtiendrait par exemple « $(\square \square)$ est un objet». L'expression est étrange, mais ce n'est pas ce caractère qui la disqualifie. Dans l'acception (1) le mot «objet» devrait s'appliquer à tout élément et à tout assemblage. Il perdrait donc son intérêt. Pour lui rendre sa force, il faut qu'il s'oppose à des éléments qui n'ont pas la même auto-suffisance. Il s'ensuit que, tout élément n'étant pas classifié comme objet, ce qui porte cette appellation a quelque statut plus éminent, et il faut donc dire lequel.

Le sens (2) contraint à une interprétation forte des combinateurs. Si **W** est une chose, de quelle chose s'agit-il ? Il est évidemment possible de soutenir un «platonisme», les combinateurs sont des existants hors de l'espace et du temps, mais (comme le dit Rosenbloom de manière plus générale) on peut aussi les considérer comme «Existant dans l'espace-temps» en les identifiant, par exemple, à certaines réactions chimiques dans les cerveaux des hommes. Ceci n'est pas la seule interprétation possible, mais l'existence d'au moins une telle interprétation montre qu'on peut travailler avec ces entités sans supposer une métaphysique comme l'idéalisme platonicien ([26], p. 201). Curry soutenait aussi que le «platonisme» n'est pas impliqué par la LC ([7], p. 11).

La position de Rosenbloom, toutefois, a le défaut de laisser entendre qu'une entité «Abstraite» (par exemple un combinateur) est susceptible d'avoir une interprétation (ou plusieurs interprétations) en termes de référent concret, mais que celui-ci demeure conjectural. Elle laisse donc une grande fragilité à l'entreprise formelle. Pour la supprimer, il convient d'interpréter un combinateur comme étant n'importe quel intervenant qui se comporte dans son domaine selon une règle de réduction. En ce sens, un combinateur est bien un objet quelconque malgré la règle de réduction qui semble en faire un objet précis. Pour fixer les idées (mais seulement pour cela !), disons que **W** c'est indifféremment un mécanisme linguistique de redoublement d'une consonne, de formation du pronom réflexif *se*, une cause psycho-physiologique du bégaiement, un procédé biologique de clonage, etc. Une entité abstraite c'est une réalité concrète dont certains aspects sont

indifférents à l'objectif. Les objets dont parle le système, dit Curry, sont «*essentiellement des blancs dans lesquels on peut faire entrer tout objet qu'on veut*» ([8], p. 93).

Certes, on peut retrouver un «*platonisme*» à partir de cette interprétation puisque le «*platonisme*» consiste à dire que toute ressemblance à l'égard d'une certaine propriété impose un être méta-empirique qui possède (ou, plutôt, qui soit en lui-même) cette propriété, et auquel participent les semblables. Mais qu'on aille ou non jusque-là, on sait déjà de quoi on parle dans ce qui a été dit d'un combinateur.

Il est important que Curry traite les objets comme «*des blancs*» et qu'il refuse de les identifier (avec Hilbert) aux objets matériels que sont les symboles, en faisant valoir qu'il s'agirait encore, justement, d'un type particulier d'objet. Il en irait de même si on disait qu'on élabore une «*physique de l'objet quelconque*», selon l'une des formules que Gonsseth applique à toute logique ([17], p. 69 *sq.*), et on peut comprendre que Dopp la récuse pour la LC ([12], p. 585-6). Piaget la récuse aussi, et plus généralement, en objectant qu'en logique il s'agit «*d'opérations sur l'objet quelconque*» ([24], p. 6, note). C'est tout à fait le cas en LC puisqu'une certaine opération, l'application, à la fois bien spécifiée et universelle y est une notion première.

3. NORMES ET PROGRAMME DE LA LOGIQUE COMBINATOIRE

Ce qui motive la LC semble pouvoir être présenté selon la ligne suivante d'arguments :

- (1) L'idée la plus fondamentale peut s'exprimer sous la forme de ce qu'on appellera *le principe de pansémie* : tout objet de pensée (toute production mentale) possède un sens à quelque égard. Les deux premiers articles de Curry le soulignent fortement dès leur début, et l'idée revient très souvent dans les écrits postérieurs. Il serait contradictoire en effet de dire qu'un objet de pensée est à la lettre impensable, et donc rien de ce que la pensée prend pour objet (qu'on le dise raisonnable, incertain, extravagant, absurde, antinomique, etc.) ne peut être écarté. Des techniques strictes peuvent montrer qu'une entité ou qu'un assemblage est dépourvu de tel ou tel sens particulier (dit «*étroit*»), par exemple n'a pas de sens propositionnel, mais il demeurera toujours pourvu d'un sens, dit «*plus large*» ([6], p. 261), celui auquel se réfère le principe de pansémie, il est saisi «*comme concept*» : d'une entité ou d'un assemblage qu'on rejette on comprend au moins qu'il se distingue de certains autres. En bref, il donne à penser⁸. Whitehead lui-même a reconnu, dit Curry, que les combinaisons condamnées par la théorie des types des *Principia Mathematica* ne sont pas «*absolument sans signification*» puisqu'on peut raisonner sur elles ([8], p. 95). En outre, quand une entité est dépourvue d'un certain sens étroit, elle peut avoir un autre sens étroit qu'il faut lui trouver (ce qui n'est pas sensément une proposition peut être sensément autre chose), et de plus il y a des cas de polysémie.

⁸ Dans ce passage et «*tout au long de l'article*», un rapporteur anonyme estime retrouver «*des confusions propres au psychologisme dénoncées par Frege et Wittgenstein*». Nous nous proposons de répondre à cette objection dans un autre article.

- (2) Eu égard à l'acception étroite du mot "sens", les logiques formelles usuelles présupposent que certaines combinaisons d'éléments peuvent être acceptées parce que douées de sens, par exemple $p \rightarrow q$ en calcul des propositions, fx en calcul des prédicats, et que certaines combinaisons d'éléments ne peuvent pas être acceptées parce que dénuées de sens, par exemple xx en calcul des propositions, xx en calcul des prédicats. Mais on accepte $p \rightarrow q$, et non xx , pour des raisons quasi langagières, extérieures à la formalisation, qui représentent une démission de la pensée formelle. Pour s'en assurer, il suffit d'ailleurs d'ajouter que l'opposition langagière à xx convient particulièrement mal puisque l'objet xx n'est pas un «*du*», mais tout objet qui accepte le traitement qu'on donne de xx . Certes, les règles de formation autorisent ou interdisent formellement les combinaisons, mais elles ont été choisies pour le faire. Il n'y a pas de procédure formelle qui impose les règles de formation elles-mêmes. Une règle de formation pourrait permettre xx aussi bien que l'interdire. Il est vrai aussi que certaines combinaisons évitées, par exemple $p \sim$ ou xf ne sont pas rejetées informellement comme des non-sens, mais expriment seulement un choix assez conventionnel pour un autre ordre dans l'écriture, celui de $\sim p$, fx . Il reste qu'une combinaison comme ff par exemple est l'objet d'une censure informelle à portée théorique. D'une manière plus générale, on peut remarquer qu'il y a beaucoup de données qui sont transportées informellement (et même clandestinement) de la pensée commune à la logique mathématique.
- (3) Il faut donc donner des critères formels de rejet et d'acceptation de telles ou telles combinaisons. Cela impose qu'on élabore d'abord une théorie générale des combinaisons où toutes les combinaisons seront admises, en terrain neutre, ensuite qu'on élabore des techniques qui discrimineront entre les combinaisons (par exemple qui diront propositionnelle la combinaison $p \rightarrow q$, et non propositionnelle ou de tel autre type la combinaison xx). Comme c'est le cas dans toute formalisation, on pourra retrouver ou non des distinctions faites intuitivement. Cette méthodologie présuppose *le principe de formalisation complète*. Un système est dit «*complètement formalisé*»
- S'il ne se réfère qu'à des objets entre lesquels aucune distinction catégorielle n'est faite ou supposée.
 - Dans le cas où son alphabet comporte d'autres éléments que des objets (par exemple des éléments «*opérations*»), aucune distinction catégorielle n'est faite ou supposée. On peut parler d'une opération n -aire, mais toute opération n -aire s'appliquera à tout n -uple d'objets.
 - Dans le cas où son alphabet comporte une troisième ou une n -ième sorte d'éléments, on adapte ces mesures à chacune (par exemple, s'il y a des variables elles ont toutes le même domaine de variation ; s'il y a des énoncés portant sur des n -uples d'objets, il y a un énoncé sur n'importe lequel de ces n -uples ; s'il y a des règles de déduction, elles ne portent pas sur des éléments d'une certaine sorte (contrairement aux règles de substitution), etc.
Certes, parler d'objets, puis d'éléments d'une deuxième sorte, etc. suppose des distinctions informelles, mais,
 - Pour que les problèmes logiques puissent se formuler, il semble inévitable

d'accepter autre chose que des éléments «objets» (notamment des opérations et des énoncés). Ces autres éléments ne seraient pas admissibles si leur distinction était une solution intuitive à un problème. Ils sont requis parce qu'ils constituent la problématique elle-même.

- On cherche à réduire le nombre de ces sortes d'éléments autant qu'il est possible, et à l'intérieur même d'une sorte on cherche à réduire à l'unité les éléments requis (un seul objet [6], p. 31, une seule opération, etc.). Par économie des moyens, mais aussi, peut-être, parce qu'il est douteux qu'il puisse y avoir deux éléments d'une même sorte sans qu'on les suppose, au fond, (même par ailleurs) de deux sortes.
 - On peut admettre différents degrés de satisfaction de la formalisation complète. On s'écarte de la norme d'autant plus qu'on admet plus de sortes d'éléments. Sous sa forme idéale (une seule sorte d'éléments, à savoir des objets) elle est très certainement hors d'atteinte. Même sous des formes plus lâches, elle est d'obtention difficile.
 - À l'inverse, pourtant, la formalisation complète pourra se voir renforcée par le développement des sciences. D'autre part, il y a des normes formelles qu'elle n'exprime pas (elle ne se prononce pas sur les exigences de constructivité, de vérifonctionnalité, etc.).
- (4) L'interdiction de présupposer des types d'être impose ce qu'on peut nommer *le principe de neutralité ontologique* : on ne doit pas distinguer des réalités destinées à être des «opérateurs absolus» (des agents) et des réalités destinées à être des «opérandes absolus» (des patients). Supposer qu'il y a, même au plan mathématique, des objets inertes qui en toutes circonstances subiraient l'action (comme p , q en calcul des propositions), face à des opérateurs qui en toutes circonstances accompliraient l'action (comme $\sim$, $\rightarrow$) serait une transposition d'une donnée intuitive. Ajoutons qu'en outre, cette donnée serait discutable : le clou subit le marteau et s'enfonce, mais il n'est passif que dans certaines circonstances. Il lui arrive aussi de détériorer le marteau, qui dès lors subit l'action. C'est «En situation» qu'il faut juger quel est l'objet qui opère et quel est l'objet sur lequel il est opéré. Il est donc dangereux de parler des atomes en les qualifiant comme le fait J.-P. Desclés «*atomes* (opérateurs et opérandes absolus)» ([11], p. 138). Il faut au contraire se donner des termes qui seront selon leur situation et non selon quelque «nature» opérateurs, opérandes ou même non affectés. Cette situation s'exprimera par la position d'un symbole donné dans une chaîne de symboles. Même une indéterminée n'est pas un opérande absolu, bien que son indétermination interdise d'en donner l'effet. On nommera *principe de réversibilité* l'exigence que tout objet qui est opérateur puisse être opérande et que tout objet qui est opérande puisse être opérateur.
- (5) La théorie générale des combinaisons ne peut pas être simplement celle de l'analyse combinatoire car l'objectif est ici logique. On s'intéresse à la pensée, ce qui introduit des contraintes spécifiques⁹. Ici aussi il s'agit d'une exigence informelle requise pour constituer la problématique. Or, la pensée, sous sa forme la plus manifeste,

⁹ En 1980, encore, Curry rappelle qu'« λ a défini la logique mathématique comme l'étude des structures mathématiques ayant quelque connexion avec l'analyse et la critique de la pensée», ([8], p. 97).

c'est-à-dire discursive, s'énonce en phrases, enchaîne de manière liée des items provenant d'un lexique. D'autre part, comme il faut lier dès qu'on fait intervenir deux unités, soit x_1, x_2 , la liaison fondamentale est binaire. En outre, elle se révélera suffisante. $(x_1 \square x_2)$ – ou en bref (x_1x_2) – peut indiquer que x_1, x_2 sont liés. Si cette liaison exprime que l'une des unités affecte l'autre (par exemple *est mortel* affecte *Socrate*), cela peut être avec ou sans réciprocité, et il faut donc qu'on puisse écrire $(x_1x_2), (x_2x_1)$, comme distincts, si on décide conventionnellement d'écrire à gauche celui qui affecte. On s'accordera aussi la possibilité de $(x_1x_1), (x_2x_2)$, pour ne rien exclure d'emblée comme impensable. Lorsqu'il y a trois unités et qu'on s'adresse au schème général de la pensée, on n'a pas de raison d'imposer $((x_1x_2)x_3)$, et on accordera aussi $(x_1(x_2x_3))$, aux conventions près sur les omissions de parenthèses. On ajoutera seulement, et par commodité, que dans le premier cas les parenthèses ne seront pas mises. On suppose dans cette approche générale (mais temporaire) que tous les items sont des catégorèmes (sont pourvus d'un sens plein), car cela est vrai pour le sens dans l'acception «large» du mot, celle du principe de pansémie.

Autrement dit, comme l'exprime clairement D. Dubarle : «On y suppose que l'esprit est en présence d'objets absolument quelconques et indéterminés, et que penser consiste à leur appliquer mentalement quelque chose, n'importe quoi d'ailleurs. Il n'y a même pas lieu en principe de distinguer les symboles affectés aux objets et à ce que la pensée entend leur appliquer» ([13], p. 67).

Dans le cas le plus simple, un exemple de liaison effectuée par $\square$ est celle par laquelle la lettre *o* forme avec la lettre *r* un tout (qui est un mot) dans la conjonction *or*. En somme, (or) a pour but de distinguer *o, r* du mot *or*. C'est bien ce qu'on suppose d'ailleurs quand on emploie les combinateurs pour engendrer un mot de son étymon, par exemple *or* de *ore* par le combinateur **BK**, **BKore** $\square$ **K**(*or*)*e* $\square$ *or*.

C'est aussi la manière dont les deux premiers items des trois configurations

•• • □□ □ •• □

se groupent dans chacune par rapport au reste en formant du tout une Gestalt (visuelle, rythmique, etc.), ou la manière dont $\bullet \square$ se groupent dans $\bullet\bullet \square$, en formant donc $\bullet(\bullet\square)$ plutôt que $(\bullet\bullet)\square$, au nom de quelque norme, par exemple en privilégiant un autre contraste. Cela peut être la manière dont une fonction unaire porte sur un objet. Il faut aller jusqu'à voir dans ' $\square$ ' l'opération d'associer une chemise avec une cravate qui lui convient. C'est pourquoi Curry dit de (xy) que c'est «un mode de combinaison, le seul de la théorie. Deux entités x et y se combinent pour donner une troisième entité appelée l'application de x à y et notée par (xy) » ([2], p. 370). La troisième entité dans nos exemples est respectivement le mot, la prégnance perceptive, la valeur d'une fonction, l'accord esthétique. Le sens le plus général de (xy) est le suivant : «l'objet x formant avec l'objet y une totalité dans laquelle x , selon sa nature, affecte y ».

Ce que dirait (xy) réduit au simple parenthésage, c'est que x et y sont pensés comme un tout. Ce qu'ajoute l'opérateur $\square$ (généralement sous-entendu), c'est que cette association est supposée produire un effet. Cet effet peut rester indéterminé. Il

l'est notamment si on s'adresse à des entités x, y dont l'une au moins n'est pas déterminée. En revanche, l'effet de $\square$ est toujours le même, associer et soumettre y à l'action éventuelle de x . Les objets sont quelconques, mais l'opération qui les connecte (et qui est unique dans la théorie) est à la fois générale – « affecter en quelque chose » – et précise. Cela est requis d'elle pour pouvoir être appropriée à n'importe quel objet qu'on lui soumettra.

Non seulement, par conséquent, la LC est-elle une logique de l'objet quelconque, mais d'un même mouvement elle est une logique de l'opération quelconque, en ce sens au moins qu'à son opération de liaison peut correspondre n'importe quel nexus.

Il ne faut pas croire qu'il y a des objets quelconques, les indéterminées, et des objets déterminés, les combinateurs, et déterminés à être des opérateurs par leurs règles de réduction, en constituant le « noyau » de la logique combinatoire. En réalité, les combinateurs eux-mêmes se comportent comme des indéterminées quand ils sont opérands et les indéterminées peuvent bien posséder en droit un effet combinatoire, mais qu'on suppose ignorer.

Quand (xy) n'a pas d'interprétation, dit même Curry, « le lecteur est dérangé par ce manque, il peut en inventer une arbitrairement, e. g. (xy) peut alors être égal à $x \square y$ ([2], p. 370). On remarquera que cela veut dire que x appliqué à y n'a pas d'effet : x peut parfaitement s'appliquer à y sans qu'il y ait d'effet (au moins au plan qui est celui de l'analyse), ou plutôt sans qu'il y ait de *deuxième* effet puisqu'il y a toujours celui d'avoir totalisé et soumis y à x . C'est notamment cela qui permet de dire qu'on a pour tout x , pour tout y , $(x \square y)$. Ce serait donc une grande erreur de comprendre ce qu'on nommera le *principe de libre application* comme affirmant quelque influence de tout objet sur tout autre.

On voit donc pourquoi « l'analyse de la pensée » impose que les combinaisons soient applicatives (et non celles de l'analyse combinatoire), correspondant à toutes les manières de lier deux par deux des objets, comme le prévoient les règles de formation.

Quand on décrit la LC comme une logique de l'opérateur, on a tendance à minorer l'adjectif « combinatoire ». J.-P. Desclés écrit par exemple : « Il eût été certainement préférable d'appeler « logique des combinateurs » ce qui a été appelé par Curry « logique combinatoire », pour éviter ainsi un certain nombre de malentendus. En effet, la logique des combinateurs ne vise pas à engendrer toutes les « possibilités combinatoires » d'un jeu de symboles mais plutôt à étudier les processus opératoires conçus comme des organisations d'opérateurs plus ou moins complexes » ([11], p. 16). Bien que nous discutons la fin de cette phrase, c'est vrai que la LC ne vise pas à engendrer toutes les « possibilités combinatoires » d'un jeu de symboles, et en ce sens elle est bien distincte de l'analyse combinatoire, mais c'est seulement parce qu'elle n'a pas à s'intéresser à tous les ensembles qu'on peut former à partir de m objets. Le programme reste bien combinatoire, cependant, car il s'intéresse en fait aux combinaisons archétypiques de deux objets (via $\square$), une *combinatio* étant une association d'objets deux à deux.

- (6) Puisqu'on demande de formaliser en laissant le moins possible de place au jugé intuitif, on ne va pas se donner les combinaisons applicatives en prélevant et en appliquant. Ce sera le rôle des combinateurs de les obtenir. Bien sûr, on ne peut pas éviter que certaines combinaisons et certains combinateurs (les primitifs) ne soient fournis, mais tout le reste s'obtient formellement. D'autre part un combinateur doit bien combiner, au sens intuitif, ce qu'il affecte. Or, combiner c'est modifier une séquence d'objets (ou, par extension, la conserver) sans affecter la nature des objets eux-mêmes, ni *a fortiori* ajouter d'objets (alors qu'on peut, évidemment, en éliminer). Si la nature de quelque objet change, cela doit *résulter* de la combinaison. S'il fait autre chose, le mot "combinateur" qu'on emploie devient conventionnel et on ne peut plus juger d'un programme qui soit, sans jeu de mots, combinatoire. Certes, (xy) se présente comme une troisième entité par rapport à x et à y , c'est-à-dire comme l'ajout d'un objet, mais si on restitue tous les *combinandi* qui sont en réalité $x, y, \square$, on voit qu'on a seulement des modifications de cette séquence puisqu'on obtient x ou y (qui éliminent $\square$ et l'autre objet, ou, si l'on préfère, qui n'en font pas usage), et $(x \square y)$.

C'est pourquoi la sémantique d'un combinateur est interne, alors que la sémantique d'un foncteur propositionnel classique, par exemple, est externe. En effet, un combinateur X affecte des éléments nommés $x_1, x_2 \dots$ dans la métalangue et formés d'atomes qui figurent syntaxiquement dans l'alphabet, en obtenant une combinaison de ces éléments, tandis qu'un foncteur propositionnel comme P , l'implication (s'il est tout à fait classique) n'affecte pas directement les opérandes qui lui sont rapportés dans la syntaxe, p, q , mais des éléments 1, 0 qui sont associés à p, q et qui n'appartiennent pas à la même langue (ne sont pas dans l'alphabet à côté de p, q), et en obtenant l'un de ces éléments (une combinaison par extension, au sens où un atome est dit être une combinaison)¹⁰.

- (7) Toutefois, si les combinateurs étaient donnés comme des opérateurs (des opérateurs absolus), on contreviendrait au principe de neutralité ontologique. En eux-mêmes, ils ne peuvent être dits très généralement que des objets. Certes, il est commode de dire

¹⁰ Nous avons exprimé l'effet des combinateurs par des ' $\square$ ' et non par des '=' (contrairement à Schönfinkel qui n'emploie d'ailleurs jamais le mot "combinateur", mais seulement l'expression «fonction particulière» parce qu'à nos yeux l'entreprise n'est plus strictement combinatoire si elle accepte par exemple $Cx_1x_2x_3 = x_1x_3x_2$. Cette expression permet en effet d'obtenir à partir de l'expression $x_1x_3x_2$ l'expression $Cx_1x_2x_3$ qui n'est pas une combinaison de la précédente (au sens où l'impose (6), une restructuration sans adjonction). On dira peut-être que $Cx_1x_2x_3$ ajoute à $x_1x_3x_2$ un signe, à savoir C , plutôt qu'une entité (Schönfinkel dit d'ailleurs que les deux membres d'une égalité «expriment la même chose» ([29], § 3, p. 16)). Toutefois, $Cx_1x_2x_3$ serait alors une simple manière indirecte de dire « $\square_1x_3x_2$ » par la périphrase « $\square_1x_2x_3$ à l'inversion près des deux derniers éléments», alors qu'on entend faire servir les combinateurs à éviter ces moyens jugés plus rhétoriques que logiques que sont les variables. On dirait simplement ce qu'il faut faire au lieu de le faire ! Pour que l'entreprise réussisse, ne soit pas de la «préciosité» mathématique, il importe donc que les combinateurs soient des entités, des constantes. Dans ce cas, $Cx_1x_2x_3$ ne combine pas $x_1x_3x_2$. En revanche, on peut adjoindre '=' à ' $\square$ ' (dès LC_1) si on lui fait exprimer que deux combinateurs forment sémantiquement un seul objet parce qu'ils ont les mêmes résultantes sur les mêmes suites. On ajoute aux règles de formation : si x_1 et x_2 sont des objets, $x_1 = x_2$ est un énoncé. Ainsi, $I = WK$, car $Ix_1 \dots \square x_1 \dots, WKx_1 \dots \square Kx_1x_1 \dots \square x_1 \dots$, avec $(\square)$.

que **I** par exemple est un opérateur qui transforme un opérande comme a_1 en une résultante. Et même que $\square$ est l'opération d'appliquer un opérateur, par exemple **I** à un opérande, par exemple a_1 . Pourtant, il est plus exact de dire que le seul opérateur est $\square$ et qu'il active, qu'il rend opérant une entité inerte en tant que telle comme **I**, lorsque celle-ci figure comme le premier opérande de $\square$. Bien qu'on dise souvent qu'un combinateur X est unaire, il est à vrai dire 0-aire. $\mathbf{K}xy$, par exemple, n'est pas $\mathbf{K}(x, y)$, mais pas non plus $(\mathbf{K}x)y$. En toute rigueur, c'est $(\mathbf{K} \square x) \square y$.

Il s'ensuit que c'est par manière de parler qu'on dit opérateurs **I**, **K**, etc., voire un élément quelconque à gauche d'un autre, ils sont toujours opérandes, en fait, premier opérande de $\square$ et rendu par là opérant, ou deuxième opérande de $\square$ et laissé inerte.

Plutôt que de parler d'une logique de l'opérateur, mieux vaudrait dire à cet égard que la LC est une logique de l'opérande car tous les objets y figurent comme opérandes de l'unique opérateur $\square$. L'unicité de l'opération (ou plutôt de l'opérateur) s'explique par le fait que tout langage qui dispose d'autres opérations peut être réexprimé de telle sorte qu'il n'ait que cette seule opération (il devient alors ce qu'on nomme un langage applicatif).

Soit en effet un langage disposant des opérations $O'_1, O'_2, \dots$, où O'_i comporte n_i opérandes. Soit $O_1, O_2, \dots$, une série de nouvelles marques. Soit $\square$ l'opération binaire en question (et ici en notation préfixée par commodité d'expression).

Remplaçons toute occurrence de l'expression « $O'_i a_1 a_2 \dots a_{n_i}$ » par « $\square_{n_i} O_i a_1 a_2 \dots a_{n_i}$ » où « $\square_{n_i}$ » représente « $\square$ » répété n_i fois. Le nouveau langage a seulement l'opération $\square$. D'après [4], p. 35. Soit par exemple « $\square_{n_3} a_1 a_2 a_3$ », il vient « $\square \square \square O_3 a_1 a_2 a_3$ ». Notons que le contenu de « $\square_{n_i} a_1 a_2 \dots a_{n_i}$ » est conservé dans « $\square_{n_i} O_i a_1 a_2 \dots a_{n_i}$ ». On ne peut pas dire, évidemment, que la LC exprime avec « $\square$ » ce que dirait O'_i . Elle permet de traiter comme un opérande ce qui était originellement un opérateur.

Si donc les combinateurs sont dits des objets, c'est qu'ils ne sont pas des opérateurs, et pas non plus destinés à n'être que des opérandes. Il faut dès lors une dénomination aussi large que possible, qui est bien fournie par le mot « objet » et plus encore par le néologisme *ob*. Il n'y a pas d'inconvénient à dire que **I**, **K**, etc. sont des opérateurs si on y voit seulement une ellipse pour dire « objets rendus opérateurs par leur position à gauche d'un $\square$ ». Quand nous voulons le rappeler, nous dirons *opérateurs délégués*. Ils sont des opérateurs « seconds » comme il y a des « causes secondes » en métaphysique classique.

Après les deux types de système présentés, et puisqu'il faut des techniques qui discriminent entre les combinaisons, une troisième étape sera fournie par la donnée d'un système de LC illative. Un objet nouveau, noté F (qui n'est ni un combinateur ni une indéterminée), s'ajoute avec la mission de répartir en différentes catégories les autres objets. Il en sera question dans la dernière partie.

- (8) À un plan plus mathématique, l'usage du mot « objet » est à comprendre aussi à partir de la manière dont la LC donne la primauté au concept de fonction (d'ailleurs revu et modifié) sur le concept d'ensemble.

En théorie des ensembles, une fonction est un ensemble, elle est un ensemble de couples, par exemple $\{(0, 1), (1, 2), (2, 3), (3, 4), \dots\}$. Le sens intuitif de ce qu'on faisait (à savoir, «Ajouter +1 à un entier donné» ou «Passer au successeur immédiat») est repris et dépassé dans une position extensionnelle ; un acte (qu'il soit effectué par le sujet lui-même ou par un être mathématique fondé de pouvoir) disparaît derrière son effet. En LC une fonction (ou plutôt un foncteur) est un producteur d'objet par action sur un autre objet. Comme son précurseur intuitif, «+1» transforme un entier en celui qui le suit immédiatement. On a recours à des relations causales, là où la théorie des ensembles, plus phénoméniste, ne retient que des concomitances. Certes

- Il n'est pas du tout scandaleux qu'une donnée intuitive se modifie par formalisation. En même temps, toute modification n'est pas appropriée.
- Il est incontestable didactiquement qu'on fait admettre plus facilement la position causale que la position ensembliste, mais cela ne prouve pas la supériorité de la première car la paresse d'esprit explique aussi bien le fait que la pertinence théorique, le sujet retrouvant ses habitudes.
- Au plan théorique, d'ailleurs, l'approche ensembliste (bien adaptée à l'extensionnalisme) a porté ses fruits, et d'autre part il faut voir que la LC, sur ce point, en revient à une position ancienne que l'ensembliste avait dépassée. Il est connu (depuis Malebranche et Hume, au moins) que le concept de causalité n'a pas la clarté de celui de concomitance. On peut donc refuser d'engager une mathématique derrière la notion obscure (métaphysique, si l'on préfère) de causalité, et se borner à parler des concomitances constatées, quitte à se priver d'un prolongement que certains admettraient.

On peut défendre cependant la position de la LC de trois manières au moins :

- Il n'est pas nécessaire d'avoir réfuté une voie (ici, l'ensembliste) pour en essayer une autre (ici, le fonctionnalisme).
- On peut estimer qu'on ne suppose que la notion commune et fruste de causalité employée par tout le monde (ici comme obtention par une procédure de calcul), à un plan qui est donc aussi phénoméniste.
- En évitant la notion de causalité, on ne neutraliserait pas prudemment une philosophie aventureuse. Tout se passerait comme si on rejetait la notion de causalité, ce qui n'est pas moins aventureux. La seule façon de ne pas embrigader le formalisme derrière une philosophie controversable, c'est d'élaborer autant de formalismes qu'il y a de philosophies, et avec l'espoir d'être conduit à juger celles-ci à leurs fruits.

Cela a pour conséquence que dans l'ensembliste une fonction est une certaine correspondance, alors que dans la LC, comme on en fait une chose qui sur une chose produit une troisième chose, il est naturel de traiter une fonction f comme on traite habituellement son argument a , c'est-à-dire comme un objet. Une fonction est un objet, en somme, parce qu'elle est comprise comme une machine, ou au moins comme un mécanisme.

La LC va jusqu'à retourner la situation. Une fonction n'est plus un ensemble particulier. « λ est plus simple, dit Curry, de définir un ensemble en termes de fonction que vice versa λ ([8], p. 96).

Dans une conception ensembliste, il n'est pas possible de détacher la fonction de son domaine (d'arguments) et de son contre-domaine (de valeurs) puisqu'elle se définit par leur association. Notamment cela n'aurait pas de sens qu'une fonction f ait f pour argument ou f pour valeur. En revanche, en LC le domaine d'une fonction est ce à quoi la fonction peut convenir, et le contre-domaine (dans notre terminologie) est ce qu'elle en obtient. Rien n'interdit donc d'appliquer une fonction à elle-même ou d'obtenir f lui-même de f . On nomme souvent « λ opérateur λ », aujourd'hui, une fonction (ou plutôt un foncteur) quand elle est pensée comme indépendante d'un domaine et d'un contre-domaine présupposés. On pourrait dire *fonction-opérateur* face à *fonction-ensemble*.

- Au sens plus particulier, on nommera « λ opérateur λ (opérateur₂) une fonction, quand elle est comprise, à la manière de la logique combinatoire, comme l'entité qui produit une donnée par intervention sur une autre (au lieu d'être une association entre des valeurs d'entrée (ou d'arguments) et des valeurs de sortie, à la manière ensembliste. Ce sens, un peu conventionnel, est assez classique aujourd'hui. « λ opérateur₂ λ indique qu'on traite en soi-même l'entité, en l'autonomisant par rapport à ce qui a pu heuristiquement la susciter. Comme le dit ([20], p. 44 sq), l'essentiel du concept d'opérateur (au sens opérateur₂) est peut-être qu'il n'y a pas de domaine assigné. (Voir aussi [9])¹¹.

En outre, certains combinatoristes au moins ne sont pas loin de l'idée que tous les êtres mathématiques, et non seulement les ensembles, sont des fonctions, au sens où ils prennent le mot, des fonctions-opérateurs. Certains auteurs, dit Rosser, « λ croient que tous les êtres de la mathématique sont des fonctions λ ([28], p. 18), voire ces « λ fonctions particulières λ (comme disait Schönfinkel) que sont les combinateurs.

Nous donnerons quelques exemples de ce qui fait pour nous, à cet égard, le « λ style λ combinatoire.

4. EXEMPLES (CONNUS PUIS NOUVEAUX) DU STYLE COMBINATOIRE

- Avec Church, Kleene et Rosser, on peut assimiler chaque entier 0, 1, 2, ... à un combinateur, respectivement noté **0**, **1**, **2**, ... selon les règles de réduction suivantes :

¹¹ (1) On emploie souvent improprement « λ opération λ (l'action) pour « λ opérateur λ (l'agent), ou un mot qui désigne l'un pour un mot qui désigne l'autre, comme lorsqu'on dit « λ égation λ au lieu de « λ égateur λ pour « λ ». (2) Sur cette assimilation abusive opération / opérateur, il arrive qu'on réintroduise (conventionnellement) une différence en employant « λ opération λ pour désigner certaines entités et « λ opérateur λ pour en désigner d'autres, généralement plus abstraites (sans distinguer entre action et agent). (3) « λ opérateur λ désigne parfois seulement le symbole de l'entité qui opère.

$0xy \sqsupset y$	0 applique zéro fois x à y , il reste y
$1xy \sqsupset xy$	1 applique une fois x à ce que produit 0
$2xy \sqsupset x(xy)$	2 applique une fois x à ce que produit 1

etc. (mais reformulable en combinateurs standards, **0** par exemple est **KI**). Un entier est alors un opérateur qui effectue une certaine transformation. Comme on peut assimiler aussi les opérations sur les entiers à des combinateurs (on montre par exemple que $m \sqsupset n$ s'exprime par **B**, m^n par **CI**), les objets et les opérations, dans cette arithmétique, sont donc des combinateurs. « $\lambda \sqsupset 0$ fait 0λ » s'exprimera par **B10xy** $\sqsupset$ **1(0x)y** $\sqsupset$ **0xy** $\sqsupset$ y , ou par **B(SB(KI))(KI)xy** $\sqsupset$... $\sqsupset$ y , où **S** serait **B(B(BW)C)(BB)** dans L_1 .

- En grammaire applicative, on peut aussi traiter certaines unités comme des opérateurs. De même que « λ on λ » est un opérateur qui sur un opérande propositionnel forme une proposition, de même l'article « λ n λ » peut-il être dit « λ pérateur λ » (il forme un nom à partir d'un nom). G. E. Révész remarque à juste titre « λ ue presque tout peut se représenter de cette manière λ » ([27], p. 40).
- De même, on peut assimiler une valeur de vérité comme 0 ou 1, à un combinateur, respectivement noté **0**, **1**, selon les règles de réduction suivantes :

$$0xy \sqsupset y$$

$$1xy \sqsupset x$$

(**0** est **KI**, **1** est **K**), comme dans la consigne « λ i p est vrai, faire x , si p est faux, faire y ». On n'a plus seulement des fonctions de vérité portant sur des objets, les valeurs de vérité ; les valeurs de vérité elles-mêmes sont devenues des fonctions ¹². À partir de là, nous présenterons l'esquisse de trois traitements, dont nous sommes l'auteur, destinés à faire mieux comprendre ce qu'est une investigation combinatoire (en définissant trois langages : **P**, **1**, **0**, puis λ , **1**, **0**, et enfin **H**, **1**, **0**).

Il y a des systèmes qui disposent à côté des variables de proposition, $p, q, m, \dots$, de constantes de proposition (une proposition toujours vraie et une proposition toujours fausse) et qui notent celles-ci respectivement 1 et 0, au prix, certes, d'un abus d'écriture puisqu'on assimile la proposition et sa valeur, mais avec un intérêt que nous expliciterons. Dans les systèmes classiques de cette forme, 1 et 0, comme $p, q, m, \dots$, sont toujours opérandes, jamais opérateurs. On a par exemple (en langage $\sim, \sqsupset, 1, 0$) des formules comme $p \sqsupset q, p \sqsupset 1, p \sqsupset 0$ (qui définit $\sim p$), $0 \sqsupset \sim 1$, etc.

Mais puisqu'on vient de traiter 1 et 0 comme des combinateurs, on peut se donner un système dans lequel ils seront (comme tous les combinateurs) tantôt opérateurs, tantôt opérandes, et en cela donc de purs objets. En écrivant de manière préfixée (' λ ' devenant **P**, ' $\sim$ ' devenant **N**) et avec parenthésage combinatoire, on a par exemple **Ppq**, **Pp1**, **P0(N1)**, mais aussi **1pq**, **0pq**. On obtient bien des lois dans cette extension, par exemple

¹² **0** et **1** dans ce contexte sont évidemment différents de **0** et **1** dans le contexte précédent.

$P(1pq)(0qp)$, puisqu'on a :

p	q	$1pq$	$0qp$	$P(1pq)(0qp)$
1	1	1	1	1
1	0	1	1	1
0	1	0	0	1
0	0	0	0	1

$P(11p)(0q1)$, qui a la table de $P11 ; 0(P1q)(11p) ; 1(0p1)q$ ¹³.

Les constantes propositionnelles, 1, 0, s'assimilaient aux valeurs 1, 0, parce qu'elles représentaient des expressions qui s'effectuaient en ne conservant que 1 ou 0 (comme Ppp pour 1). De même, les combinateurs **1**, **0** (qui remplacent et étendent, ici, les constantes propositionnelles) s'assimileront aux valeurs 1, 0, parce qu'ils représentent des expressions qui s'effectuent en ne conservant que **1** ou **0**, comme $11p \sqsubseteq 1$ ou $0p1 \sqsubseteq 1$; $0p0 \sqsubseteq 0$, $10p \sqsubseteq 0$. Évidemment, un tel langage étend le calcul qui n'est plus étroitement «propositionnel».

Quant aux variables p, q , etc., opérands par excellence, il est conforme à un programme combinatoire de les éliminer plutôt que de vouloir les transcrire en opérateurs, car on y voit des symboles qui, comme les parenthèses, sont des moyens utilisés pour s'exprimer et non des objets sur lesquels on s'exprimerait. En utilisant les combinateurs standards et selon la méthode habituelle, voir par exemple ([16], p. 54 sq), on peut exprimer le langage $P, 1, 0$ sans mentionner de variables propositionnelles. Ainsi, $1(0p1)q$ sera $W(BC(B^2CB^2))10$ car : $W(BC(B^2CB^2))10pq \sqsubseteq BC(B^2CB^2)110pq \sqsubseteq B^2CB^2101pq \sqsubseteq B^210p1q \sqsubseteq 1(0p1)q$ ¹⁴ (voir aussi la note 24 et son contexte).

1 et **0**, traités comme des foncteurs, figurent nécessairement parmi les 16 connecteurs binaires. Ils correspondent respectivement aux deux éléments, notés ici $\sqsubseteq$ et $\sqsupseteq$ (de manière infixée) et qu'on peut lire «potence gauche», «potence droite» :

p	q	$p \sqsubseteq q$	$P \sqsubseteq q$
1	1	1	1
1	0	1	0
0	1	0	1
0	0	0	0

Cependant, ce qu'il y a de combinatoire dans cette approche est la réversibilité

¹³ Les formules pourraient être engendrées formellement si on se donnait des règles explicites de formation, et certaines apparaîtraient comme des lois à l'évaluation par les tables. Nous nous bornons ici à en exhiber informellement quelques-unes.

¹⁴ Soit X l'un des combinateurs, **I**, **K**, **W**, **C**, **B**. X^n est défini par : $Xx_1x_2\dots x_m \sqsubseteq y_1y_2\dots y_p$, $Xy_1y_2\dots y_p \sqsubseteq z_1z_2\dots z_q$, etc., et en appliquant X n fois. On établit que B^nK (respectivement B^nW) élimine (respectivement duplique) au rang $n + 2$, que B^nC (respectivement B^nB) permute (respectivement compose) les rangs $n + 2, n + 3$.

obtenue des rôles d'opérateurs et d'opérandes. Ce qui semble même un abus d'écriture conduit à ne s'adresser qu'à un seul type d'objet, un objet aussi quelconque que possible, pour éviter en somme les «abus d'intuition», les recours subreptices à des données informelles.

Nous ne chercherons pas, pour l'instant, si tous les foncteurs de vérité ($\sim$, $\vee$, etc.) sont exprimables par des combinateurs, comme $\square$ et $\square$ viennent de l'être, ou comme les opérations arithmétiques l'ont été. Mais nous montrerons qu'on peut les obtenir au moins dans une extension de la LC standard qui ajoute aux combinateurs classiques un discriminateur, à savoir $\square$, tel que

- $\square x_1 x_2 x_3 \square x_1$, si x_3 et x_2 sont sémantiquement le même objet
- $\square x_1 x_2 x_3 \square x_3$, si x_3 et x_2 ne sont pas sémantiquement le même objet.

On peut évidemment définir de différentes manières la propriété pour deux objets morphologiques d'être sémantiquement le même objet, suivant ce qu'on entend distinguer ou confondre. Cela implique des choix ontologiques, donnant cours à plusieurs traitements possibles, même si la définition (dite par abstraction) de «être sémantiquement le même objet que» semble moins embarrassante que celle d'«être objet».

Nous dirons que, dans LC_1 , LC_2 , avec égalité, deux expressions (deux objets morphologiques) x_1, x_2 sont sémantiquement le même objet, ce que nous écrirons $x_1 =_{os} x_2$, ssi l'une au moins des conditions suivantes est remplie :

- [$\square$] Elles sont symboliquement indiscernables.
- [$\square$] Si x_1 et x_2 sont des combinateurs (classiques) $x_1 = x_2$ (au sens de la note 10), comme **I** et **WK**, **BWW** et **BW(BW)**.

On se réfère à la définition classique de l'égalité ($X = Y$ ssi pour toute suite $x_1 \dots$ telle que $Xx_1 \dots$ et $Yx_1 \dots$ peuvent se réduire, $Xx_1 \dots$ et $Yx_1 \dots$ obtiennent la même résultante). Ainsi, **BWW** = **BW(BW)** car **BWW** $xy \dots \square \dots \square xyxy \dots$; **BW(BW)** $xy \dots \square \dots xyxy \dots$. Dès lors, **1** et **0** (**K** et **KI**) ne sont pas égaux puisqu'il y a une suite xy telle que **1** $xy \square x$ et **0** $xy \square y$. **1** et **0** ne sont pas le même objet sémantique (ce que nous écrivons $\mathbf{1} \neq_{os} \mathbf{0}$). Deux expressions x_1 et x_2 peuvent donc se réduire à des expressions symboliquement indiscernables sans être sémantiquement le même objet, comme **K** xy et **I** x , bien que **K** $xy \square x$ et **I** $x \square x$.

Notons qu'on ne posera pas l'égalité d'une expression $Xx_1 \dots x_n$ et de sa résultante $y_1 \dots y_m$, car $Xx_1 \dots x_n = y_1 \dots y_m$, nous ferait sortir d'une logique strictement combinatoire (note 10). Ne sont donc pas le même objet sémantique **I** x et x , **K** xy et x , **W** xy et xyy , etc.

- [$\square$] Si $x_1 =_{os} x_2$, on a aussi, pour x_3, x_4 quelconques : $x_3 x_1 =_{os} x_3 x_2$, $x_1 x_3 =_{os} x_2 x_3$, $x_3 x_1 x_4 =_{os} x_3 x_2 x_4$. Ainsi, comme **I** = $_{os}$ **WK** selon [$\square$], on a **I** $x =_{os}$ **WK** x selon [$\square$], et donc aussi **I** $xy =_{os}$ **WK** xy , à nouveau selon [$\square$], etc.

On a par exemple **BI** $x =_{os}$ **I** x en employant [$\square$] sur **BI** = $_{os}$ **I**, bien que **I** x puisse se réduire, mais non **BI** x . Cela tient à la définition classique de l'égalité que nous

avons rappelée¹⁵. L'égalité $\mathbf{BI} = \mathbf{I}$ s'établit sur $x_1x_2\dots$, mais s'affirme de $\mathbf{BI}$ et de $\mathbf{I}$ comme tels.

[□] Une expression de forme $\square x_1x_2x_3$ (qui est un objet) et un objet x_i sont sémantiquement le même objet ssi $\square x_1x_2x_3 \square x_i$ (cette mesure, rejetée pour un combinateur classique, semble naturelle pour un discriminateur d'objet sémantique dont la raison d'être se borne à désigner l'un (ou tels et tels) des objets sémantiques d'une suite). Il en résulte notamment que dans $\square x_1x_2x_3$ il faut appliquer [□], [□], [□], [□] à x_1, x_2, x_3 , éventuellement à l'aide de (□), avant d'effectuer □.

- Nous ajouterons □ dans l'ensemble E de LC_2 , avec égalité (□ se traite donc morphologiquement comme C ou B).
- Nous nommerons $\text{LC}_{2\square}$ le système ainsi produit¹⁶.
- Nous admettrons les clauses [□], [□], [□], [□] pour $\text{LC}_{2\square}$.

La notion générale de discriminateur a été introduite par J. Kearns [21] qui considère les discriminateurs comme des combinateurs d'un type particulier (des combinateurs «*qui tiennent compte* de leurs arguments», p. 561). Kearns les ajoute dans son système aux combinateurs classiques. Soit, pour le discriminateur Z, dit *de base*

$Zx_1x_2x_3x_4 \square x_2x_4$, si x_4 est le même atome ou la même molécule que x_1

$Zx_1x_2x_3x_4 \square x_3x_4$, si x_4 n'est pas le même atome ou la même molécule que x_1 .

mais il forme ensuite des discriminateurs plus complexes. D'autre part, l'identité ou la non identité exigée est celle de certains *objets morphologiques* (les mêmes symboles ou non). Les discriminateurs que nous introduisons portent sur les *objets sémantiques*, et ils sont seulement de la forme :

$Dx_1x_2\dots x_n \square \mathcal{Q}$, si pour tel x_i et pour tel x_j distincts ($i, j = 1, 2, \dots, n$), $x_i =_{os} x_j$

$Dx_1x_2\dots x_n \square \mathcal{Q}'$, si pour cet x_i et pour cet x_j , $x_i \neq_{os} x_j$,

où $\mathcal{Q}, \mathcal{Q}'$ sont des combinaisons des $x_1, x_2, \dots, x_n$ différentes ; x_i et x_j seront dits *les discriminants* (pour Z et pour D)¹⁷.

¹⁵ On doit évidemment respecter les groupements d'origine. Si on place, par exemple, x_3 devant x_1 et devant x_2 , alors x_1 et x_2 *doivent* être mises entre parenthèses s'il s'agit de molécules.

¹⁶ Nous supposons, cependant, la morphologie de $\text{LC}_2 + \square$ adaptée à ce qui va suivre (notamment, p, q, m , s'ajoutent aux indéterminées).

¹⁷ Avec Z, l'emploi de (□) peut conduire à un autre effet que la réduction sans son emploi. Ainsi, $Z(\mathbf{I}x)yzx \square zx$, par (Z), sans application de (□), alors que $Z(\mathbf{I}x)yzx \square Zxyzx \square yx$, par (Z) après application de (□). Pour éviter de telles issues, Kearns n'admet pas la règle (□) dans son système.

On aurait aussi avec □, $\square x(\mathbf{I}y)y \square y$, par (□), sans application de (□), puisque $(\mathbf{I}y) \neq_{os} y$, alors que $\square x(\mathbf{I}y)y \square \square xyy \square x$ par (□), (I) et (□), selon [□] pour les deux occurrences de y .

Pourtant, ni pour Z ni pour un D, il n'est nécessaire de supprimer (□). Il suffit d'*interdire d'appliquer* (□) *aux discriminants avant l'application des clauses* [□], [□], [□], [□] *qui les concernent* – si elles ne requièrent pas elles-mêmes (□) – ce qui est imposé par la notion même de discriminateur, et non par une mesure *ad hoc* prise pour éviter une incohérence. Certes, on ajoute des restrictions informelles à l'emploi de la règle (□), mais elles correspondent aux restrictions informelles faites sur les règles de réduction ; d'autre part, (□) convient sans restriction aux objets qui ne sont pas les discriminants car le

Alors que dans beaucoup d'usages de la LC, il faut ajouter aux combinateurs certains éléments qui appartiennent au domaine qu'on veut formaliser (par exemple, jusqu'ici, ajouter P à $\mathbf{1}, \mathbf{0}$), Kearns estime que ses discriminateurs permettent d'éviter ou de réduire ces emprunts. Nous allons montrer ici – mais avec des discriminateurs d'objets sémantiques – que c'est le cas pour le calcul des propositions, si on traite $\square$ comme un combinateur, puisque P et N sont définissables par les seuls combinateurs $\square, \mathbf{1}, \mathbf{0}$, soit $\square, \mathbf{K}, \mathbf{KI}$. Rappelons, d'abord, qu'on ne distingue pas la valeur 1 et le combinateur $\mathbf{1}$ qui sont donc un même objet sémantiquement, ni la valeur 0 et le combinateur $\mathbf{0}$, même objet sémantiquement, mais différent du premier. Il vient alors : $Ppq =_{df} \square \mathbf{1}pq$ (soit, en effet, en abrégant $\square \mathbf{1} \mathbf{1} \mathbf{1} \square \mathbf{1}$ (car $x_3 =_{os} x_2$) en $\mathbf{1} \mathbf{1} \mathbf{1} \mathbf{1}$, $\square \mathbf{1} \mathbf{0} \square \mathbf{0}$ (car $x_3 \neq_{os} x_2$) en $\mathbf{0} \mathbf{1} \mathbf{1} \mathbf{0}$, etc.).

Ppq	$\square \mathbf{1}pq$
111	$\mathbf{1} \mathbf{1} \mathbf{1} \mathbf{1}$
010	$\mathbf{0} \mathbf{1} \mathbf{1} \mathbf{0}$
101	$\mathbf{1} \mathbf{1} \mathbf{0} \mathbf{1}$
100	$\mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0}$

et $Np =_{df} Pp\mathbf{0}$, soit $\square \mathbf{1}p\mathbf{0}$.

Puisqu'un combinateur est un objet, on ne doit pas s'étonner que $\square \mathbf{1}pq$ puisse obtenir $\mathbf{1}$ (qui est $\mathbf{K}$). Ce n'est pas plus anormal que la résultante $\mathbf{K}$ obtenue de $\mathbf{IK}$. De là, puisque $\{P, N\}$ forme un ensemble complet fonctionnellement, tout foncteur est exprimable en langage $\square, \mathbf{K}, \mathbf{KI}$ ¹⁸. On a notamment

$$p \square q =_{df} \square \mathbf{1}(\square \mathbf{1}p(\square \mathbf{1}q\mathbf{0}))\mathbf{0}^{19}.$$

En effet,

p	q	$\square \mathbf{1}(\square \mathbf{1}p(\square \mathbf{1}q\mathbf{0}))\mathbf{0}$
1	1	$\mathbf{1} \mathbf{1} \mathbf{0} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0}$
1	0	$\mathbf{0} \mathbf{1} \mathbf{1} \mathbf{1} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0} \mathbf{0}$
0	1	$\mathbf{0} \mathbf{1} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0}$
0	0	$\mathbf{0} \mathbf{1} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0} \mathbf{1} \mathbf{1} \mathbf{0} \mathbf{0} \mathbf{0}$

discriminateur est à leur égard un combinateur classique. On aurait dès lors seulement $Z(\mathbf{I}x)yzx \square zx$, par (Z) , $Z(\mathbf{I}x)yz(\mathbf{I}x) \square y(\mathbf{I}x)$, par (Z) , puis yx par $(\mathbf{I})$ et $(\square)$ (et alors que chez Kearns $y(\mathbf{I}x)$ ne serait pas réductible). $\square x(\mathbf{I}y)y \square y$, par $(\square)$. Pour employer (D) , il faut donc chercher d'abord si les discriminants sont sémantiquement ou non le même objet (ce qui consiste à appliquer $[\square]$, $[\square]$, $[\square]$, $[\square]$), puis, selon le résultat, à appliquer la ligne pertinente de (D) . Ainsi $\square x(\mathbf{WK})\mathbf{I}$, où $\mathbf{I} =_{os} (\mathbf{WK})$, par $[\square]$, donc $\square x(\mathbf{WK})\mathbf{I} \square x$, par $(\square)$. $\square x(\mathbf{WB})\mathbf{I}$, où $\mathbf{I} \neq_{os} (\mathbf{WB})$, par $[\square]$, donc $\square x(\mathbf{WB})\mathbf{I} \square \mathbf{I}$, par $(\square)$.

¹⁸ De même qu'on peut effectuer $\mathbf{B}mnxy$ et vérifier qu'en obtenant $m(nx)y$ on retrouve bien le résultat de $\square mn$, de même peut-on effectuer $\square \mathbf{1}pq$ et vérifier qu'on obtient bien la table de Ppq , mais ici en disant si p et q sont ou non le même objet sémantique. D'autre part, on obtient aussi $Ppq =_{df} \square \mathbf{1}pq$ comme on obtient $\square mn =_{df} \mathbf{B}mnxy$ pour le cas où $m = 1, n = 1$, le cas où $m = 1, n = 0$, etc.

¹⁹ Si on admet des discriminateurs de forme plus générale, par exemple $\square_g x_1 x_2 x_3 x_4 \square x_1$, si x_4, x_3, x_1 sont le même objet sémantique, et $\square_g x_1 x_2 x_3 x_4 \square x_2$, si x_4, x_3, x_1 ne sont pas le même objet sémantique, on a $p \square q =_{df} \square_g \mathbf{1}pq$.

Pour transcrire dans notre langage une formule propositionnelle exprimée en langage $P, 1, 0$ (et les variables usuelles), il suffit de remplacer partout P par $\Box 1$ et d'utiliser le parenthésage combinatoire. Ainsi Ppp est-il $\Box 1pp$; $PpPqp$ est-il $\Box 1p(\Box 1qp)$. On vérifie facilement que chacune de ces formules est une tautologie (d'une manière générale, la composition des formules à transcrire ne conduit pas à utiliser les règles de réduction de $\mathbf{1}, \mathbf{0}$).

Church a démontré que le foncteur ternaire $Tpqm$ qui a la même table que $(p \Box q) (m \Box \sim q)$, à savoir 11100010, forme à lui seul un système fonctionnellement complet (quand on dispose des constantes de proposition 1, 0), [1], p. 129 *sq.* Or, $Tpqm$ signifie «Si q alors p , si non q alors m »²⁰ Il peut donc être rendu par le discriminateur $\Box'$ (dont il est très proche)²¹

$$\begin{aligned} \Box' x_1 x_2 x_3 x_4 &\Box \Box_2, \text{ si } x_3 \Box_{os} \Box_1 \\ \Box' x_1 x_2 x_3 x_4 &\Box \Box_4, \text{ si } x_3 \Box_{os} \Box_1. \end{aligned}$$

On a $Tpqm =_{df} \Box' 1pqm$. Cela fournirait une autre solution à l'expression combinatoire de tous les foncteurs.

Comme on l'a fait pour le langage $P, \mathbf{1}, \mathbf{0}$, on peut exprimer le langage $\Box, \mathbf{K}, \mathbf{KI}$ sans mentionner de variables propositionnelles. Ainsi, Ppp devient $\Box 1pp$, puis $\mathbf{BW}\Box 1$, car $\mathbf{BW}\Box 1p \Box \mathbf{W}(\Box 1)p \Box \Box 1pp$. On pourrait aussi éviter les parenthèses (et pour chaque langage traité) en notant $\Box xy$ ce qu'on a écrit (xy) , ou xy quand (AG) s'appliquait. $\mathbf{W}(\Box 1)p$, par exemple, s'exprimerait $\Box \Box \mathbf{W}\Box 1p$ (voir aussi la note 24 et son contexte).

Certes, il reste le problème de savoir si un discriminateur (de forme D) peut être dit «combinateur» par une extension du mot qui ne soit pas abusive. C'est le cas, à notre avis. S'il a deux règles de réduction, en effet, chacune est cependant de la forme $Dx_1x_2\dots x_n \Box Q$, où Q est une combinaison, au sens habituel, des x_i ²². Le discriminateur combine car il modifie sans ajouter. D'autre part, les x_i sont bien des objets quelconques, on demande seulement si deux symboles (qui évidemment ont toujours été eux-mêmes des graphismes précis) désignent un même objet quelconque ou non. Remarquons que le discriminateur n'effectue pas lui-même un choix (sinon on pourrait hésiter à dire qu'il se borne à combiner). Il obtient une réduction différente selon un choix effectué par ailleurs. Il est vrai que les clauses conditionnelles endommagent la rigueur formelle. Dans le cas où on jugerait perdu l'essentiel de la notion de combinateur, on remarquera au moins qu'ajouter le discriminateur $\Box$ aux combinateurs $\mathbf{1}, \mathbf{0}$ (soit $\mathbf{K}, \mathbf{KI}$) est beaucoup plus proche de l'attitude combinatoire (pour les raisons données) que d'ajouter le foncteur P aux combinateurs $\mathbf{1}, \mathbf{0}$.

On pourrait nous objecter que les discriminateurs n'ont pas besoin d'être des discriminateurs d'objets sémantiques et que $\Box$ jouerait le même rôle dans les expressions

²⁰ Dans sa première règle de réduction (sans sa condition) $\Box$ est d'ailleurs $\mathbf{B}(\mathbf{BK})\mathbf{K}$, car $\mathbf{B}(\mathbf{BK})\mathbf{K}x_1x_2x_3 \Box \mathbf{BK}(\mathbf{K}x_1)x_2x_3 \Box \mathbf{K}(\mathbf{K}x_1x_2)x_3 \Box \mathbf{K}x_1x_2 \Box x_1$, et dans la seconde (sans sa condition) $\Box$ est $\mathbf{BK}(\mathbf{KI})$, car $\mathbf{BK}(\mathbf{KI})x_1x_2x_3 \Box \mathbf{K}(\mathbf{KI}x_1)x_2x_3 \Box \mathbf{KI}x_1x_3 \Box \mathbf{I}x_3 \Box x_3$.

auxquelles nous le destinons ²¹ s'il était un discriminateur (qui serait plus simple) d'objet morphologique, et en effet puisque nous ne supposons que la différence entre **1** et **0**, qui est à la fois morphologique au sens de Kearns et sémantique à notre sens. Toutefois, il nous semble important, en logique combinatoire, et pour bien la comprendre, de travailler dans le cadre plus général des objets sémantiques. En effet :

- Dans une logique standard, les constantes (de relation ou d'opération), qui se trouvent être ici des connecteurs, portent sur des supports spécialisés (les variables ou des expressions qui comprennent des variables), jamais sur des constantes (de relation ou d'opération). Ainsi, NNp se lit $N(Np)$, comme s'il y avait q à la place de Np , et non pas $(NN)p$.
- En LC, les constantes portent toujours sur des constantes. On peut effectuer **KWCKB**, par exemple : **KWCKB** $\square$ **WKB** $\square$ **KBB** $\square$ **B**, et s'en tenir à la dernière constante obtenue puisque c'est, à ce titre, un véritable objet mathématique (et même si elle était moléculaire). On peut aussi ajouter des variables, mais elles tiennent lieu de n'importe quelle constante (ou suite de constantes) sur laquelle on voudrait poursuivre l'effectuation (c'est-à-dire utiliser **KWCKB**), **Bxyz** $\square$ $x(yz)$, et donc **KWCKBxyz** $\square$ **WKBxyz** $\square$ **KBBxyz** $\square$ **Bxyz** $\square$ $x(yz)$.

On a, d'une part, en logique usuelle, des variables nécessaires pour *obtenir* l'effectuation, et d'autre part, en logique combinatoire, des variables pour *prolonger*, si on le veut, l'effectuation (et sur des constantes non spécifiées). Les expressions combinatoires proprement dites sont donc des séquences de constantes.

Dès lors, puisqu'une constante a pour but de représenter directement un objet mathématique, il devient capital de savoir si elle y parvient, si, au-delà de leur diversité symbolique, deux expressions représentent bien des objets mathématiques distincts, comme en droit elles y prétendent. Même en logique usuelle où il n'y a pas de séquences de constantes – au sens indiqué – on n'accorde pas que deux constantes symboliquement distinctes (comme **N** et **P**) puissent de plein droit représenter des objets mathématiquement distincts.

Accepter que deux expressions symboliquement distinctes puissent toujours appréhender des objets mathématiquement distincts, comme le suppose Kearns (par exemple pour **WK** et **I**), c'est transposer en LC une mesure qui se comprend pour les variables-supports et pour les expressions où celles-ci interviennent, dans les logiques qui en sont pourvues, mais qui manque ce qui est propre à la LC (sauf, toutefois, si on entend rendre celle-ci fortement intensionnelle, en rendant donc sémantiquement distinct tout ce qui est morphologiquement distinct), position admissible quand l'objectif est clairement annoncé) ²².

²¹ Mais non, évidemment, dans le cadre plus général d'emploi de $\square$, $\square$, $\square$, $\square$.

²² Kearns déclare par ailleurs « $\square$ ne comprends pas pourquoi l'extensionnalité [qui admet **WK** = **I**, par exemple] est désirable, si ce n'est que les systèmes extensionnels sont plus faciles à étudier que les systèmes non extensionnels» ([22], p. 43), mais c'est alors un problème essentiel de montrer qu'on peut encore discriminer sémantiquement **WK** de **I**, par exemple. Bref, Kearns traite de **WK** et de **I** comme aussi

Précisons donc la manière dont la logique combinatoire est susceptible d'intervenir en ces matières :

- Le premier niveau d'approche combinatoire est fourni par l'adjonction des combinateurs aux constantes ordinaires du domaine traité, disons $c_1, c_2, \dots$ (par exemple les foncteurs du calcul propositionnel), et principalement pour exprimer sans variables les formalismes originels.
- Un deuxième niveau possible d'approche combinatoire est fourni par le recours à des opérateurs qui puissent encore être dits "combinateurs", mais en un sens élargi du mot (par exemple à des discriminateurs), qui permettent d'éviter la nécessité des constantes $c_1, c_2, \dots$ parce qu'ils ont l'effet de celles-ci.
- Un troisième niveau d'approche combinatoire, le plus ambitieux, est d'obtenir au moyen des seuls combinateurs classiques (des combinateurs au sens le plus strict du mot) ce qu'obtiennent $c_1, c_2, \dots$

Dans les trois cas, on cherche principalement à réduire le nombre des entités nécessaires (par économie des moyens et pour élaborer l'ontologie), mais dans les deux derniers en recourant seulement à celles des constantes auxquelles on attribue la plus grande intelligibilité – les combinateurs classiques au premier chef, ou au moins des combinateurs par extension – car éliminer, dupliquer, permuter, etc. expriment des données simples, des actes élémentaires, jugés beaucoup moins énigmatiques que le mode d'existence de ce que désignent $\sim, \vee, \square$ par exemple. C'est là ce qui fait, à nos yeux, bien au-delà du rejet des variables, le véritable projet de la logique combinatoire.

Nous allons présenter ici notre contribution personnelle à l'objectif du troisième niveau, dans un cas particulier : employer les combinateurs classiques à la place des foncteurs du calcul propositionnel²³.

Le problème de trouver un combinateur qui puisse remplacer un foncteur peut se formuler ainsi :

distincts que le sont **WK** et **B**, donc sans utiliser les propriétés **WK** = **I**, **WK** ≠ **B** (dont nous tenons compte). Il importerait à une position intensionnelle de dire à quel autre égard on peut admettre **WK** ≠ **I**.

²³ Le traitement entièrement combinatoire qui va être donné ne rend pas caduc, à tous égards, le traitement par les discriminateurs car nous entendons illustrer au plan de l'analyse épistémologique chacun des trois niveaux d'approche qui ont été distingués. Nous comprenons, cependant, qu'un rapporteur anonyme ait estimé « douteuse » l'égalité sémantique. Nous répondrons que la formalisation n'intervient pas seulement quand un concept est sûr (à supposer, d'ailleurs, qu'il puisse l'être avant sa formalisation), mais pour chercher s'il l'est (au vu des difficultés, des moyens exigés, des conséquences, etc.) de son appréhension stricte. La formalisation peut prendre les devants sur une analyse discursive achevée, et c'est même l'un de ses mérites (qu'elle soit ou non combinatoire). Nous admettons facilement que l'égalité sémantique reste à explorer. Nous avons dit en outre qu'il y a plusieurs options possibles à son égard, comme d'ailleurs plusieurs notions de l'égalité en LC. Nous nous adressons, derrière la diversité des symboles, **WK**, **I**, et derrière la diversité des procédures (dupliquer puis éliminer, ou reprendre à l'identique) à l'invariance d'un objet, comme on le fait après tout pour 3 et III, 2 + 1 et $\sqrt{9}$. Nous n'ignorons pas les critiques que Wittgenstein oppose à une position semblable de Frege (en bref : à chaque objectivation, un autre objet).

Remplacer un foncteur F tel que $Fp_1p_2\dots p_n$ par un combinateur X , c'est exhiber un combinateur X (qui peut être moléculaire) tel que

$$Xx_1x_2\dots x_n \sqsubseteq x_i$$

et tel que pour chaque assignation de valeurs

$$v_1v_2\dots v_n \text{ (identiques ou différentes) à}$$

$$p_1p_2\dots p_n,$$

si $Fv_1v_2\dots v_n$ prend la valeur v_i alors

$$Xv_1v_2\dots v_n$$

prend la valeur v_i .

- v_i est nécessairement l'un des $x_1x_2\dots x_n$ puisque X est un combinateur (et donc v_i est l'un des $v_1v_2\dots v_n$).
- L'application de X à $v_1v_2\dots v_n$ suppose que $v_1v_2\dots v_n$ soient des objets du système combinatoire. Il suffit de les placer dans l'alphabet de celui-ci.
- Un combinateur possédant une sémantique interne, comme on l'a vu, il remplacera un foncteur si le foncteur est lui-même appréhendé comme transformant une certaine suite d'objets en l'un des objets de cette suite. Il faut donc renvoyer le foncteur à la manière dont il traite les valeurs de vérité. On a par exemple $v(P10) = 0$.
- Cependant, un des problèmes pour exhiber le X vient de ce que $v_1v_2\dots v_n$ ne prend pas nécessairement une valeur v_i qui soit l'une des $v_1v_2\dots v_n$. On a par exemple $v(P00) \not\sqsubseteq 1$.

Considérons pourtant le combinateur H tel que $Hxyz \sqsubseteq yzx$ et les combinateurs **1, 0**. **H1** remplace P puisqu'on a $H1pq \sqsubseteq pq1$ et donc

$$\text{pour } v(p) = 1, v(q) = 1 \sqsubseteq H111 \sqsubseteq 111 \sqsubseteq 1$$

$$\text{pour } v(p) = 1, v(q) = 0 \sqsubseteq H110 \sqsubseteq 101 \sqsubseteq 0$$

$$\text{pour } v(p) = 0, v(q) = 1 \sqsubseteq H101 \sqsubseteq 011 \sqsubseteq 1$$

$$\text{pour } v(p) = 0, v(q) = 0 \sqsubseteq H100 \sqsubseteq 001 \sqsubseteq 1$$

D'autre part, on peut définir la négation. Comme $Np =_{\text{df}} Pp0$, il vient $\sqsubseteq H1p0$. En effet,

$$H1p0 \sqsubseteq p01$$

$$\text{pour } v(p) = 1 \sqsubseteq H110 \sqsubseteq 101 \sqsubseteq 0$$

$$\text{pour } v(p) = 0 \sqsubseteq H100 \sqsubseteq 001 \sqsubseteq 1$$

De là, comme le système P, N est complet fonctionnellement, tous les foncteurs sont exprimables par les trois combinatoires **H, 1, 0**.

- Ainsi $(p \rightarrow p) \rightarrow p$ serait en notation préfixée et parenthésage combinatoire $P(Ppp)p$ et donc **H1(H1pp)p**, donc **H1(H111)1** $\sqsubseteq$ $\sqsubseteq 11111 \sqsubseteq 11111 \sqsubseteq 111 \sqsubseteq 1$

et $\mathbf{H1(H100)0} \sqsubseteq \mathbf{H10001} \sqsubseteq \mathbf{00101} \sqsubseteq \mathbf{101} \sqsubseteq \mathbf{0}$

$\sim p \quad q$, soit $\mathbf{P(Np)q}$, serait $\mathbf{H1(H1p0)q}$.

- Nous conservons les parenthèses par commodité, mais elles sont éliminables, comme on l'a vu. $\mathbf{H1(H1p0)q}$, par exemple, se notera $\square\square\square\mathbf{H1}\square\square\square\mathbf{H1p0q}\square$
- Dans chaque formule F exprimée en foncteurs N et P , nous remplaçons chaque occurrence de N et chaque occurrence de P par leurs équivalents combinatoires, par simplicité, mais comme chaque formule F' exprimée ainsi est une combinaison (applicative) de $\mathbf{H}, \mathbf{1}, \mathbf{0}, p_1, p_2, \dots, p_n$, où $p_1, p_2, \dots, p_n$ sont les n variables distinctes de F , et comme toute combinaison (applicative) peut être obtenue par un combinateur (classique) qu'on sait produire, quand il précède et affecte les éléments constitutants de la combinaison (comme nous l'avons rappelé) il existe un combinateur X tel que

$$X\mathbf{H10}p_1p_2\dots p_n \sqsubseteq F'.$$

On exprime alors F par $X\mathbf{H10}^{24}$. On a par exemple pour $(p \rightarrow p) \rightarrow p$, c'est-à-dire $\mathbf{P(Ppp)p}$, soit $\mathbf{H1(H1pp)p}$, l'expression $\mathbf{W(B^2W(B^4W^2(BC(BB^3))))H1}$ car $\mathbf{W(B^2W(B^4W^2(BC(BB^3))))H1p} \sqsubseteq \mathbf{B^2W(B^4W^2(BC(BB^3)))HH1p} \sqsubseteq \mathbf{B^4W^2(BC(BB^3))HH11p} \sqsubseteq \mathbf{W^2(BC(BB^3))HH11p} \sqsubseteq \mathbf{BC(BB^3)HH11ppp} \sqsubseteq \mathbf{BB^3H1H1ppp} \sqsubseteq \mathbf{B^3(H1)H1ppp} \sqsubseteq \mathbf{H1(H1pp)p}$.

C'est d'une façon comparable que les formules des langages $\mathbf{P}, \mathbf{1}, \mathbf{0}$ et $\square, \mathbf{1}, \mathbf{0}$ ont déjà donné lieu dans notre analyse à cette deuxième vague du traitement combinatoire.

Les formules du calcul propositionnel sont ainsi réexprimées sans variables ni foncteurs, et avec une seule occurrence de chaque élément, ce qui est naturel car on s'adresse à telle ou telle entité, et non aux occurrences du symbole qui s'y rapporte. Elles peuvent même l'être sans parenthésage, comme on l'a vu. À vrai dire, la traduction obtenue est applicative et combinatoire puisque la présence de $\square$ qui n'est pas un combinateur, est requise (y compris quand elle est sous-entendue). On peut toutefois se borner à la qualifier de combinatoire en raison du fait que l'applicateur semble indispensable dans toutes les formalisations possibles et n'est donc pas discriminant.

- Les combinateurs $\mathbf{H}, \mathbf{1}, \mathbf{0}$ sont respectivement en système standard $\mathbf{CC}, \mathbf{K}$ et $\mathbf{KI}$. On a en effet : $\mathbf{CCxyz} \sqsubseteq \mathbf{Cyxz} \sqsubseteq \mathbf{yzx}$. $\mathbf{H1pq}$ devient donc $\mathbf{CCKpq}$, soit pour $\mathbf{CCK11}, \mathbf{CCK10}, \mathbf{CCK01}, \mathbf{CCK00}$, successivement :

$$\mathbf{CCKKK} \sqsubseteq \mathbf{CKKK} \sqsubseteq \mathbf{KKK} \sqsubseteq \mathbf{K}, \text{ c'est-à-dire } \mathbf{1}$$

²⁴ On peut donner à X la suite $\mathbf{H10}p_1p_2\dots p_n$ et utiliser d'abord des éliminateurs pour exclure ceux des éléments $\mathbf{H}, \mathbf{1}, \mathbf{0}$ qui ne figureraient pas dans F' . On peut aussi pour abrégé donner à X une suite ne comportant que ceux des éléments $\mathbf{H}, \mathbf{1}, \mathbf{0}$ qui figurent dans F' . L'expression $X\mathbf{H10}$ peut alors ne pas comprendre certains des éléments $\mathbf{H}, \mathbf{1}, \mathbf{0}$. Sous ses différentes formes, comme elle est privée des variables $p_1p_2\dots p_n$, elle ne peut donner lieu au calcul que celles-ci permettraient *via* leur valeur. La LC, plutôt que de supprimer les variables, veut surtout éviter qu'elles soient des réalités à côté des constantes, et à parité. Elle exige donc que les variables soient assignées dans leur nombre et leur place par des constantes dont ce soit le rôle, à savoir les combinateurs. Or, c'est bien $X\mathbf{H10}$ qui détermine le nombre et la place des $p_1p_2\dots p_n$.

$CCKK(KI) \sqsubseteq CCK(KI) \sqsubseteq K(KI)K \sqsubseteq KI$, c'est-à-dire **0**

$CCK(KI)K \sqsubseteq C(KI)KK \sqsubseteq KIKK \sqsubseteq IK \sqsubseteq K$, c'est-à-dire **1**

$CCK(KI)(KI) \sqsubseteq C(KI)K(KI) \sqsubseteq KI(KI)K \sqsubseteq IK \sqsubseteq K$, c'est-à-dire **1**.

- On peut évidemment conserver P, N comme abréviation commode des combinateurs associés (topiques ou standards), et donc la forme habituelle du calcul des propositions $\square$
- D'autres combinateurs pourraient être employés pour obtenir d'autres systèmes propositionnels de primitifs $\square$
- *Le remplacement des foncteurs par des combinateurs, malgré sa simplicité, ne transcrit pas un système de notation dans un autre. Il repense des entités selon les normes qu'on a exposées de la logique combinatoire.* Ainsi l'irruption d'une valeur 1 dans les données strictement dites pour $v(P00) = 1$ disparaît-elle dans le traitement qui devient $H100 \sqsubseteq M01 \sqsubseteq \square$. Cela est important si on estime qu'une explication rationnelle doit se faire sans importation *ad hoc*, c'est-à-dire en système clos, bref combinatoirement (et bien que la recherche doive évidemment «ouvrir» les systèmes pour recruter les éléments nécessaires à une bonne clôture). Ce serait une grande erreur de réduire ce que nous avons nommé «le style combinatoire» (et d'une manière générale un style, qu'il soit artistique ou mathématique) à un certain goût pour tels ou tels moyens d'expression sans voir qu'ils sont au service d'un projet spécifique²⁵

Dans la solution qui est proposée ici comme dans l'arithmétique combinatoire dont on a donné une idée plus haut :

- Chaque valeur de vérité est assimilée à un combinateur comme l'était chaque entier $\square$
- Chaque foncteur propositionnel est assimilé à un combinateur comme l'était chaque opération arithmétique $\square$
- Les combinateurs qui interviennent sont exprimables en combinateurs standards (ou dans n'importe quel système complet de combinateurs)²⁶.

Ce traitement permet aussi de souligner l'importance de s'adresser à des objets quelconques $\square$

²⁵ Dans le cadre apparenté du calcul- λ , Church a déjà réexprimé à sa manière les foncteurs, *The calculi of lambda-conversion*, Princeton, 1941, p. 68-69. En bref, il procède ainsi:

(1) Il se donne l'axiome unique du calcul de λ -conversion, à savoir la formule exprimant l'entier 2 (identifié maintenant au vrai) $\square ab. a(ab) \square$

(2) Il ajoute les quatre règles de déduction du calcul de λ -conversion. Il obtient donc les thèses du système $\square$

(3) Il dit à quelles formules de ce système correspondent $\sim A, A \sqsubseteq B, A \sqsupset B$ (au moyen de définitions complexes) $\square$

(4) Il ajoute que dans ce système (qui est consistant) $A \sqsupset B$ n'est une thèse que si soit A soit B est une thèse, contrairement aux systèmes classiques. Les objectifs de Church sont donc assez différents des nôtres.

²⁶ Cela illustre bien la formule déjà citée de D. Dubarle « $\square$ n'y a même pas lieu en principe de distinguer les symboles affectés aux objets et à ce que la pensée entend leur appliquer».

- On sait établir d'une part que toute formule propositionnelle E' comportant $p_1, p_2, \dots, p_n$ et des foncteurs $F_1, F_2, \dots, F_m$ peut être exprimée comme une formule E comportant $p_1, p_2, \dots, p_n$ et le seul foncteur D □
- On sait d'autre part, voir [16], que pour toute formule ainsi décrite il existe un combinateur X tel que□

$$XDp_1p_2\dots p_n \sqsubseteq E$$

On dit alors que XD exprime (sans variables) la formule E , et donc aussi E' .

Si la logique combinatoire ne s'adressait pas à des objets quelconques, on pourrait exprimer E sans le foncteur D , en sous-entendant D comme on sous-entend $p_1p_2\dots p_n$ et on pourrait estimer qu'on a éliminé les foncteurs au profit des combinateurs comme on a éliminé les variables. Si cela n'est pas possible c'est justement parce que D possède une individualité, et qui importe au traitement, alors que $p_1p_2\dots p_n$ n'en possèdent pas. Il faudrait dire que c'est D comme tel qui est sous-entendu. On lui garderait donc son sens usuel, on ne l'atteindrait pas par une autre analyse ; on se bornerait à raccourcir les formules qui l'emploient. D ne peut être évité au profit des combinateurs que si un combinateur figure à sa place en se trouvant, au moins extensionnellement, indiscernable. Dans notre langage, le foncteur D , de table 0111, serait $Pp(Pq0)$, donc **H1p(H1q0)**.

Ces remarques nous permettent de dire ce qui fait pour nous le «**style**□ combinatoire :

- (1) Ne se référer qu'à des objets quelconques, pour éviter de spécifier intuitivement ce à quoi on s'adresse, et donc se référer à des objets tantôt opérateurs, tantôt opérandes□
- (2) Travailler dans un système où aucun objet n'entre comme un *deus ex machina*, donc travailler dans un système clos n'admettant que des modifications internes□
- (3) Éliminer les faux objets (comme les variables ou les parenthèses) et même, autant qu'il est possible, les vrais objets (comme les foncteurs) mais qui sont des constantes non combinatoires, au profit des combinateurs, qui par définition gardent (et sont seuls à garder) la clôture d'un système, ou à tout le moins, au profit d'opérateurs aussi proches que possible, comme □, des combinateurs²⁷

Évidemment, on peut dire que l'arithmétique combinatoire ne parle plus strictement de nombres, mais d'objets appartenant à un domaine muni de la même structure que celle des entiers [14], p. 331. On peut dire qu'en écrivant **1** $xy \sqsubseteq x$, **0** $xy \sqsubseteq y$, on ne parle plus du vrai et du faux, mais seulement du «**fait** que les valeurs de vérité sont normalement utilisées pour sélectionner l'un des deux termes d'une alternative□, notamment dans un

²⁷ Si on n'acceptait pas l'assimilation de **1**, **0** et des valeurs 1, 0, on pourrait se borner à traiter **1**, **0** comme de simples constantes propositionnelles. Les ajouter à □ serait encore plus proche, à notre avis, du style combinatoire (sous sa forme la plus forte) que l'adjonction de P aux combinateurs standards, car les constantes de proposition (toujours opérandes) n'ont qu'un rôle de variable, alors que P est une constante proprement dite, et donc «**ivale**□ des combinateurs. Rappelons que dans le langage □, **1**, **0** on n'a pas utilisé les règles de réduction de **1**, **0**.

énoncé conditionnel d'un langage de programmation [27], p. 38²⁸. On peut objecter aussi qu'on a «changé de sujet» (comme dirait Quine), plutôt qu'exprimé le sujet autrement. Pourtant il n'est pas nécessaire de tout dire sur un concept pour en traiter valablement, et il peut être nécessaire de «changer de sujet» pour obtenir la pertinence. Il suffit d'avoir conscience que les investigations possibles ne sont pas terminées, qu'un objet n'est jamais que «le géométral de toutes les perspectives», selon une formule de Merleau-Ponty [23], p. 162. Il reste toujours à apprécier l'intérêt d'un traitement combinatoire dans tel ou tel cas.

Faut-il estimer qu'en disant qu'une fonction est un objet, la LC voudrait dire aussi, ou plutôt, qu'un objet est une fonction (une fonction-opérateur) ? On retrouve là l'idée qu'elle serait une logique de l'opérateur. Peut-on concilier :

- a) toute fonction est un objet
- b) tout objet est une fonction ?

L'énoncé a) s'oppose, comme on l'a vu, à l'idée qu'une fonction serait seulement une correspondance. L'énoncé b) ne peut pas signifier que tout ce qu'on estime être quelque chose en mathématiques est à chaque mention une fonction qui opère. Cela reviendrait à dire que tous les êtres sont en toutes circonstances des agents, en violation du principe de neutralité ontologique, et en supprimant la possibilité même des opérandes. b) signifie qu'il faut supposer que tout objet est susceptible d'être agent (activé par λ). À chaque fois qu'on réussit à penser un être comme étant, à tel ou tel égard, un opérateur, alors qu'il était vu tout autrement, on valide un peu plus la sagesse du principe de neutralité ontologique : on ne doit pas supposer des données inactives face à des agents. Inversement, les entiers ou les valeurs de vérité, 0, 1, etc., devenus combinateurs à certains égards, doivent pouvoir être traités (et c'est le cas) comme de simples opérandes, selon le principe de réversibilité.

Bref, pour qu'on puisse mener à bien le programme de trouver la fonction-opérateur derrière un objet, il faut qu'une fonction ait été analysée comme objet, et donc qu'on ne se réfère pas déjà à des opérateurs.

À ce que la LC soit une logique de l'objet quelconque, on peut opposer, pourtant, ce qu'on nomme aujourd'hui la curryfication des fonctions. Rappelons-en le principe.

5. STATUT DE LA CURRYFICATION ET OPERATEURS

Soit F une fonction à deux arguments x, y . Schönfinkel la remplace par une fonction unaire f qui, appliquée à son argument x , prend la valeur fx qui est elle-même une fonction (ceci est novateur²⁹), fonction qui est unaire et qui prend pour son argument y la

²⁸ On pourrait d'ailleurs employer un discriminateur, notamment $\lambda x_1 x_2 x_3. x_1$ tel que $\lambda x_1 x_2 x_3. x_1$, si $x_3 =_{os} x_1$; $\lambda x_1 x_2 x_3. x_2$, si $x_3 \neq_{os} x_1$. Avec les constantes de proposition 1, 0, on aurait $\lambda p. 10p = 1$ si $p = 1$, et $\lambda p. 10p = 0$ si $p \neq 1$.

²⁹ À ceci près que Frege avait déjà employé cette méthode (que Schönfinkel l'ait su ou non). Nous emploierons aussi par commodité des expressions comme «curryfié», «décurryfié», et même à propos

valeur prise par $F(x, y)$. Cette transformation, dite «Curryfication» – «Curryf» – se généralise à toutes les fonctions qui ont un nombre supérieur d’arguments. Elle convient aussi quand on s’exprime en termes d’opérateurs et d’opérandes.

Ainsi, dans l’expression $2 \square 3$, lue «Deux fois trois», la fonction $\square$ est à deux arguments. Si on fournit le premier seulement des arguments, on obtient une fonction unaire $2 -$ c’est-à-dire la fonction «Doubler» – qui affectera l’ancien argument 3, soit $(2)3$, simplifié en 23 , et en obtenant le résultat de $2 \square 3$. Plus généralement, on peut se représenter $x \square y$, ou $M(x, y)$, comme xy . Il est clair que x est requis pour doter x d’un sens, mais évidemment x ne produit une fonction (et non seulement un schéma) qu’ «Après substitution d’une valeur pour x », comme le précise Schönfinkel [29], § 2, p. 15. En tout cas, si on dit « $\square$ sur x forme la fonction $x\square$ », au lieu de dire exclusivement, par exemple, «forme la fonction G », ce n’est pas seulement pour rappeler la formule d’origine, x fait partie de ce que dit la nouvelle fonction. Autrement dit, dans x , on traite x comme un paramètre (c’est une constante comme 2, mais de contenu non précisé)³⁰. Quant à 2 , c’est la fonction unaire «Trois» qui s’applique à un argument comme 2 pour obtenir la fonction «3 fois».

De la même façon, ce qu’on a pu comprendre d’abord $W(x, y)$ pourra se formuler $(Wx)y$, et x est bien constitutif du sens de (Wx) , bien qu’on ne puisse donner ce sens qu’ «Après substitution d’une valeur pour x ». (Wx) ne prend toujours pas de sens univoque quand on substitue à x une indéterminée, il en prend un si on substitue à x un objet comme **I** ; (WI) signifie alors «redoubler». Dans $(WI)y$, (WI) redouble ce que note y .

On peut interpréter (Kx) , (Wx) comme Schönfinkel interprète la fonction unaire $(x -)$ formée à partir de $x - y$: «la différence entre x et n’importe quelle grandeur donnée» (§ 2, p. 15). On aura pour (Kx) «l’élimination par rapport à x de n’importe quel objet donné», ou pour (Wx) «la duplication par rapport à x de n’importe quel objet donné»³¹. Contrairement à nos exemples, il arrive que la curryfication ne puisse pas donner lieu à des formulations intuitives, mais cela n’affecte pas le traitement comme tel.

Schönfinkel y parvient (§ 2, p. 14-15) en distinguant deux types de fonction : «le cas le plus simple» où la fonction produit un objet du domaine fondamental, fonction qui intéresse donc pour elle-même, comme $2 \square 3$ ou $(2)3$, et «la notion élargie» où la fonction «ne produit pas encore un objet du domaine fondamentale», mais une autre fonction, comme 2 et (2) , qui s’interpose pour constituer une fonction du «cas le plus simple». Nous dirons «fonction simple» et «fonction élargie» pour rester proche des

des travaux de Schönfinkel (qui dit seulement «Transformation», «Umformung» [29], § 2, pour ce qu’on nomme aujourd’hui «Curryfication», ou «Curriage»).

³⁰ L’expression de Schönfinkel («Après substitution d’une valeur pour x ») est d’ailleurs imparfaite, x n’étant pas une variable. Il faudrait plutôt «Après attribution d’une valeur à x ».

³¹ L’expression de Schönfinkel est de nouveau imparfaite car la différence [Differenz] est le résultat d’une soustraction, donc une valeur numérique, alors qu’ici on se réfère à une fonction ; il faudrait «la fonction de soustraire de x n’importe quelle grandeur donnée». «Élimination» et «duplication» conviennent si on les comprend comme des actions et non comme leurs résultats.

expressions de Schönfinkel. Dans une conception causaliste de la fonction, celle de la LC (cf. III, (8)), la valeur d'une fonction étant, à la lettre, l'effet qu'elle produit, on peut admettre qu'une fonction n'ait pour valeur que d'en construire une autre, comme un mécanisme peut avoir pour effet d'en enclancher un autre, voire celui-ci un troisième, etc. jusqu'à obtenir l'effet qui était en lui-même recherché. En bref, une fonction «simple» étant un outil d'obtention pour un certain objet, une fonction «élargie» est alors un outil d'outil³².

Dans le cas où plus de deux variables sont requises, on a plusieurs possibilités qu'on va expliciter sur l'exemple de **B**. Ou bien

- a) Sans curryfication, **B** est une fonction ternaire qui s'applique à x, y, z et obtient $x(yz)$, soit $\mathbf{B}(x, y, z) \sqcap x(yz)$. Ou bien,
- b) Avec curryfication complète, **B** est une fonction unaire qui s'applique à x , en formant $\mathbf{B}x$ qui est une fonction unaire qui s'applique à y en formant une fonction unaire $\mathbf{B}xy$ qui s'applique à z en obtenant $x(yz)$, soit $((\mathbf{B}x)y)z \sqcap x(yz)$, ou par (AG) $\mathbf{B}xyz \sqcap x(yz)$. Ou bien,
- c) Avec curryfication initiale (ou «de bête»), **B** est une fonction unaire qui s'applique à x , en formant $\mathbf{B}x$ qui est une fonction binaire qui s'applique à y, z , en obtenant $x(yz)$, soit $(\mathbf{B}x)(y, z) \sqcap x(yz)$. De manière générale, une fonction F telle que $f(x_1, x_2, x_3, \dots, x_n)$ est comprise comme une fonction f unaire qui forme (fx_1) laquelle est une fonction de $n-1$ arguments (ceux qui restent droite). Ou bien,
- d) Avec currification «terminale», **B** est une fonction binaire qui s'applique à x, y, z en formant $\mathbf{B}(x, y)z \sqcap x(yz)$.

Les versions partielles c) et d) s'expliquent évidemment par le fait qu'on peut appliquer, sur leur modèle, le même traitement aux fonctions qui restent et qui ne sont pas unaires. C'est pourquoi on s'adressera à la forme complète de la currification. Cependant, c'est la forme d) qui permet de comprendre les raisons d'être de la curryfication, comme on va le montrer³³.

Remarquons d'abord les deux points suivants :

1. Le principe de libre application ne suffit pas à valider (curryf), car si on peut bien appliquer **B** à x , $\mathbf{B}x$ à y , etc., la question est de savoir ce qu'on exprime et si cela est explicatif de l'effet d'un combinateur. Curry admettrait que si on peut appliquer

³² Parler de fonction fx , ou $f(x)$, est incorrect dans le cas des fonctions dites «simples» – comme c'est bien connu – parce qu'on confondrait fx , qui est par exemple un entier, avec f qui est son moyen d'obtention, la fonction proprement dite. En revanche, dans le cadre des fonctions dites «élargies», donc quand la curryfication intervient, on peut dire «la fonction fx », sans qu'il s'agisse de f mal désignée, car fx , justement, n'est plus un entier, mais un nouveau moyen d'obtention, donc une fonction, même si ce n'est pas encore un moyen d'obtention d'un entier. D'autre part, fx (qui est à double face) demeure bien la valeur de fonction de f .

³³ Nous disons que la curryfication est de tête ou terminale selon que la fonction traitée comme unaire est en tête ou en fin de l'expression fonctionnelle. Dans les deux cas, la curryfication fait intervenir une fonction «au sens élargi».

n'importe quel objet à n'importe quel autre, il ne s'ensuit pas qu'il soit toujours fructueux de le faire.

2. On ne peut pas comprendre une expression $Xx_1x_2x_{n-1}x_n \sqsupset Q$, de la manière suivante: $(Xx_1) \sqsupset (Xx_1)$ selon (AG) et ($\sqsupset$), $((Xx_1)x_2) \sqsupset ((Xx_1)x_2)$, etc. selon (AG) et ($\sqsupset$), et ainsi de suite jusqu'à $((((Xx_1)x_2)x_{n-1})x_n) \sqsupset Q$ selon une règle propre au membre de gauche tout entier. Il ne s'agirait pas là de la curryfication qu'on voudrait fonder. $(Xx_1) \sqsupset (Xx_1)$ par ($\sqsupset$) dit seulement que (Xx_1) se réduit à (Xx_1) alors que la curryfication dit que la fonction X sur x forme une *nouvelle* fonction, notée Xx .

Ni (AG) et ($\sqsupset$) n'obtiennent à eux seuls (curryf). Il ne suffit pas de transcrire $F(x,y)$ en $(fx)y$, il faut ajouter à cela la clause suivante : fx est la valeur de f pour x et en même temps une fonction unaire nouvelle où x est un paramètre (ce qui est donné verbalement par Schönfinkel, mais qui n'est pas comme telle une mesure du système, simplement une de ses interprétations possibles).

À partir de son introduction de **B** (§ 3, p. 17-18), on peut comprendre pourquoi Schönfinkel écrit d'abord ce qui l'intéresse sur un modèle général qui serait :

$$[X'(x_1, x_2, \dots, x_{n-1})]x_n = \dots$$

et notamment ici : $[X'(x_1, x_2)]x_3 = \dots$ ³⁴.

Il s'agit d'exprimer : «On applique la fonction g à x puis la fonction f à ce qui est obtenu», soit $f(gx)$. On s'adresse ainsi à une fonction déterminée elle-même par f et par g (dite improprement «*composée* de f et g »), et qui est appliquée à x . Si on appelle **B'** cette fonction, il vient : $[B'(f, g)]x = f(gx)$. La fonction **B'**, d'arguments f et g , est binaire, mais elle forme une fonction unaire puisqu'elle porte sur x . On peut ensuite, pour normaliser, transformer $[B'(f, g)]$ en des successions de fonctions unaires, par curryfication, et en écrivant **Bfg**, comme le fait Schönfinkel, c'est-à-dire $((Bf)g)$, mais la seule fonction qui ait besoin d'être unaire, dans un tel programme, c'est la dernière, la fonction dite «*simple*», celle qui obtient directement l'expression-but. C'est donc la forme de curryfication dite «*terminale*» qui importe. Schönfinkel prend ensuite des mesures tout à fait comparables pour une fonction **S'**.

À vrai dire, néanmoins, **B'** devrait être une composée particulière des éléments f, g, x , et non de f et de g , puisque $f(gx)$ n'est pas une fonction f d'une fonction g , mais une fonction f «*d'une valeur de fonction*» (gx), comme le dit Schönfinkel («On parle improprement d'une *fonction de fonction*»). Dans une expression sans impropriété, donc, **B'** serait bien une fonction ternaire. C'est seulement par concession à un abus d'expression qu'on a besoin de parler d'une fonction unaire $[B'(f, g)]$ de x , alors que, si on s'exprime sans impropriété, on a une fonction ternaire, une certaine composition de f, g, x , donc $B''(f, g, x) = f(gx)$, (si on emploie des égalités à la manière de Schönfinkel).

³⁴ À ceci près que Schönfinkel s'adresse souvent à des signes marquant habituellement des fonctions, $\sqsupset, \sqsubset$, plutôt qu'à des x_n , pour aller plus directement vers les applications qui lui importent, voir plus bas. Il écrit $[Z'(\sqsupset, \sqsubset)]x = \sqsupset(\sqsubset x)$, où Z correspond à ce qu'on note aujourd'hui **B**.

On est donc fondé à *penser* la fonction sans la curryfier, dès qu'on la comprend plus exactement. C'est la *première raison* pour ne pas majorer l'importance de la curryfication : comme elle s'adresse à des données impropres, elle est elle-même impropre.

- *Deuxième raison* : comment sait-on que pour un x_1 quelconque f peut obtenir la fonction (fx_1) nécessaire pour la curryfication, c'est-à-dire qui produit la fonction $(fx_1)x_2$, etc., que $\mathbf{B}$ pour un x_1 quelconque peut obtenir la fonction $\mathbf{B}x_1$ qui produit la fonction $(\mathbf{B}x_1)x_2$, etc.³⁵. Suffit-il de pouvoir comprendre et décomposer une expression fonctionnelle d'une certaine manière pour valider une thèse sur l'existence des entités correspondantes ? En revanche, conçu sans curryfication, $\mathbf{B}$ ne pose pas de difficultés : il est clair que $x_1x_2x_3$ peut se restructurer $x_1(x_2x_3)$, et on appelle $\mathbf{B}$, au fond, tout intervenant qui le fait.
- *Troisième raison* : Schönfinkel n'en parle qu'au titre des «*moyens auxiliaires*», § 2, p. 14, et en relativise lui-même l'importance (par les mots que nous soulignons) : «*Conformément à notre convention [Verabredung] antérieure, nous préférons remplacer Z' [ici $\mathbf{B}'$] par la fonction correspondante à un argument*», § 3, p. 18. Il signale aussi qu'on peut toujours decurryfier une expression. «*On peut évidemment aussi, à l'inverse, considérer toujours une expression fx_y comme obtenue de $F(x, y)$ où F est déterminé de manière unique par le f donné*» (p. 17).

Toutefois, on ne peut pas nous reprocher de donner à (curryf) plus d'importance que l'auteur ne le fait lui-même car quand on défend «*Une logique de l'opérateur*», c'est justement qu'on surestime le rôle de la curryfication (ce qui explique notre insistance).

- *Quatrième raison*, fournie par l'objectif *essentiel* de la curryfication : «*L'essentiel ici est qu'on n'a pas à penser les valeurs comme substituées simultanément pour x et y* » (§ 2, p. 15). La curryfication n'a donc pas pour but de n'avoir que des opérateurs (qu'ils obtiennent des objets ou qu'ils obtiennent des opérateurs), mais d'éviter d'avoir à faire plusieurs choses à la fois, d'avoir à remplacer plusieurs objets du domaine fondamental en même temps. La curryfication n'est pas faite pour éviter de s'adresser à des objets, en ne parlant que d'opérateurs.

Il faut montrer en effet que certaines opérations effectuées simultanément («*En parallèle*») sont mathématiquement compossibles (même si quelque donnée empirique interdisait de les effectuer en même temps), et cela revient à montrer qu'on peut en mettre d'autres en oeuvre successivement avec le même effet³⁶.

³⁵ La notion d'objet quelconque se marque chez Schönfinkel par l'expression «*Il est soumis à aucune limitation*» (pour le x auquel f s'applique, § 2, p. 15, ou le x auquel $\mathbf{I}$ et $\mathbf{K}$ s'appliquent, § 3, p. 16-17) ; x n'est donc en rien qualifié de fonction (ni d'opérateur).

³⁶ Ainsi, une fois établie que la règle de substitution simple du calcul des propositions conserve la validité (on peut remplacer notamment p par q , ou q par p), une règle de substitution dite «*Simultanée*» peut être définie par laquelle plusieurs variables sont remplacées simultanément par plusieurs formules, non nécessairement distinctes (remplacer par exemple p par q et aussi q par p , ou remplacer p par un substitut comportant q et aussi q par un substitut comportant p), et il faut montrer que cette règle conserve aussi la validité. On le fait classiquement en démontrant qu'elle peut toujours être évitée par une succession de substitutions simples, à savoir $2n$ substitutions au plus s'il y a n variables distinctes sur lesquelles on effectue des substitutions simultanées. Certes, au lieu de curryfier, on pourrait estimer que f

Or, il est très simple de former une fonction n -aire par un seul acte à la fois, il suffit d'introduire un à un ses n arguments. Or, c'est là ce que permet l'association à gauche, sans qu'on ait besoin de curryfier.

(xy) signifiera qu'on fait intervenir x puis y , ou qu'on remplace successivement, et non simultanément, x par son contenu ou par telle valeur a puis y par le sien. Les « $(\lambda x) \lambda y$ » correspondent à « $\lambda x \lambda y$ traite successivement» ; il faut donc deux objets pour qu'on puisse exprimer « $\lambda x \lambda y$ successivement», au sens le plus immédiat du terme. Ainsi, dans les cas où f note une fonction et où x, y, z sont des arguments qui sont dans son domaine :

(fx) est l'application d'une fonction unaire à son argument x , comme $f(x)$.

$((fx)y)$ est l'application d'une fonction binaire à ses arguments x, y comme le $f(x, y)$ usuel, mais où on fait intervenir successivement x et y , alors que dans $f(x, y)$ on suppose qu'ils sont simultanément introduits ou remplacés, en considérant comme extramathématique le fait empirique qu'on doit écrire y l'instant qui suit celui où on a écrit x . Et ainsi de suite pour $((fx)y)z$, etc.

Avec (AG) les parenthèses indiquent l'élément unique auquel on applique ce qui précède à gauche, alors que la notation classique suppose en droit la simultanéité des traitements.

On comprendra que $(fx)y$ désigne la fonction binaire pour laquelle on a introduit d'abord le premier argument *comme si* elle devait être une fonction unaire, et ensuite le second puisqu'elle est en réalité binaire. Dans $(fx)y$, (fx) n'est pas une fonction unaire mais *l'aurait été* si on s'en était tenu à cette notation, mais puisqu'on a ajouté y , (fx) n'a été qu'une partie de la fonction binaire et rien d'autre³⁷. On donne par là un moyen de construire toutes les fonctions n -aires en ajoutant simplement un élément à une fonction $(n-1)$ -aire. Inversement, si $fx_1x_2...x_n$ est une fonction n -aire, $fx_1x_2...x_{n-1}$ est une fonction $(n-1)$ -aire. Notons qu'on ne s'adresse qu'à des sous-expressions bien formées, comme il le faut pour que la procédure s'applique toujours, de proche en proche³⁸.

telle que $f(x_1, \dots, x_n)$ a pour argument (unique) la suite ordonnée $\langle x_1, \dots, x_n \rangle$, mais chacun des x_i est un objet immédiat et non $\langle x_1, \dots, x_n \rangle$ (on parle d'ailleurs de chacun des x_i comme du i -ème « λ argument secondaire» de f). La curryfication, en cela, met bien en jeu une des « λ actions les plus simples de l'esprit humain», comme le demande Curry de manière générale, [2], p. 368 (elle traite une seule donnée et non composée), mais elle a pourtant ses propres complexités, que nous analysons par ailleurs.

³⁷ Nous admettons des expressions comme « λx la fonction fx » (qui sont abusives quand f est une fonction « λ simple» puisqu'il faudrait $\lambda x fx$), à titre d'abréviation d'une expression- λ .

³⁸ À cet égard, (AG) est le symétrique de la règle (U) qui chez Schönfinkel élimine la dernière variable liée x d'une expression telle que $fx \vdash gx$ où f et g sont données comme unaires, en obtenant Ufg (§5, p. 20-21). À vrai dire, si on veut pouvoir employer U pour une expression comme $fywx \vdash gzwx$ (ce que fait Schönfinkel) en obtenant $U(fyw)(gzw)$, il faut que fyw et gzw soient unaires. Chaque fonction doit donc être unaire et non pas seulement la fonction-but de l'expression d'origine. Toutefois, cela n'impose pas la curryfication, mais seulement (AG) et l'interprétation que nous avons donnée. Ajoutons que U n'est pas un combinateur, et qu'une logique strictement combinatoire ne peut pas l'employer.

En ce sens, (AG) tire toutes les conséquences du principe de la notation préfixée des opérateurs. Une des raisons de celle-ci est de pouvoir écrire les opérateurs qui sont plus que binaires car si on écrit les opérateurs en plaçant l'opérateur entre les arguments, on ne peut noter ainsi que les binaires. On en vient donc à composer une série unique où on passe d'un opérateur n -aire à un opérateur $(n + 1)$ -aire en ajoutant seulement une variable de plus à l'extrême-droite, pour un opérateur toujours situé à l'extrême-gauche. Nous appellerons cette procédure «*la formation graduelle*» (FG) des fonctions (ou des opérateurs). En conséquence, si on note g_1 la fonction fx_1 , g_2 la fonction fx_1x_2 etc., chaque g_i est construite *comme si* on s'adressait à une fonction unaire.

Il n'est donc pas nécessaire de traiter tous les opérateurs comme unaires, il suffit qu'on veuille les écrire tous sur le modèle pour lequel on a écrit les opérateurs unaires, l'opérateur suivi des n éléments qu'il lui faut. Autrement dit, dans la fonction $fxyz$, fx n'est pas «*un objet du domaine fondamental*», en effet, ni non plus une partie encore mal formée d'une fonction ternaire incomplète, mais elle n'a pas non plus à être une fonction unaire qui serait en acte et opèrerait dans $fxyz$ ³⁹. En bref

- (curryf) a besoin de (AG). Pour remplacer une fonction binaire $F(x, y)$ par deux fonctions unaires, il convient d'écrire $(fx)y$
- Mais (curryf) n'est pas nécessaire pour lire les expressions associées par (AG), comme on l'a montré plus haut, grâce à (FG)
- Enfin, (AG) n'est pas seulement, non plus, au service de (FG). Elle permet aussi d'exprimer la chaîne discursive de la manière la plus immédiate (dans les langues européennes, construire des expressions de gauche à droite), voir III, (5)
- Certes, (curryf) présente au moins deux intérêts : (1) ne supposer qu'un acte élémentaire de l'esprit humain (ce qui est capital chez Curry), en appliquant f à un objet et non F à 2, 3, ..., n objets qu'il faut supposer traités simultanément (mais notre lecture satisfait ce point), et (2) ne laisser qu'un seul type de fonction à traiter, pour simplifier les preuves, mais cet avantage ne disparaît pas. On peut toujours traiter $F(x_1, x_2, \dots, x_n)$ *comme si* on avait des opérateurs successifs dont chacun n'est appliqué qu'à un seul objet, et non pas un opérateur unique appliqué successivement à chacun de ses n objets, bien que ce soit cette donnée qui soit propre et que l'autre soit reconstructive.

Il reste à préciser ce qu'on fait en plaçant un combinateur comme **B** devant une suite comme xyz . Puisqu'on admet (AG), on veut obtenir $x(yz)$ à partir de $(xy)z$. Or, on ne forme pas **B** $(xy)z$, mais **B** xyz , c'est-à-dire $((\mathbf{B}x)y)z$. Notre lecture permet de comprendre que $((\mathbf{B}x)y)z$ revient à introduire **B** puis x , puis y , puis z . On a donc bien **B** sur xyz mais en plusieurs pas, comme $(xy)z$ signifie «*puis x , puis y , puis z* ».

Toutefois, la curryfication n'est-elle pas nécessaire pour qu'on puisse appliquer une fonction à une fonction ? Pour qu'on puisse exprimer sans variable par exemple la converse fyx de fxy en Cf tel que $Cfxy \equiv fyx$, il faut appliquer **C** (une fonction) à f

³⁹ On peut donc comprendre pourquoi Quine [25] a pu estimer «*gratuite*» la curryfication et a proposé une «*interprétation*» (que nous n'examinerons pas ici) des expressions formées par (AG).

(une fonction). En fait, on n'a pas besoin de la curryfication pour appliquer une fonction à une fonction. Le principe de neutralité ontologique, y suffit, voir ici III, (4). Comme **C**, f , **K** sont seulement des opérands de $\square$, ils sont comme tels des objets et ils peuvent donc intervenir au même titre que a , b , etc. (ou que x , y , etc. dans les formulations en schémas). On écrit donc **BKWC** par exemple comme on écrit **Babc**, **B** s'applique à l'objet **K** d'abord et forme une partie de fonction ternaire, etc., exactement comme **B** s'applique à l'objet a et forme une partie de fonction ternaire, dans notre lecture.

Dans notre interprétation, (AG) et (FG) remplacent (curryf).

On peut donc se demander pourquoi Schönfinkel ne s'est pas contenté de la lecture simple que nous avons présentée.

L'une des raisons, au moins, est sans doute qu'on doit employer un conditionnel irréal (pour une fonction n -aire, fx_1 serait unaire, si on s'y arrêta, (fx_1x_2) serait unaire, si on s'y arrêta, etc., et $(fx_1x_2...x_{n-1})$ ne pourrait plus qu'être unaire), alors qu'une telle formulation est moins bien maîtrisée (n'étant pas vérifonctionnelle) que la formulation à l'indicatif donnée par (curryf) : (fx_1) est une fonction unaire qui sur x_2 construit $((fx_1)x_2)$, qui est une autre fonction unaire, etc.

À partir de ces analyses, nous voudrions montrer :

- Si on applique directement la curryfication pour comprendre les règles de réduction (mais on a tort de le faire), on est conduit à soutenir la conception selon laquelle la LC serait une logique de l'opérateur $\square$
- Si on applique directement les règles de formation pour comprendre les règles de réduction (et on a raison de le faire), on est conduit à soutenir la conception selon laquelle la LC est une logique de l'objet quelconque.

La conception «logique de l'opérateur $\square$ » lit (selon la curryfication)

- Pour **Bxyz** $\square$ $x(yz)$, le combinateur **B** est un opérateur unaire qui, appliqué à l'opérateur x , forme l'opérateur unaire (**Bx**) qui, appliqué à l'opérateur y , forme l'opérateur unaire (**Bxy**) qui, appliqué à son opérande z , obtient $x(yz)$ $\square$
- Ou encore, le combinateur **B** est un opérateur qui combine les opérateurs x et y pour obtenir l'opérateur complexe **Bxy** qui sur son opérande z obtient $x(yz)$.

Ainsi analysés, les combinateurs sont des moyens pour former des opérateurs complexes à partir d'opérateurs plus simples. J.-P. Desclés exprime cette conception de la manière suivante (à quelques détails près), un combinateur X combine entre eux des opérateurs plus élémentaires $f_1, ..., f_n$ pour former l'opérateur complexe $Xf_1...f_n$ dont l'action sur une suite d'opérands $a_p, ..., a_1$ est spécifiée par une règle de réduction ayant la forme :

$$Xf_1...f_na_p...a_1 \square b_1...b_q$$

où $b_1, \dots, b_q$ sont des expressions combinatoires qui expriment certains agencements entre des $f_i (i = 1, 2, \dots, n)$, les opérandes du combinateur, et des $a_j (j = 1, \dots, p)$, les opérandes de l'opérateur complexe construit à l'aide de X [11], p. 160-161.

La conception «logique de l'objet quelconque» lit (selon les seules règles de formation)

- Pour $\mathbf{B}xyz \sqsubset x(yz)$, le combinateur $\mathbf{B}$ est un objet qui, appliqué à l'objet x , forme l'objet $(\mathbf{B}x)$ qui, appliqué à l'objet y , forme l'objet $(\mathbf{B}xy)$ qui, appliqué à l'objet z , obtient l'objet $x(yz)$. Comme on l'a vu, cela revient à dire que $\mathbf{B}$ porte sur xyz mais avec une introduction de ses arguments un par un.

Dans une lecture de ce type, on en reste à la règle telle qu'elle est donnée :

$$Xx_1 \dots x_n \sqsubset y_1 \dots y_m$$

où $y_1, \dots, y_m$ sont des expressions combinatoires qui expriment certains agencements entre des $x_i (i = 1, 2, \dots, n)$.

Or, la première lecture, celle qui défend la conception «logique de l'opérateur», ne convient pas, car

1. Ni $\mathbf{I}$, $\mathbf{K}$, etc. ni ce qui est désigné par x, y , etc., x_1, x_2 , etc. (ou par a_1, a_2 , etc.) ne sont des opérateurs. Ils sont des objets quelconques, selon le principe de neutralité ontologique, essentiel à une logique combinatoire (cf. III, (4))
2. Même si on lit comme Schönfinkel en curryfiant (et on peut le faire), on ne doit pas comprendre que $\mathbf{K}$ forme sur l'opérateur x l'opérateur plus complexe $\mathbf{K}x$. On doit lire : l'objet $\mathbf{K}$, qui est ici en position d'opérateur (délégué, voir III, (7)), sur l'objet quelconque x («soumis à aucune limitation») forme l'opérateur $\mathbf{K}x$ qui sur l'objet quelconque y forme l'opérateur $\mathbf{K}xy$. $\mathbf{K}$ ne s'adresse pas à des opérateurs, il rend opérateur ce à quoi il s'adresse et selon les principes dits de neutralité ontologique et de réversibilité (III, (4))
3. (Dans une logique vraiment combinatoire, en effet, les opérateurs doivent être des combinateurs et seulement des combinateurs, bien que ce soit via l'applicateur. Or, dans la lecture que nous contestons, Xx, Xxy , etc. sont des opérateurs, mais ils ne sont ni des combinateurs ni formés par des combinateurs. On ne peut donc pas avoir recours à des opérateurs comme $\mathbf{C}x, \mathbf{B}xy$ qui s'ajouteraient aux combinateurs. Si on objecte que $\mathbf{C}x, \mathbf{B}xy$ sont eux-mêmes obtenus par les combinateurs, on répondra que si $\mathbf{B}$ obtient en deux pas $\mathbf{B}xy$, $\mathbf{B}$ n'est plus un combinateur. Il ne combine pas xy , il s'ajoute à xy . $\mathbf{B}xy$ est bien une combinaison de $\mathbf{B}, x, y$, mais $\mathbf{B}xy$ n'est pas produit par le $\mathbf{B}$ de cette combinaison. C'est par exemple $\mathbf{BIB}xy$ qui obtient $\mathbf{B}xy$. $\mathbf{BIB}xy \sqsubset \mathbf{I}(\mathbf{B}x)y \sqsubset \mathbf{B}xy$.

Peut-on dire que les combinateurs agissent de proche en proche, par relais, tout en étant, cependant, responsables des résultantes ? Le combinateur X n'agirait directement que sur x_1 en produisant un autre terme Xx_1 qui agirait sur x_2 , et ainsi de suite, X produisant bien la résultante, mais indirectement.

Remarquons, d'abord, que ce n'est pas une caractéristique du mode d'action d'un combinateur de travailler par relais : il y a des combinateurs $Xx \sqsubseteq Q$ (où Q est une combinaison applicative de x , comme $Ix \sqsubseteq x$, $Ax \sqsubseteq xx$, $Dx \sqsubseteq xx(xx)x$, (entre autres) où l'action du combinateur s'effectue sans intermédiaire, ce qui établit que l'effet d'un combinateur n'en exige pas. Mais on peut même estimer qu'il l'interdit.

En fait, l'action d'un combinateur X ne peut pas être relayée. Puisque c'est l'opérateur Bxy qui sur z obtient directement la résultante $x(yz)$, il faudrait qu'il soit obtenu lui-même de manière combinatoire, ou au moins que le premier relais de B , c'est-à-dire Bx , le soit. Mais ce n'est pas le cas. Le B de Bx ne forme pas une combinaison qui serait Bx puisqu'il figurerait dans la combinaison. Or, aucun combinateur proprement dit ne peut être présent dans la combinaison qu'il obtient (que ce soit par réduction ou différemment) puisqu'il ajouterait aux données un élément nouveau (lui-même), au lieu de les combiner. En revanche, on peut admettre que $\cap$ et $\oplus$, par exemple, opèrent par relais ($\cap$, la fonction « $\cap$ fois $\cap$ », puis x , la fonction « $\cap$ fois $\cap$ ») car rien n'a à limiter la manière dont on obtient le produit, contrairement à ce qui fait pour B l'objectif combinatoire (si du moins on l'identifie pleinement, contrairement à Schönfinkel qui n'a recours qu'à des « $\cap$ fonctions particulières $\cap$ »).

Autrement dit, on ne peut *comprendre* une logique *combinatoire* que sur des combinateurs décurryfiés. Cela veut dire que B et que C portent sur xyz . Si on joint les trois idées : les combinateurs sont des objets-opérateurs (même délégués), qu'il faut comprendre non curryfiés et en soutenant une conception causaliste, cf. III, (8), on doit les interpréter comme exprimant des actions sur les éléments de leur suite : $Kxy \sqsubseteq x$ exprime l'action d'éliminer y , $Cxyz \sqsubseteq xzy$ exprime l'action de permuter y et z , etc. Dans cette conception, on caractérise les mathématiques plutôt par leurs activités que par des entités.

En même temps – ce qui est capital pour notre sujet – cela établit que les combinateurs portent sur des objets quelconques puisqu'ils ne forment pas d'opérateurs complexes mais affectent des arguments qui sont (en cette situation) de simples opérandes. La curryfication retrouve les effets, mais non les processus. Elle constitue en somme une fiction. Si les combinateurs ne portaient pas sur des objets quelconques, il faudrait un combinateur différent par combinaison d'objets distincts à produire ; soit par exemple $Cx_1x_2x_3 \sqsubseteq x_1x_3x_2$, $C'x'_1x'_2x'_3 \sqsubseteq x'_1x'_3x'_2$, etc. (ce que ne fait pas la LC). En particulier, si les combinateurs ne portaient pas sur des objets quelconques mais sur des opérateurs (et pour former d'autres opérateurs), il faudrait ou bien un combinateur différent par combinaison d'opérateurs à produire (et donc à nouveau par exemple $Cx_1x_2x_3 \sqsubseteq x_1x_3x_2$, $C'x'_1x'_2x'_3 \sqsubseteq x'_1x'_3x'_2$, etc.), ou bien que les combinateurs portent sur des opérateurs quelconques (concept que la LC n'emploie pas).

On a l'habitude de dire que tout se passe comme si X portait sur $x_1x_2\dots$ pour obtenir Q , alors que, au plan du calcul, X porte sur x_1 , puis Xx_1 sur x_2 , etc. En fait, l'objectif combinatoire (former à partir d'une combinaison associée à gauche, donc sans parenthèses (dite « $\cap$ suite $\cap$ », $x_1x_2\dots$), parce qu'elle correspond aux données les plus simples, une combinaison quelconque Q des éléments de cette suite, donc pouvant associer

à droite), impose de dire au contraire que tout se passe comme si, au plan du calcul, X portait sur x_1 , puis Xx_1 sur x_2 , etc., alors que X porte sur $x_1x_2\dots$ pour obtenir Q .

4. Selon la conception « λ -logique de l'opérateur
- λ
- », on est obligé de modifier les règles de réduction pour les appliquer. Dans
- $\mathbf{B}xyz$
- , par exemple, on parle des opérateurs
- x, y
- et de l'opérande
- z
- , alors que la règle se rapporte à trois éléments de même statut. On interprète comme si on avait, en notant (pour expliciter les concepts employés)
- f, g
- , etc. les opérateurs affectés, et
- x, y
- , etc. les opérandes du dernier opérateur complexe formé à l'aide du combinateur, respectivement
- $\mathbf{B}fg$
- et
- $\mathbf{C}f$
- :

$$\mathbf{B}fgx \equiv f(gx)$$

$$\mathbf{C}fxy \equiv fyx$$

Il ne s'agit plus des règles telles qu'elles sont données, comme on le voit par le fait qu'on pourrait décomposer différemment. Au lieu de lire $\mathbf{B}xyz \equiv x(yz)$ comme $\mathbf{B}fgx \equiv f(gx)$, on pourrait lire $\mathbf{B}fxy \equiv f(xy)$. $\mathbf{B}f$ formerait de xy un seul argument, $\mathbf{B}$ formerait une fonction unaire à partir d'une fonction binaire, ce qui ne serait pas davantage la règle, mais qui ne le serait pas moins cependant.

CONFIRMATION HISTORIQUE

Peut-être objectera-t-on qu'il arrive à Curry d'écrire sur le modèle qui explicite la conception « λ -logique de l'opérateur

 λ ». Dans [6], il donne notamment, p. 169, le schéma général d'une règle de réduction pour un combinateur de la forme λ_n de la manière suivante :

$$\lambda_n f g_1 g_2 \dots g_n x \equiv f(g_1 x)(g_2 x) \dots (g_n x)$$

(au symbole près de réductibilité) qui généralise λ , tel que $\lambda f g x y \equiv f(gx)(gy)$, pour les fonctions de n variables. λ écrit aussi $\mathbf{C}fxy$ et fyx , $\mathbf{B}fgx$ et $f(gx)$, p. 152.

D'autre part, il indique que chez Schönfinkel et dans sa propre thèse [3], les combinateurs sont interprétés comme des fonctions applicables « λ d'autres fonctions, y compris à elles-mêmes, [...] $\mathbf{C}$ transforme une fonction à n -places pour $n \geq 2$ en une fonction similaire avec ses deux premiers arguments interchangés» [7], p. 10-11. « $\mathbf{K}$, lorsqu'il est appliqué à une constante, la transforme en une fonction dont la valeur est cette constante pour tout argument, et lorsqu'il est appliqué à une fonction, la transforme en une fonction qui a un argument fictif au début», *ibid.*

On pourrait donc estimer la lecture « λ -logique de l'opérateur

 λ » validée par Curry lui-même et donc admises les formulations de la forme

$$Xf_1 \dots f_n a_p \dots a_1 \equiv b_1 \dots b_q$$

Pourtant, cela est donné par Curry sous la réserve expresse que la distinction entre les $f_1, \dots, f_n$ et les $a_p, \dots, a_1$ ne soit pas formelle. Il a toujours insisté en effet sur l'importance de se référer à des objets quelconques, même s'il admet des usages de la LC

où on s'occupera seulement de combinaisons entre objets spécifiés. On va le rappeler par quelques textes.

Les énoncés concernent «Des objets non spécifiés» [6], p. 13. «Les choses assignées [aux objets de la morphologie] peuvent être n'importe quelle sorte d'objets au sujet desquels on peut faire des énoncés : symboles (ou combinaisons de symboles), qualités, nombres, idées, choses manufacturées, naturelles», p. 20, et même des propositions [7], p. 9⁴⁰. «Les combinaisons seront souvent telles qu'elles sont spécialement significatives quand une ou plusieurs des variables sont – du point de vue de l'interprétation, bien sûr – elles-mêmes des fonctions», p. 151. Les x, y , etc. dans les règles de réduction sont des variables non spécifiées ; f, g, h , etc. seront employées quand la variable est pensée comme une fonction, a, b, c , etc. quand elle est pensée comme une constante. «Mais il n'y a pas de signification formelle à ceci, car toutes les variables sont des indéterminées [ou des métavariations] et sont formellement semblables», *ibid.*

Curry va jusqu'à admettre intéressant qu'on développe «Des systèmes partiels dans lesquels seules les combinaisons qui sont interprétables de quelque point de vue sont admises». C'est le cas d'un système proposé par Quine où «Les opérateurs [...] sont appliqués directement aux prédicats» et à rien d'autre [5], p. 131. Mais Curry ajoute qu'on limite alors les moyens proprement combinatoires dont on pourrait disposer. Bref, *c'est parce que la LC est une logique de l'objet quelconque qu'elle peut être à plus forte raison une logique de l'objet particulier qu'est un opérateur (au sens 1 ou 2).*

On comprend pourquoi, à lire Schönfinkel, ce point de vue peut ne pas apparaître, soit qu'il ne soit pas le sien, soit qu'il se trouve occulté par la manière dont l'auteur introduit ses «Fonctions particulières». Il ne les définit pas toutes sur une même suite. Il ne donne pas toujours l'effet de ses fonctions particulières de manière abstraite, mais en rapport avec un exemple ou plutôt en rapport avec un type d'exemple. Il écrit $Ix = x$, mais par exemple $C\lambda xy = \lambda yx$, $B\lambda\lambda x = \lambda(\lambda x)$, $S\lambda\lambda x = (\lambda x)(\lambda x)$, à la notation près des majuscules. Il est clair que Schönfinkel écrit $C\lambda xy$ plutôt que $Cxyz$, pour montrer que l'intérêt de C sera en général de permuter les arguments d'une fonction. Cela, cependant, accredit l'idée que C se définit en portant sur des fonctions. Curry écrit, au contraire (dès 1929), $Sxyz = xz(yz)$, $Kxy = x$, etc. sur des suites comparables où ne figurent que des objets quelconques («Any entities whatever», [2], p. 370). D'autre part, il ébauche des règles de formation, alors que Schönfinkel laissait intuitive la construction des expressions. Elles contribuent à dégager clairement la notion d'objet quelconque.

En conclusion

- Les objets figurant dans les règles de réduction ne sont pas des opérateurs, mais sont des objets quelconques ; les règles ne supposent pas la curryfication d'opérateurs, mais seulement une association à gauche d'objets quelconques

⁴⁰ On doit distinguer évidemment entre une proposition-objet et une proposition-énoncé portant sur la première. Dopp remarque qu'une proposition-objet, «Une sorte de substantif dénotant une éventualité», par exemple celle que Pierre soit blond, devrait se formuler par une infinitive, comme c'est possible en latin, «Pierre être blond» [12], p. 577, note 27.

- D'autre part, tout en disant qu'il n'y a pas d'opérateurs absolus ni d'opérandes absolus, la LC s'en prend surtout à l'idée qu'il y aurait des opérandes absolus, au fond des épiphénomènes (des objets sans efficace). On entend donc chercher en quoi un objet qu'on tient habituellement pour inerte possède une activité. C'est pourquoi il est caractéristique du style combinatoire qu'un objet qui n'est pas classiquement un opérateur₁ (tel qu'un entier qui peut apparaître comme résultat mais n'en produit pas lui-même, ou une valeur de vérité), placé devant un autre « joue le rôle d'un opérateur », et qu'un objet, placé après un autre, « joue le rôle d'un opérande », selon les expressions de J.-P. Desclés, dans [10], p. 69. L'application permet, en effet, « de fonctionnaliser des objets tout à fait arbitraires » [9], p. 20.
- Pourtant, il serait inexact de dire sans plus que la LC promeut en opérateurs des objets absolus, qu'elle s'adresse donc, dès qu'elle intervient, à des opérateurs et qu'on retrouve par là en sous-main la conception « logique de l'opérateur ». Il ne faut pas oublier que traiter en LC un objet jusque là « absolu » comme un opérateur ne le rend tel qu'en certaines occurrences ; il est essentiel au point de vue combinatoire qu'il demeure « objet absolu », opérande en certaines autres. Ainsi **1**, **2** sont des combinateurs (IV, premier exemple) mais dans **12** l'objet **2** est un simple opérande de **1**. On n'en revient pas à la distinction pour laquelle il y a par nature des opérateurs et des opérandes absolus, en se bornant à élargir le cercle des premiers. Un élément demeure traité comme objet, c'est-à-dire tantôt agissant, tantôt pâtissant. Le mouvement qui permet à des opérandes de devenir opérateurs est solidaire de celui qui contraint les opérateurs à n'être dans certains cas que des opérandes, même si on retient moins cette dernière mesure (liée intuitivement à une disqualification !). En LC, les rôles doivent s'échanger. C'est dire qu'on continue à ne s'adresser qu'à des objets *non marqués* (au sens linguistique du mot, des unités aussi neutres que possible).
- Il est ambigu de dire que x dans (xy) – ou, pour être plus exact, x dans $(x \square y)$ – est un opérateur. D'abord parce qu'on n'est pas contraint de mettre en premier argument de * ce qui est classiquement un opérateur (on l'a vu sur nos exemples), ensuite parce qu'on ne rend pas non plus opérateur au sens classique (et donc notamment dans un contexte classique de traitement) un objet qui classiquement ne serait pas un opérateur (ce qui serait un peu magique). On l'appréhende toujours par rapport à une morphologie combinatoire (c'est dans un calcul propositionnel remodelé que des expressions comme **1pqmn** $\square$ **pmn**, **1001** $\square$ **01**, deviennent admissibles). Bref, on doit prendre à la lettre les expressions de J.-P. Desclés « joue le rôle d'un opérateur », « joue le rôle d'un opérande », c'est-à-dire, justement, sans être comme tels ni l'un ni l'autre.
- Même en tant qu'on dote un objet d'un rôle d'opérateur, cette propriété ne lui est que prêtée, car on ne pourrait la lui donner qu'informellement, par une relecture intuitive de son statut. Il reste à établir formellement que sa catégorie est bien celle (ou l'une de celles) d'opérateur.

6. LA THÉORIE DE LA FONCTIONNALITÉ

Pour expliquer comment la LC parvient à distinguer formellement les objets qu'elle refuse de distinguer de manière informelle et d'entrée de jeu, pour expliquer, autrement dit, comment elle rejoint des objets particuliers à partir de ses objets quelconques, on va rappeler quelques données de sa partie illative. Pour ce faire, ajoutons à LC_1 :

- Un nombre donné d'éléments de cinquième espèce : $\Box_1, \Box_2, \Box_3, \dots, \Box_n$.
- Un élément de sixième espèce, noté F .

et les règles de formation :

- Un élément de cinquième espèce est une catégorie.
- Si $\Box$ et $\Box$ sont des catégories, $F\Box\Box$ est une catégorie.
- Rien d'autre n'est une catégorie.
- Si x est un objet et si $\Box$ est une catégorie, $\Box x$ est un énoncé.

On obtient ainsi un langage, disons LC_3 , nommé «Théorie de la fonctionnalité» (ici dans une version limitée). $\Box, \Box$ (ainsi que $\Box \Box$) sont des metavariables de catégories. Un énoncé de forme $\Box x$ attribue à x la catégorie $\Box$. F est un opérateur préfixé binaire. Il n'exige donc pas de parenthèses, mais on écrit évidemment $F\Box\Box(xy)$. Il n'est pas un combinateur. Il n'a donc pas de règle de réduction, mais on lui joint deux règles de déduction, respectivement d'introduction et d'élimination de F .

$$\begin{array}{ll} \text{(intro)} & \frac{\Box(xy) \quad \Box y}{F\Box\Box x} \qquad \qquad \qquad \text{(élim)} \quad \frac{F\Box\Box x \quad \Box y}{\Box(xy)} \end{array}$$

En bref, $F\Box\Box x$ signifie « $\Box$ appliqué à un $\Box$ forme un $\Box\Box$ (comme le prédicat unaire f appliqué à l'individu i_1 forme la proposition $f i_1$). Quant à (élim) il est clair que si « $\Box$ appliqué à un $\Box$ forme un $\Box\Box$, il faut que y soit un $\Box$ pour qu'on puisse appliquer x à y , et dans ce cas on forme un $\Box$ »⁴¹.

On peut déterminer notamment la catégorie à laquelle appartient un combinateur, à partir des conditions suivantes :

- (i) On pose que les deux membres d'une règle de réduction appartiennent à la même catégorie (ce qui est une mesure formelle convenable puisque le premier membre se réduit au second)»

⁴¹ On trouve aussi couramment le mot «Type» à la place du mot «Catégorie» (que nous préférons pour éviter une confusion avec la notion russellienne). Le mot «Espèce» est employé seulement pour désigner une collection d'éléments syntaxiquement groupés et utile pour formuler les règles de formation. Une «Catégorie» entend plus fortement rassembler des objets de même nature, ou, si l'on préfère, de même rôle (et c'est, évidemment, un problème de savoir comment se correspondent les espèces et les catégories). Le texte formel que nous donnons comprend une catégorie comme une qualité («Être une proposition», «Être un connecteur», etc.) et non directement comme un «Ensemble» d'éléments apparentés («Appartenir à l'ensemble des propositions», «Appartenir à l'ensemble des connecteurs», etc.), même si la formulation ensembliste apparaît par commodité dans le texte informel.

- (ii) La métavariable $\square$ peut être employée (et non une catégorie précise $\square_i$, pour $i = 1, 2, \dots$), car on s'adresse à des objets quelconques. On notera qu'on ne se donne pas une catégorie des opérateurs, et précisément parce qu'il ne s'agit pas de cela. Toutefois $\square$
- (iii) Si on employait la même catégorie $\square$ pour tous les objets intervenants, on supposerait qu'il suffit de vouloir s'adresser à des objets quelconques pour qu'on soit fondé à le faire et donc on en reviendrait à la position condamnée qui, selon la LC, transporte dans les logiques usuelles des notions informelles en leur donnant droit de cité dans les systèmes. On va donc accepter de faire intervenir $\square$ (puis $\square \square$), quand, pour une application de (intro), rien ne dit formellement que l'objet qu'on veut quelconque est indiscernable d'un objet invoqué par ailleurs ($\square$ est une catégorie éventuellement différente de $\square$) $\square$
- (iv) On réduira au minimum le nombre des catégories requises, en employant, sur les deux membres de la règle de réduction, la règle (intro), de toutes les façons que permet celle-ci.

On procède donc en gardant aussi peu de catégories que possible et *afin que les différences ontologiques émergent inévitablement et formellement d'une indifférenciation originale*. Ce qui a toujours été l'objectif de la LC se trouve bien atteint dans son propre domaine. Ainsi :

Comme $\mathbf{K}xy \square x$, il est clair qu'on aura $\square(\mathbf{K}xy)$, $\square x$, par (i), mais $\square y$ car rien ne dit formellement que la catégorie de y est celle de $\mathbf{K}xy$ et de x . On obtient facilement alors $F\square F\square\square\mathbf{K}$ (la catégorie de $\mathbf{K}$ est $F\square F\square\square$).

Dans le cas de $\mathbf{W}xy \square xyy$, on aura $\square(\mathbf{W}xy)$, $\square(xyy)$, par (i), mais $\square y$ car rien ne dit formellement que la catégorie de y est celle de $\mathbf{W}xy$ et de xyy . Cependant, la catégorie de x n'a pas à être nouvelle, car on peut appliquer (intro) sur la résultante et obtenir $F\square F\square\square x$. Il vient alors facilement $FF\square F\square\square F\square\square\mathbf{W}$ ⁴².

Certes, si la théorie donnée (qui est consistante) supprime la distinction entre les objets et les catégories, en traitant les secondes comme des objets quelconques, elle devient inconsistante, mais on ne doit pas en conclure, évidemment, que la LC ne pouvait pas porter sur des objets quelconques, puisque c'est son deuxième objectif (classer des objets d'abord indistincts en catégories) qui impose qu'on fasse appel à des entités hétérogènes.

Toutefois, pour que cette approche remplisse la deuxième partie du programme de la LC, il ne suffit pas qu'elle soit au service des combinateurs, il faut qu'elle soit au service de la logique usuelle, qu'il y ait donc des cas où celle-ci attribuait intuitivement une catégorie (à juste titre ou non) et où la logique combinatoire confirme ou réfute cette attribution formellement. Or, après avoir adapté la morphologie de LC_3 aux besoins des logiques usuelles (l'objectif à moyen terme de la LC pure), il ne semble pas qu'on ait obtenu des résultats très significatifs.

⁴² Nous renvoyons à notre analyse dans [16], p. 93-94, pour le détail de cette preuve.

Dans la LC pure, dite *sans types*, on peut appliquer tout x à tout y puisque xy a toujours un sens «comme concept» (c'est-à-dire donne à penser) et puisqu'il ne s'agit de rien d'autre, mais si on entend attribuer un sens particulier à x , on limitera l'applicabilité de x à y (en employant la logique combinatoire dite *typée*). Seule sera sensée une application de P (l'implication) à des éléments qui soient des propositions. Et en effet, on peut démontrer par exemple que Ppq est une proposition et que PNN ne l'est pas. En supposant connues les catégories de p , de q et de P , il vient, en appliquant deux fois (élim) :

$$\frac{\frac{F \square F \square \square P \quad \square p}{F \square \square (Pp)} \quad \square q}{\square (Ppq)}$$

Il est facile de voir que si on essaie d'obtenir PNN (au lieu de Ppq), où $F \square \square N$ est déduit par ailleurs, la démonstration ci-dessus ne peut plus s'effectuer. On a :

$$\frac{F \square F \square \square P \quad F \square \square N}{\quad}$$

Pour appliquer (élim), il faut que $\square$ désigne à gauche et à droite une même catégorie. Or il est $\square$ à gauche et $F \square \square$ à droite, et il n'est pas possible qu'on ait $\square = F \square \square$. PNN n'est donc pas susceptible de recevoir une catégorie de manière cohérente. On dit que PNN n'est pas stratifiable.

Néanmoins, si l'on peut admettre qu'on sache que $\square p, \square q$, l'emploi de $F \square F \square \square P$ pose déjà un problème. Si la catégorie de P a été obtenue par (intro), on a eu :

$$\frac{\frac{\square (Ppq) \quad \square q}{F \square \square (Pp)} \quad \square p}{F \square F \square \square P}$$

La règle d'élimination inversant, comme il est naturel, la règle d'introduction (pour que ce soit bien le même élément qu'on manipule), on a dû prendre pour prémisse ce qu'on obtient maintenant comme conséquence. Si au contraire la catégorie de P n'a pas été obtenue par (intro), pour éviter cette issue, comment peut-on dire qu'on a obtenu *formellement* des résultats sur Ppq et sur PNN ? On pourrait obtenir par (intro) $F \square F \square \square A$, par exemple, où A est la disjonction, puis ajouter que P , foncteur binaire comme A , doit avoir la même catégorie que A , et enfin prouver par (élim) qu'on a $\square (Ppq)$. La deuxième étape, certes, est informelle, mais la procédure d'ensemble l'est moins que la simple construction intuitive (ou que la postulation) de $F \square F \square \square P$, et il est clair que la formalisation est toujours une notion relative. Au demeurant, on peut rester insatisfait d'une telle démarche.

Mais nous voudrions montrer sur un point plus stratégique pourquoi la LC paraît décevoir les attentes. On a travaillé d'abord avec un système admettant sans restriction l'application d'un objet à tout autre. Le problème de savoir si une limitation formellement justifiée de l'application peut intervenir, ainsi que celui de savoir si tel assemblage reçu

comme une proposition l'est en effet, se posent clairement à propos du paradoxe de Russell⁴³. Le programme combinatoire est donc au pied du mur. Il devrait conduire à montrer :

- Ou bien qu'on traite à tort FF comme une proposition, ce qui ferait apparaître indûment la contradiction $\square$
- Ou bien qu'on traite avec raison FF comme une proposition, et il faudrait expliquer ce qui se passe dans un tel cas⁴⁴.

Pourtant, à notre connaissance, la LC n'établit rien sur le caractère propositionnel de FF (ni rien sur le caractère de propriété de F). Au lieu de cela, on trouve une analyse par Curry qui montre que FF peut donner lieu à une formulation délibérément non propositionnelle et qui fait donc échapper FF à la contradiction. Mais on se borne ainsi à éviter le problème, peut-on (ou peut-on aussi) accepter FF comme proposition ?

Cela peut jeter le doute sur le bien fondé du programme, notamment sur la nécessité d'une logique pure de l'objet quelconque (ou d'une «logique de l'opérateur» qui ne serait pas mieux placée à cet égard). Bien sûr, la LC pure n'est pas seulement préparatoire à une théorie illative. Elle a déjà des sous-produits utiles (par exemple l'expression sans variables des formalismes usuels). Cependant, pour les évaluer, il faut savoir si elle pouvait admettre de prendre une fonction pour argument, de pratiquer l'auto-application, etc. et la théorie de la fonctionnalité peut sembler destinée à le montrer. Mais on voit qu'il n'en est rien. En fait, on avait d'ailleurs récusé par avance que PNN puisse voir son sens remis en cause par quelque technique formelle de rejet. En vertu du principe de pansémie, PNN, PP, etc. ont un sens large, même s'ils n'ont pas de sens propositionnel. Être dénué de sens propositionnel n'empêche pas, mais suppose au contraire, la possession d'un sens plus large et qui est inaliénable, formellement invincible. Que devient dès lors le programme de soumettre à formalisation les notions prélogiques importées clandestinement depuis la pensée ordinaire ? Il ne peut pas porter en somme sur les présupposés combinatoires. On n'a pas de prise formelle sur l'auto-application, par exemple. On peut établir en effet que xx est inacceptable, quand il s'agit d'un x quelconque, car selon (élim)

$$\frac{F \square \square x}{\square x}$$

⁴³ Soit f une propriété quelconque, f aura par définition la propriété F (*être imprédicable*) ssi elle ne s'applique pas à elle-même (comme dans [6] on distingue ici, par son italique, la propriété F , de l'opérateur F). On a donc (en notation combinatoire) :

(1) $Ff =_{\text{df}} N(ff)$, où N désigne la négation.

Si on substitue à f dans (1) la valeur particulière F , on obtient :

(2) $FF =_{\text{df}} N(FF)$,

où une contradiction apparaît entre les deux membres. Or, il semble y avoir intuitivement pertinence de la distinction (sauf rectification, «être rouge» est imprédicable, «être abstrait» est prédicable).

⁴⁴ Si FF n'appartient pas à la catégorie des propositions ou si F lui-même n'appartient pas à celle des propriétés, il n'y a pas de contradiction, disait Curry en 1930, [3], p. 511.

On ne peut pas déduire la catégorie de xx , car il faudrait que dans $\Box x, \Box = F\Box\Box$, ce qui est impossible ; xx est non stratifiable. **KK**, de même, n'est pas stratifiable puisque, selon (élim), où la première prémisse est donnée, il viendrait :

$$\frac{F\Box F\Box\Box K \quad \Box K}{F\Box\Box(KK)}$$

alors que $\Box$ n'est pas la catégorie (qui est unique) de **K**⁴⁵.

Dans la LC pure, cependant, qui n'a pas de catégories, rien ne peut s'opposer par hypothèse à xx , quel que soit x . Quant à savoir pourquoi on peut y éviter les catégories, la réponse est donnée par le principe de pansémie qui n'est pas soumis à une approbation logique mais qui transcende les formalismes.

On raisonne souvent, pourtant, comme si la consistance de la LC pure donnait cette approbation logique, établissait formellement qu'un opérateur peut s'appliquer à lui-même ou à un autre (et en rendant au moins admissible la conception «logique de l'opérateur»). Curry lui-même écrit : «Puisque des inconsistances (au sens de Post) ne peuvent pas se produire, les types ne sont pas nécessaires» [8] p. 94⁴⁶.

Et pourtant, formellement, la consistance n'établit pas que des opérateurs *quelconques* f et g peuvent *ès qualités* se combiner sans dommage en ff, fg, gf, gg . Elle n'établit pas le principe de libre application. Elle établit seulement que f, g peuvent s'appliquer entre eux si on fait abstraction du fait que par ailleurs ils seraient des opérateurs, ou qu'ils peuvent s'appliquer entre eux parce qu'ils en ont, justement, la *particularité*⁴⁷.

Bref, *ce n'est pas du tout «comme concept» (avec leur sens d'opérateur quelconque) qu'ils se combinent de manière consistante*. Que tout concept comme tel convienne à tout concept comme tel – la norme fondamentale de la LC – ne peut se défendre que par une analyse extra-formelle («En prose», selon une expression de Wittgenstein). Ce qu'on soutient c'est qu'une activité mentale trouvera toujours à s'exercer, si on s'en donne la peine, à partir d'un assemblage de concepts qui semble d'abord hétéroclite (comme le surréalisme a montré qu'un esprit vif tire toujours parti d'un «cadavre exquis»). Ainsi, soit PNN (une expression qui ne peut pas recevoir une catégorie de manière cohérente, comme on l'a vu, et qui pourrait donc sembler un simple rebut). Or, même en pensant P comme le concept d'implication et N comme le concept

⁴⁵ **KK** est stratifiable dans une version de la théorie où plusieurs catégories sont assignées à un combinateur (voir [19], p. 82).

⁴⁶ On peut montrer en effet qu'en LC pure deux expressions combinatoires quelconques ne sont pas réductibles l'une à l'autre, ou obtenir ce résultat à partir du théorème de Church-Rosser, ou encore construire un modèle (comme le fait D. Scott).

⁴⁷ Ainsi, dans le modèle de Scott, s'agit-il d'opérateurs, mais très précis et non «quelconques». On parvient à étendre, au fond, à tous les opérateurs d'un domaine particulier bien choisi l'auto-application, notamment, qui semble raisonnable, comme dit [20], p. 45, pour **I, K, S**, puisqu'on a $\mathbf{I}x \Box \dots \Box x \Box \mathbf{K}Kxyz \Box \dots \Box y ; \mathbf{S}Sxyz \Box \dots \Box yz(xyz)$. Comme d'habitude, le modèle établit qu'il existe un cas favorable. Il ne fait pas admettre, ici, que $\Box x (xx)$.

de négation, une activité mentale peut s'exercer sur PNN. P et N se combinent comme concepts (avec leur sens d'opérateurs) et PNN apparaît, quand on l'applique à p , comme une abréviation sensée de l'expression classique $P(Np)(Np)$. Ici, PNN est rendu tel informellement, mais on dispose aussi, formellement, de moyens pour en prouver la possibilité. Le passage intuitif et décisif de PNN à $P(Np)(Np)$ donne lieu à une justification stricte par le combinateur $\mathbf{B}^3\mathbf{W}(\mathbf{B}^2\mathbf{C}(\mathbf{B}\mathbf{B}(\mathbf{B}\mathbf{B})))$ affectant PNN^{48} . On a, en effet, $\mathbf{B}^3\mathbf{W}(\mathbf{B}^2\mathbf{C}(\mathbf{B}\mathbf{B}(\mathbf{B}\mathbf{B})))\text{PNN}p \sqsubseteq \mathbf{B}^2\mathbf{C}(\mathbf{B}\mathbf{B}(\mathbf{B}\mathbf{B}))\text{PNN}pp \sqsubseteq \mathbf{B}\mathbf{B}(\mathbf{B}\mathbf{B})\text{PN}p\text{N}p \sqsubseteq \mathbf{B}\mathbf{B}p(\text{N}p)\text{N}p \sqsubseteq P(\text{N}p)(\text{N}p)$. (On obtiendrait évidemment des expressions comparables si on exprimait PNN en langage λ , $\mathbf{1}, \mathbf{0}$, ou en langage $\mathbf{H}, \mathbf{1}, \mathbf{0}$).

De ce point de vue, la construction d'un modèle, comme notre reprise de PNN, ou comme l'interprétation d'un «cadavre exquis», c'est d'abord et surtout une confirmation qu'il y a matière à penser (même formellement) dans l'insolite.

En ce qu'elle soutient primitivement cela – le principe de pansémie – la LC n'est pas même une logique (consistante ou non) de l'objet quelconque. Elle en est plutôt (fructueuse ou non) une *ontologie*.

BIBLIOGRAPHIE

- [1] CHURCH A., *Introduction to mathematical logic*, Princeton University Press, 1956.
- [2] CURRY H. B., «An analysis of logical substitution», *The American Journal of Mathematics*, 51, 1929, p. 363-384.
- [3] CURRY H. B., «Grundlagen der kombinatorischen Logik», *ibid.*, 52, 1930, p. 509-536, p. 789-834.
- [4] CURRY H. B., *Outline of a formalist philosophy of mathematics*, Amsterdam, North-Holland Publishing Company, 1951.
- [5] CURRY H. B., «The elimination of variables by regular combinators», *The critical approach to science and philosophy* (M. Bunge, ed.), in honor of Karl R. Popper, Glencoe, Illinois, The free Press, 1964, p. 127-143.
- [6] CURRY H. B., FEYS R., *Combinatory logic*, I, North-Holland Publ. Comp., 1968, 1^e éd. 1958.
- [7] CURRY H. B., HINDLEY J. R., SELDIN J. P., *Combinatory Logic*, II, North-Holland Publ. Comp., 1972.
- [8] CURRY H. B., «Some philosophical aspects of combinatory logic», in J. Barwise, H. J. Keisler and K. Kunen, eds, *The Kleene symposium*, North-Holland Publ. Comp., 1980, p. 85-101.
- [9] DESCLÉS J.-P., «De la notion d'opération à celle d'opérateur ou à la recherche de formalismes intrinsèques», *Mathématiques et Sciences humaines* 76, 1981, p. 5-32.

⁴⁸ Nous appliquons ici le combinateur à PNN (et non à PN) pour garder l'expression elle-même qu'on veut élucider.

- [10] DESCLÉS J.-P., «Théorème de Church-Rosser et structuration des langues naturelles», *Mathématiques Informatique et Sciences humaines* 103, 1988, p. 67-92.
- [11] DESCLÉS J.-P., *Langages applicatifs, langues naturelles et cognition*, Paris, Hermès, 1990.
- [12] DOPP J., «La formalisation de la logique», *Revue philosophique de Louvain* 50, 1952, p. 533-586.
- [13] DUBARLE D., *Initiation à la logique*, Paris, Gauthier-Villars, Louvain, E. Nauwelaerts, 1957.
- [14] FREY L., «Langages logiques et processus intellectuels», *Les modèles et la formalisation du comportement*, colloques internationaux du CNRS, Paris, Éd. du CNRS, 1967, p. 327-345.
- [15] GINISTI J.-P., «Présentation de la logique combinatoire en vue de ses applications», *Mathématiques Informatique et Sciences humaines* 103, 1988, p. 45-66.
- [16] GINISTI J.-P., *La logique combinatoire*, Paris, Presses Universitaires de France, 1997, coll. Que sais-je ?, n° 3205.
- [17] GONSETH F., *Qu'est-ce que la logique*, Paris, Hermann, 1937.
- [18] GRIZE J.-B., «Quelques problèmes logico-linguistiques», *Mathématiques et Sciences humaines* 35, 1971, p. 43-50.
- [19] HINDLEY J. R., LERCHER B., SELDIN J. P., *Introduction to combinatory logic*, Cambridge University Press, 1972.
- [20] HINDLEY J. R., SELDIN J. P., *Introduction to combinators and λ -calculus*, Cambridge University Press, 1990, 1^e éd. 1986.
- [21] KEARNS J. T., «Combinatory logic with discriminators», *The Journal of symbolic logic*, vol. 34, number 4, déc. 1969, p. 561-575.
- [22] KEARNS J. T., «Fully explicit deductive systems», 35-54, in *To H. B. Curry : Essays on combinatory logic, lambda calculus and formalism*, J. P. Seldin, J. R. Hindley (eds), Academic Press, Londres, New York, 1980.
- [23] MERLEAU-PONTY M., *Sens et non-sens*, Paris, Nagel, 1961.
- [24] PIAGET J., *Traité de logique*, Paris, A. Colin, 1949.
- [25] QUINE W. V., «A reinterpretation of Schönfinkel's logical operators», *Bulletin of the American Mathematical Society* 42, 1936, p. 87-89.
- [26] ROSENBLOOM P. C., *The elements of mathematical logic*, New York, Dover Books, 1950.
- [27] RÉVÉSZ G. E., *Lambda-calculus, combinators, and functional programming*, Cambridge University Press, 1988.
- [28] ROSSER J. B., *Deux esquisses de logique*, trad. française, Paris, Louvain, Gauthier-Villars, Nauwelaerts, 1955.
- [29] SCHÖNFINKEL M., «Über die Bausteine der mathematischen Logik», *Mathematische Annalen*, 92, 1924, p. 305-316. Trad. par G. VANDEVELDE, «Sur les éléments de construction de la logique mathématique», Analyse et notes par J.-P. GINISTI, *Mathématiques Informatique et Sciences humaines* 112, 1990, p. 5-26.

ERRATA D'EDITION

Dans l'article de Jean-Pierre Ginisti, «La logique combinatoire : logique de l'objet quelconque ou logique de l'opérateur», *Math. & Sc. hum. / Mathematics and Social Sciences* (41^e année, n° 162, 2003, p. 45-92).

- p. 45, en haut, lire, à propos de l'année : « 2003 » (et non 2002).
 - p. 45, dans « SUMMARY », lire : « formalization » (avec un « z »).
 - p. 45, dans KEYWORDS, lire : «Currying » (pour « Curryfication »).
 - *passim*, à la place du symbole « Q », lire partout : $\mathcal{Q}$
 - *passim*, à la place des chiffres romains, lire les chiffres arabes correspondants et qui se rapportent aux titres des six grandes sections.
 - p. 48, note 4, lire : « $S_{x_1 x_2 x_3} \sqcap x_1 x_3 (x_2 x_3)$ ».
 - p. 68, sur les symboles « v » et « ν », lire partout : « v », sauf quand le symbole est placé devant des parenthèses où il a le sens « la valeur de » et s'écrit « ν » (par exemple pour $\nu(P10) = 0$).
 - p. 68, ligne 18, lire : « $F\forall v_2 \dots v_n$ ne prend pas ».
 - p. 73, ligne 32, lire : « domaine fondamental » (sans le « e » final).
 - p. 74, *passim*, lire partout : « curryfication » (avec un « y »).
 - p. 74, en c), lire : « Avec curryfication initiale (ou de tête) ».
 - p. 74, lignes 18-19, lire : « une fonction F telle que $F(x_1, x_2, x_3, \dots, x_n)$ ».
 - p. 74, ligne 20, lire : « $n - 1$ arguments (ceux qui restent à droite) ».
 - p. 74, en d), lire : « $\mathbf{B}$ est une fonction binaire qui s'applique à x, y en formant $\mathbf{B}(x,y)$ qui est une fonction unaire qui s'applique à z , en obtenant $x(yz)$, soit $\sqcap(\mathbf{B}(x, y))z \sqcap x(yz)$ ».
 - p. 86, ligne 4, lire : « Toutefois, » (et non « Toutefois ; »).
 - p. 89, ligne 22, lire : « comme concepts » (et non « comme concept »).
-