MATHÉMATIQUES ET SCIENCES HUMAINES

JEAN-PIERRE GINISTI

Présentation de la logique combinatoire en vue de ses applications

Mathématiques et sciences humaines, tome 103 (1988), p. 45-66

<http://www.numdam.org/item?id=MSH_1988__103__45_0>

Math. Inf. et Sci. hum. (26e année, n°103, *1988*, pp.45-66).

PRESENTATION DE LA LOGIQUE COMBINATOIRE EN VUE DE SES APPLICATIONS

Jean-Pierre GINISTI[1]

Nous nous proposons dans cet article d'introduire à la lecture (ou à l'écriture) de travaux qui utilisent la logique combinatoire dans les sciences de l'homme. La logique combinatoire sera présentée plutôt comme un langage formel que comme un système formel. Nous chercherons avant tout à exposer ses techniques et leurs enjeux, reléguant au second plan celles des investigations qui intéresseraient plus directement les mathématiques. Il s'ensuit notamment que certaines preuves ne seront fournies qu'intuitivement. Cet article n'a pas non plus pour objet, d'ailleurs, de présenter les applications elles-mêmes. Il veut seulement en donner les conditions de possibilité ou les schémas. Toutefois, comme la compréhension sérieuse d'une approche doit aller au-delà de l'assimilation de ses techniques, seront adjoints des éléments d'analyse conceptuelle, historique et même axiomatique. Les notes ajouteront des commentaires plus généraux qui suggèrent une deuxième exploitation possible de cette présentation : celle que pourrait en faire une philosophie du langage, naturel et formalisé.

UNE LOGIQUE SANS VARIABLES

La logique combinatoire a deux promoteurs : le logicien allemand Moses Schönfinkel lors d'une conférence de 1920 qui donna lieu à un article rédigé par Behmann et publié en 1924 *Uber die Bausteine der mathematischen Logik* ; le logicien américain Haskell B. Curry (mort en 1982) qui d'abord retrouva indépendamment certains résultats de Schönfinkel puis les développa très considérablement à partir de 1930, et qui est l'auteur, avec Robert Feys pour le premier volume, avec J.R. Hindley et J.P. Seldin pour le second, de l'ouvrage théorique de référence[2].

Telle qu'elle apparaît dans l'article de Schönfinkel-Behmann, la logique combinatoire est née d'une recherche pour réduire le nombre des éléments présents dans chaque catégorie primitive de symboles, en prolongeant la réduction obtenue par Scheffer au seul connecteur d'incompatibilité de tous les autres connecteurs, et au-delà, pour réduire le nombre des catégories primitives de

1 Professeur à l'Université Lyon III.

2 *Combinatory logic*, Amsterdam, North-Holland Publ. Comp., vol. I : 1e éd. 1958, 2e éd. 1968 ; vol. II : 1972.

symboles. Si on utilise la notation préfixée (dont fera usage aussi dans son domaine la logique combinatoire), c'est-à-dire qui place chaque opérateur devant son (ou ses) opérandes(s), quelque soit le nombre, supposé connu, de celles-ci, et en écrivant *Dpq* pour *p est incompatible avec q*, on exprimera par exemple *~p (non p)* par *Dpp*, qui en possède la table de vérité, *p.q* (*p* et *q*) par *D(Dpq) (Dpq)*, qui en possède la table de vérité, etc.. D'une manière semblable, il s'agit pour Schönfinkel d'économiser dans le calcul des prédicats, à l'intérieur de la catégorie des constantes, les quantificateurs $\forall$, $\exists$ au profit d'un connecteur qui est une extension de *D*. Mais il s'agit aussi pour lui d'économiser des catégories primitives de symboles en allant plus loin que Sheffer qui opérait seulement à l'intérieur de l'une des catégories. Or, comme les parenthèses ne sont que des signes de ponctuation (d'ailleurs éliminables sans introduire d'ambiguïtés, *D(Dpq) (Dpq)* devenant *DDpqDpq*), il demeure dans le calcul des propositions, si on s'en tient à lui pour exemple, deux catégories de symboles proprement dits, respectivement constituées des variables propositionnelles (*p* et *q* dans nos formules) et de l'unique constante *D*. Il y a d'abord un intérêt général, en effet, à éliminer sans perte une catégorie de symboles "du point de vue de la méthode", dit Schönfinkel au § 1, "qui est de tendre vers la plus grande unité de pensée possible", et un intérêt particulier à éliminer la catégorie des variables puisque celles-ci apparaissent comme de simples commodités d'expresssion[3]. En écrivant, par exemple, *(x + y)* on évitera la périphrase "un certain objet plus un objet identique ou non à celui que je viens de désigner par les mots *un certain objet*". Or, il semble aventureux de renforcer le rôle des variables en faisant correspondre au classement des symboles en constantes et variables un classement comparable de deux types d'entités. Si "*0*", "*1*", etc. désignent des nombres constants, il est hasardeux de supposer que "*x*", "*y*", etc. désignent semblablement des nombres (ou des quantités) variables, même si ce vocabulaire a été naguère utilisé, puisque les propriétés d'une quantité variable seraient, elles-mêmes, et au sens usuel, "variables" (voir les remarques de Tarski au début de son *Introduction à la logique*). Dire qu'une variable (liée) "parcourt un domaine", qu'une variable "tend vers *0*" ne se réfère pas à des grandeurs qui changent mais respectivement à un symbole qui doit désigner chaque objet du domaine, non seulement tel ou tel, et à un symbole qui se rapporte à une grandeur susceptible de prendre une valeur aussi petite qu'on le voudra[4].

S'il n'y a donc pas de réalités distinctes attachées à des constantes et à des variables ni à des emplois de symboles qu'on "maintient constants" dans un certain contexte ou qu'on "autorise à

3 Cette élimination des variables est intéressante aussi, ajoute Schönfinkel, "d'un certain point de vue philosophique, ou si l'on veut du point de vue esthétique", dans la mesure où une variable "a le caractère d'un simple concept auxiliaire, à vrai dire inadéquat à la nature constante, "éternelle" de l'énoncé logique". Si on met la réserve "ou si l'on veut du point de vue esthétique" au compte de la modestie du ton, bien que l'économie des moyens satisfasse aussi un critère dit "d'élégance", l'objectif est donné d'emblée comme étant "philosophique". Le même jeu de mots que dans la traduction française souligne dans le texte allemand la convenance des seules constantes à l'expression de "la nature constante de l'énoncé logique". Par élimination des variables, il s'agit bien en réalité de dégager ce dont parle l'énoncé, sans mentionner d'éléments subalternes, c'est-à-dire d'expliciter une ontologie (même si cette dernière se révèlera porter sur des actes plus que sur des êtres).

4 Certains travaux, à vrai dire, parviennent à donner aux référents des variables des propriétés (définies par les thèses où elles figurent, quand le langage est formel) qui se modifient. C'est le cas, par exemple, des systèmes propositionnels de Leśniewski qui ajoutent étapes par étapes, comme dans une théorie qui s'enrichit au cours du temps, à l'alphabet et aux axiomes initialement donnés, des éléments nouveaux et des thèses nouvelles, les règles de déduction, notamment celle de substitution, étant exprimées de telle sorte qu'ells s'appliquent à ces adjonctions successives. Il s'ensuit que les référents des variables ont des propriétés différentes selon l'étape considérée puisque les thèses qui les définissent ne sont pas les mêmes qu'antérieurement. Les expressions bien formées et les thèses nouvelles, en effet, qui s'introduisent à cette étape permettent des démonstrations supplémentaires. Qu'on juge condamnée ou non par de tels systèmes la philosophie des "vérités éternelles", les variables y demeurent, cependant, des commodités d'expression qu'on peut vouloir éliminer.

varier" dans un autre, il faut supprimer la catégorie de variable puisque c'est elle qui n'a pas de répondant ontologique direct[5].

La logique combinatoire entend donc obtenir des langages formels sans variables, mais il faut remarquer cependant : (1) qu'il ne s'agit pas d'interdire les variables (dites *intuitives*, ou *métavariables*) qui figurent dans les expressions métathéoriques portant sur le langage formel, en cause (dans les règles de formation ou de déduction, dans les métathéorèmes, etc.) mais seulement les variables qui formeraient une catégorie de symboles intérieurs à ce langage formel (2) qu'en donnant des moyens qui remplacent les variables on cherche à comprendre leur rôle et leur statut, et sans méconnaître la commodité qu'elles apportent dans beaucoup de cas aux traitements, (3) que la logique combinatoire a plus généralement pour objectif de reconstruire et d'asseoir les disciplines formelles sur des bases originales issues d'une analyse des facilités, des présupposés et des embarras propres aux logiques usuelles, notamment des règles de substitution, qui sont liées à l'usage des variables, et des paradoxes. Elle vise, exactement, à trouver des représentations, pour le langage qui est le sien (supposé clarifiant), de notions logiques et mathématiques. Par surcroît, elle obtient aussi des représentations de notions utiles dans les sciences de l'homme, principalement en linguistique et en psychologie. On peut exposer en six étapes le développement du programme de la logique combinatoire, en commençant par ce qu'on nommera *la théorie des combinaisons applicatives*.

1. THEORIE DES COMBINAISONS APPLICATIVES

Soit donc $a, b, c, d,...$ une séquence finie d'objets, supposés donnés, de nature quelconque. Par extension, cette séquence pourra n'avoir qu'un seul élément. Soit aussi une opération binaire, notée "$*$", employée de manière infixe et dite *application*. Un élément de la séquence peut être appliqué à un autre, par exemple, a peut être appliqué à b, on écrit alors $(a*b)$. Le sens le plus général d'une expression de cette forme est d'indiquer que l'adjonction du deuxième objet (celui de droite) au premier (celui de gauche) produit quelque effet. Cet effet peut être seulement de constituer de deux objets, à quelque égard, une totalité (effet de globalisation), mais l'opération d'application comportera en outre, comme on le verra, des interprétations beaucoup plus spécifiées, quand on quittera la théorie intuitive des combinaisons. On peut appliquer également le résultat d'un premier usage de "$*$", par exemple $(a*b)$, à un élément de la séquence, par exemple à c, on obtiendra alors $((a*b)*c))$, ou appliquer c à $(a*b)$, on obtiendra alors $(c*(a*b))$, expression tenue pour distincte de $((a*b)*c)$. On peut appliquer enfin le résultat d'un premier usage de "$*$" au résultat d'un deuxième usage de "$*$", par exemple $(a*b)$ à $(c*d)$ pour obtenir $((a*b) *(c*d))$: autrement dit, les expressions précédemment formées sont considérées comme des objets nouveaux (nous dirons *molécules*, par opposition aux objets de la séquence qui sont des "atomes") et les parenthèses traitent l'expression enclose comme un objet unique. Une expression entre parenthèses résultant de l'emploi de l'opérateur "$*$" entre deux opérandes (objets atomiques ou moléculaires) est nommée *combinaison applicative*. On dira souvent par ellipse *combinaison* puisque l'application est la seule opération considérée. Par extension, un élément unique sera dit aussi *combinaison*. On peut fournir une définition inductive du concept de combinaison applicative :

5 Bien sûr, on peut aussi s'interroger sur la nature des entités qui correspondent aux constantes mais le problème est très différent de celui qui porte sur les variables. Il semble naturel de considérer que ce qui est un objet doit être un, selon une expression de Leibniz, s'énoncer par une constante, même si la question de savoir quelle est la nature de cet objet demeure ouverte : existant réel dans un monde transspatial et éternel, existant mental, et comme état de conscience, ou comme idée d'un entendement divin, ou a priori de la pensée, ou identifiable à n'importe quelle réalité du monde physique qui produit le même effet, possède les mêmes propriétés, à des différences près, jugées dans un cas donné hors sujet, etc..

Soit une séquence donnée d'objets quelconques : *a, b, c,...*

(a) chaque élément de la séquence est (aussi) une combinaison applicative,
(b) si deux expressions e_1, e_2 sont des combinaisons applicatives (distinctes ou non) $(e_1 * e_2)$ est une combinaison applicative,
(c) rien d'autre n'est une combinaison applicative.

Sur la séquence *a,b* les premières combinaisons applicatives sont donc : *a* ; *b* ; *(a*a)* ; *(a*b)* ; *(b*a)* ; *(b*b)* ; *(a *(a*a))* ; *(a *(a*b))* ; etc.

Par abus de langage, on sous-entendra l'opérateur "*" : *(a*b)* deviendra *(ab)* ; *((a*b) * c)* deviendra *((ab)c)* . Par abus de langage aussi, on supprimera toute paire de parenthèses dont l'élément ouvrant apparaît sur la gauche d'une expression, par exemple *(ab)* deviendra *ab*,*((ab)c)* deviendra *(ab)c*, ou sur la gauche d'une expression déjà ainsi simplifiée : *(ab)c* deviendra *abc*, ou sur la gauche dans une sous-expression délimitée par une paire de parenthèses non supprimables : *(a((bc)d))* devient *a((bc)d)* puis *a(bcd)*, et non *abcd* qui signifierait *(((ab)c)d)*. Nous dirons que ces conventions sur la simplification du parenthésage forment le principe *(AG) d'association à gauche des symboles*. Nous appellerons *parenthèses déviantes* les parenthèses non supprimables par *(AG)*, celles de *a(bc)* par exemple. Nous dirons *en écriture simplifiée* les expressions qui résultent de la suppression de "*" et de l'emploi de *(AG)*. Notons qu'on pourrait éviter toute parenthèse si on conservait "*" dans les expressions et si on l'utilisait de manière préfixée : *a(bc)* s'écrirait **a*bc*, *abc* s'écrirait ***abc*, voire $*^2abc$. Toutefois, nous n'utiliserons pas ici ce mode d'expression, mais l'écriture dite plus haut *simplifiée*.

Une combinaison ne peut pas utiliser d'autres objets que ceux de la séquence mais elle peut :

1. ne pas prélever un ou plusieurs objets de la séquence (*b* est une combinaison sur la séquence *a,b*) ; on dira *éliminer* un ou plusieurs objets,
2. employer plusieurs fois un objet unique de la séquence (*bb* est une combinaison) ; on dira *dupliquer* un objet,
3. prélever dans un autre ordre que celui de la séquence deux objets de la séquence (former *ba* sur la séquence *a,b*) ; on dira *permuter* deux objets,
4. associer deux objets contigus quelconques de la séquence, en les groupant par des parenthèses (sur la séquence *a,b,c,a(bc)* est une combinaison) ; on dira *composer* deux objets,
5. reproduire sans modification la séquence, ce qui arrive quand la séquence comporte un seul objet (la séquence *a* et la combinaison *a* se confondent) ; on dira *identifier* l'objet de la séquence (s'il y a plusieurs objets dans la séquence, la combinaison ne reproduit jamais sans modification la séquence puisque celle-ci comporte alors des virgules (extrathéoriques), lesquelles ne peuvent se trouver dans une combinaison).

Ces cinq transformations, définies ici sur les atomes, doivent pouvoir porter également sur les combinaisons ainsi obtenues, c'est-à-dire sur les molécules (on pourra, par exemple, permuter *ab* et *cd*).

Il s'ensuit que toute combinaison applicative est le produit d'une ou de plusieurs éliminations, duplications, permutations, compositions, identifications des objets d'une séquence donnée et éventuellement de certaines de leurs molécules, à quelque rang que ce soit de la séquence ou de la combinaison déjà obtenue.

On appellera *suite d'atomes* (en abrégé *suite* quand il n'y a pas d'équivoque à craindre) toute combinaison d'atomes qui, exprimée en écriture simplifiée, ne comporte aucune parenthèse, c'est-à-dire où les éléments sont seulement juxtaposés : *abcd...* pour *(...(((ab)c)d)...)*. Par extension, un élément unique sera dit également *suite*. On prendra soin de distinguer la séquence *a,b,c,...* ensemble des objets donnés, et la suite *abc...* qui est une combinaison applicative. La juxtaposition des éléments n'indique pas leur simple coexistence. Celle-ci est marquée par les symboles extra-théoriques que sont les virgules[6].

Dans cette première approche, toute combinaison des objets d'une séquence est donc obtenue par des successions d'emploi direct de "*" sur des atomes choisis ou sur des molécules déjà formées. Si la séquence est *a,b,c,* par exemple, et si la combinaison à obtenir est *a(bc)*, il suffit de prélever *b* et *c* dans la séquence, d'appliquer *b* à *c*, ce qui obtient *bc*, de prélever *a* dans la séquence, d'appliquer *a* à *bc*, ce qui obtient *a(bc)*. D'une manière comparable, on obtiendra par exemple, *acb*.

Lorsqu'on procède ainsi, cependant, ou bien les démarches demeurent intuitives et non formulées, ou bien elles sont formulées, comme on vient de le faire, dans une langue informelle. Or, il est intéressant d'identifier la totalité des démarches requises pour obtenir toutes les combinaisons possibles à partir d'une séquence donnée. On va le faire, pour commencer, en se donnant des opérateurs qui auront pour opérandes des éléments de la séquence et pour résultante la combinaison à former. On ne devra pas se donner, toutefois, un opérateur par séquence et par combinaison, ce qui redoublerait stérilement les pratiques, il faudra identifier un ensemble fini, aussi petit que possible, d'opérateurs suffisants. C'est ainsi qu'à propos du premier exemple donné, on pourrait concevoir d'obtenir la combinaison *a(bc)* de la séquence *a,b,c* par un opérateur, disons **B'**, tel que **B'***(a,b,c)* → *a(bc),* et en introduisant par une flèche la résultante d'un opérateur portant sur la séquence à *n* éléments qui lui convient par hypothèse. De la même façon, à propos du deuxième exemple, on pourrait se donner un autre opérateur, disons **C'**, tel que **C'***(a,b,c)* → *acb*.On aurait aussi par exemple **I'***(a)* → *a*[7]. Le problème resterait de savoir si **B'**, **C'**, **I'** feront ou non partie des opérateurs qui seront ultimement retenus, mais il s'agirait au moins de la première étape de la recherche. On écrirait donc **B'***(a,b,c)*, **C'***(a,b,c)*, **I'***(a)* comme on écrit *f(a,b,c)*, *f(a)* pour des fonctions données et respectivemnt à trois arguments ou à un argument. On a bien affaire à des fonctions, d'ailleurs, puisque la résultante exprime la transformée par **B'**, et par **C'**, de *a,b,c*,par **I'** de *a* (au lieu d'***opérandes*** on peut donc dire ***arguments***). D'autre part, afin de convenir à toute séquence, on posera : **B'***(x,y,z)* → *x(yz)*; **C'***(x,y,z)* → *xzy* ; **I'***(x)* → *x; x,y,z* étant des métavariables désignant un objet quelconque, atomique ou moléculaire (à partir d'ici, selon le contexte, *x,y,...* sont les variables usuelles, par exemple, numériques, ou les métavariables d'objets quelconques propres à la logique combinatoire). C'est toutefois différemment que les opérateurs vont se trouver écrits et pensés, en utilisant une méthode de Schönfinkel qui présentera de nombreux avantages.

Cette méthode consiste à montrer qu'on a seulement besoin des fonctions à un argument (ou des opérateurs à une opérande) et à éliminer à leur profit toutes les fonctions (ou tous les

---

6 On voit que dans cette théorie il s'agit bien de combinaison au sens usuel du terme dans la pensée commune, c'est-à-dire de procédés pour former un nouveau tout en associant des éléments (ou de résultats de ces procédés). Les combinaisons obtenues correspondent à l'une des deux formes possibles de ces combinaisons usuelles : celle dans laquelle les éléments demeurent invariants dans le tout formé, comme les lettres dans un mot, par opposition à celle dans laquelle l'identité des éléments disparaît ou se modifie, comme dans les synthèses chimiques ou dans les Gestalts psychologiques. On se gardera de confondre ce qui est connu dans la littérature sous les noms de *logique combinatoire* et *d'analyse combinatoire*, même si les critiques que Hegel a adressées à la seconde (incriminant, en bref, une incapacité à traiter des combinaisons de la deuxième forme) ont sans doute une valeur comparable, mais qu'on ne peut discuter ici, à l'égard de la première discipline.

7 Si du moins on accepte que ces opérateurs puissent obtenir en un seul coup ce qui exige dans l'obtention réelle de la combinaison (même pour **I'**) plusieurs actes : prélèvement d'un objet puis d'un autre, application de l'un à l'autre, etc., ce qui semble admissible puisque ces actes sont nécessaires pour chaque cas.

opérateurs) qui en exigent davantage. Comme $f'(x,y)$, par exemple, signifie $f'* (x,y)$ (f' sur x,y obtient quelque effet) si au lieu d'appliquer f' à (x,y), c'est-à-dire de faire un usage unique de "*", on en fait deux usages, on pourra obtenir un effet identique à celui de f' en utilisant seulement des fonctions unaires ; $f'* (x,y)$ se comporte comme le ferait une fonction f unaire qui, appliquée à x, c'est-à-dire $(f * x)$, correspondrait à son tour à une fonction unaire, disons f'', définie comme obtenant par hypothèse, quand on l'applique à y, ce qu'obtient $f'(x,y)$. Autrement dit, sans marquer l'opérateur "*", $f'(x,y)$ peut se comprendre comme $(fx)y$, qu'on peut écrire par *(AG)* fxy. On voit que $*$, dans un sens plus précis, correspond à l'application d'une fonction unaire à son argument, à la transformation apportée par f à x. On raisonnerait de la même façon pour éliminer de proche en proche les fonctions à plus de deux arguments.

Cette analyse, en outre, vaut pour les relations du calcul des prédicats, et met sur la voie d'une autre formalisation possible du langage naturel : rxy, *x est père de y*, *y a pour père x*, sera compris comme $(rx)y$, où rx est conçu comme le prédicat unaire qu'il faut attribuer à y pour que rxy, en l'occurrence le prédicat...*avoir-pour-père-x* ; r lui-même étant le prédicat unaire qu'il faut supposer appliqué à x pour obtenir rx, qu'il fasse ou non défaut en français[8].

Il s'ensuit que les opérateurs $\mathbf{B}'(x,y,z)$, $\mathbf{C}'(x,y,z)$, $\mathbf{I}'(x)$ deviendront respectivement $\mathbf{B}xyz$, $\mathbf{C}xyz$, $\mathbf{I}x$, comme $f'(x,y,z)$ devient $fxyz$, comme $f'(x)$ devient fx. On aura donc : $\mathbf{B}xyz \rightarrow x(yz)$; $\mathbf{C}xyz \rightarrow xzy$; $\mathbf{I}x \rightarrow x$. **B**, **C**, **I** seront nommés *combinateurs*. L'introduction des combinateurs constitue ce qu'on nomme *la théorie intuitive des combinateurs.*

2. THEORIE INTUITIVE DES COMBINATEURS

Comme on doit, bien sûr, comprendre $(((\mathbf{B}*x)*y)*z) \rightarrow (x*(y*z))$, etc. chaque combinateur est donc en réalité un opérateur unaire, comme **I** l'est évidemment, bien qu'il reste commode de décrire **B** et **C** comme des opérateurs à trois arguments. Si la séquence est a,b,c et si on écrit $\mathbf{B}abc$ on dira que **B** porte sur la suite abc, mais **B** s'applique en réalité sur le premier élément de la séquence, choisi comme suite à un élément, puis $\mathbf{B}a$ s'applique sur le deuxième élément de la séquence, choisi comme une autre suite à un élément, etc.. En toute rigueur, d'ailleurs, il n'est pas non plus correct à notre avis, mais il est également commode et courant, de désigner **B**, **C**, **I** comme des opérateurs : **B**, **C**, **I** n'ont pas d'effet, seule l'application à un argument de **B**, de **C**, de **I**, par exemple $\mathbf{B}*a$, $\mathbf{C}*a$, $\mathbf{I}*a$ (même si "*" est sous-entendu) possède un effet. Il n'y a qu'un seul opérateur (qu'une seule opération) et c'est l'application.

Un langage qui ne comporte ainsi que cette seule opération est dit *langage applicatif.* On peut réduire tout langage comportant un certain nombre d'opérations (par exemple, + et $\times$, c'est-à-dire A' et M' en notation préfixée) à un langage applicatif, en procédant comme on

[8] On notera que cette réduction des relations aux prédicats unaires est fort différente de celle opérée par la logique classique qui comprenait *x est père de y* comme la qualification de *x* par l'attribut *étant-père-de-y*, grâce à la copule *est*. Il ne s'agit plus de résorber les relations dans une logique des propriétés, il s'agit d'une décomposition qui a par construction le même effet que la relation proprement dite. En revanche, $*$ demeure une opération binaire, même écrite de manière préfixe, $(*a)b$ semblant perdre tout intérêt. Ajoutons qu'à la composition du calcul des prédicats *(est père x de) y* la grammaire applicative substitue *(est père de y) x*, pour raisons linguistiques .

vient de le faire sur f', **B'**, **C'**, **I'**. $A'xy$ et $M'xy$ deviendront respectivement $(A*x) * y$ et $(M * x) * y$, soit encore Axy et Mxy. Soit, d'une manière générale, une opération n-aire O'_n telle que $O'_n(x_1, ..., x_n)$, on éliminera O'_n en la remplaçant par l'opération unaire O telle que $(...((O * x_1) * x_2)... * x_n)$, c'est-à-dire $O\, x_1 x_2 ... x_n$. Cette méthode est mise en oeuvre par la logique combinatoire quand elle s'occupe de formaliser des théories. L'élimination des variables n'est donc pas la seule des réduction auxquelles elle procède.

Formulés dans le langage commode accepté précédemment, **B** effectue donc une des compositions possibles, celle du 2e et du 3e élément de la suite, **C** effectue une des permutations possibles, celle du 2e et du 3e élément de la suite, **I** effectue l'identification de l'unique élément d'une suite. On constate que ces combinateurs laissent invariant leur premier (ou unique) argument. Il reste donc à introduire semblablement un combinateur effectuant une des répétitions possibles, celle du 2e élément de la suite, on posera : $\mathbf{W}xy \rightarrow xyy$, et un combinateur effectuant une des éliminations possibles, celle du 2e élément de la suite, on posera : $\mathbf{K}xy \rightarrow x$. Ces cinq combinateurs, dits *combinateurs élémentaires*, seront placés dans l'ordre, dit *normal* : **I**, **K**, **W**, **C**, **B**. Comme chacun n'effectue son effet qu'à un certain rang et pour une suite d'une longueur donnée, il restera à montrer comment on peut pour une suite et à un rang quelconques, éliminer, dupliquer, etc.. Tout langage combinatoire qui le permettra sera alors combinatoirement complet, c'est-à-dire donnera les moyens d'obtenir toutes les combinaisons applicatives qu'on peut former à partir d'une suite donnée. Il se trouve qu'on sait construire un langage formel exprimant la théorie intuitive des combinateurs.

a. *Langage combinatoire L*

Soit donc L ce langage dont l'alphabet comporte les données suivantes :
- un ensemble E infini dénombrable d'éléments : *a,b,c,d,e,f,g,h,a',...h',a'',...* (qui sont dits des *indéterminées*), **I**, **K**, **W**, **C**, **B** (qui sont dits *combinateurs élémentaires*),
- une opération binaire : $*$ (dite *application*),
- une relation binaire : $=$ (dite *égalité*),
- une relation binaire : $\rightarrow$ (dite *réductibilité*),
- deux symboles de ponctuation : (,) (dites *parenthèses*).

Les mots français, les virgules, les points de suspension, etc. n'appartiennent pas au langage formel lui-même, mais à la métalangue, c'est-à-dire à la langue utilisée (la langue U, dit Curry) pour présenter la langue formelle. Appartiennent aussi à la métalangue les variables intuitives, ou métavariables : *x,y,z,...*, ces mêmes lettres avec apostrophes ou indices numériques. Les règles de formation des termes et des formules de L sont les suivantes :

RF_1 tout élément de E est un terme
RF_2 si x et y sont des termes, $(x*y)$ est un terme
RF_3 rien d'autre n'est un terme
RF_4 si x et y sont des termes, $x = y$ et $x \rightarrow y$ sont des formules
RF_5 rien d'autre n'est une formule.

On admettra les mêmes licences que dans la théorie intuitive des combinaisons applicatives : suppression de "$*$", principe *(AG)*, etc..

Le mot *terme* a été choisi de préférence au mot *combinaison* pour utiliser une formulation aussi peu dépendante que possible d'une interprétation particulière, comme dans tout langage formel, mais les termes en question seront bien pour nous les combinaisons dont on a traité précédemment. Il faut remarquer, toutefois, que les combinaisons ne sont plus seulement des assemblages de lettres minuscules. Des majuscules peuvent aussi entrer dans les combinaisons, soit comme la totalité, soit comme une partie des éléments combinés. Les combinateurs eux-mêmes se traitent comme des objets (c'est pourquoi les majuscules et les minuscules figurent dans le même ensemble). La raison de cette mesure est claire : les combinateurs, comme on l'a dit, ne sont pas, en toute rigueur, des opérateurs. Réduits à eux-mêmes, ils sont inertes et donc de simples éléments ; ils n'ont d'effet que comme premier argument de $*$. C'est $\mathbf{I} * a$ qui agit et non **I**. **K** est ainsi une combinaison au même titre que *a*, **KI** au même titre que *ab*, **KI(CW)** au même titre que *ab(cd)*, et ainsi toute expression bien formée comportant à la fois des majuscules et des minuscules, comme **K***a***W***b***I**.

Curry oppose les indéterminées (ou objets quelconques) à la fois aux variables substitutives, c'est-à-dire libres et pourvues d'une règle de substitution, et aux variables liées, deux sortes de variables dont déjà L s'affranchit puisqu'il n'emploie pas de règle de substitution ni d'opérateur comme $\forall$, $\exists$ rendant la valeur de la formule indépendante de la valeur de la variable affectée. On notera aussi qu'il y a désormais deux catégories d'expressions bien formées : celles des termes, c'est-à-dire des symboles désignant des objets, et celles des formules, c'est-à-dire des énoncés (vrais ou faux) sur les objets. En ce sens, la logique combinatoire est bien un calcul "propositionnel", à sa manière, mais sur une base fort différente des calculs traditionnels. On peut établir formellement que telle ou telle expression, terme (t) ou formule (f), est bien formée. Ainsi $a(\mathbf{I}b)$, exprimé exactement comme $(a * (\mathbf{I}*b))$, est un terme car :

(1) a	RF_1 (t)
(2) $\mathbf{I}$	RF_1 (t)
(3) b	RF_1 (t)
(4) $(\mathbf{I}*b)$	RF_2 sur (2) et (3) (t)
(5) $(a*(\mathbf{I}*b))$	RF_2 sur (1) et (4) (t).

$(a*(\mathbf{I}*b)) \rightarrow a$, $(a*(\mathbf{I}*b)) = a$ sont des formules par RF_4 sur (5) et (1). Dire qu'une expression est bien formée ne signifie pas qu'elle est vraie.

A chacun des cinq combinateurs élémentaires est associée une règle (dite *de récriture* ou *de réduction*) qui exprime la transformation qu'il fait subir à une certaine suite.

Nom de la règle	Formulation de la règle	Nom du combinateur présent
(I)	$\mathbf{I}x \rightarrow x$	*identificateur (élémentaire)*
(K)	$\mathbf{K}xy \rightarrow x$	*éliminateur (élémentaire)*
(W)	$\mathbf{W}xy \rightarrow xyy$	*duplicateur (élémentaire)*
(C)	$\mathbf{C}xyz \rightarrow xzy$	*permutateur (élémentaire)*
(B)	$\mathbf{B}xyz \rightarrow x(yz)$	*compositeur (élémentaire)*

Nous appellerons ces règles *(Comb)*. Elles peuvent être comprises sur le modèle des instructions utilisées dans un programme informatique. Leur expression permet à un combinateur de s'appliquer à lui-même, on a par exemple, **WW***a*, ou à un combinateur différent, **WK***a* par exemple, puisque chaque combinateur est donné comme portant sur une suite d'objets quelconques. Nous poserons :

$x \rightarrow y$ est une formule vraie si y s'obtient de x par les règles (*Comb*).

Sont ainsi des formules vraies : $\mathbf{I}a \rightarrow a$; $\mathbf{W}ab \rightarrow abb$; $\mathbf{II} \rightarrow \mathbf{I}$; $\mathbf{IK} \rightarrow \mathbf{K}$.

Remarques :

(1) plusieurs applications des règles (*Comb*) peuvent intervenir successivement. Nous écrirons $x \rightarrow y \rightarrow z$ .... pour $x \rightarrow y$, $y \rightarrow z$,...., et nous sous-entendrons que $x \rightarrow y$, etc. sont des formules vraies. On a par exemple, en numérotant les arguments et en explicitant les règles employées (ce dont nous nous dispenserons par la suite) : **WW**a → **W**aa → aaa

12 (**W**) *12* (**W**)

WKa → **K**aa → a, ou selon une autre disposition :

12 (**W**) *12* (**K**)

1 **WK**a
12

2 **K**aa par (**W**) sur **K**a
12

3 a par (**K**) sur aa

(2) un terme moléculaire enclos dans des parenthèses non supprimables par (*AG*) forme par RF_2 un seul terme et donc un argument unique : **BI**ab → **I**(ab) → ab

123 (**B**) *1* (**I**)

(exactement : (ab) puis ab, par (*AG*)).

(3) lors de certaines récritures, des parenthèses qui étaient déviantes cessent de l'être et doivent être supprimées par (*AG*) : **B(WK)**ab → **WK**(ab) → **K**$(ab)(ab)$ → ab, et non **B(WK)**ab → **(WK)**(ab), ni **K**$(ab)(ab)$ → (ab).

(4) une récriture peut ne pas s'effectuer faute d'un nombre suffisant d'arguments : **C**ab ne se récrit pas ; **BK**ab se récrit **K**(ab) mais **K**(ab) ne se récrit pas[9]. On remarquera qu'un

---

9 L'expression *récrire* est consacrée mais un peu dangereuse. Elle donne à croire que les transformations portent sur des graphismes. Il ne s'agit pas en réalité de tracer différemment une suite de signes mais d'exprimer une transformation effectuée sur les entités qu'ils représentent. On n'a pas, autrement dit, **K**"a""b" → "a", c'est-à-dire une intervention portant sur des lettres, mais **K**ab → a, une intervention portant sur les entités auxquelles renvoient les lettres. C'est d'ailleurs parce qu'il est un objet de pensée et non un symbole qu'on peut, dans la théorie intuitive des combinaisons, employer plusieurs fois un élément qui figure une seule fois dans la séquence. On le cite à nouveau plutôt qu'on ne le duplique. Certes, ces entités sont quelconques, et à ce titre des signes sur un papier en font partie mais ils ne peuvent jamais s'identifier aux entités en cause car un langage formel ne peut exclure aucune interprétation de ses graphismes. Il demeure une différence importante entre les opérateurs combinatoires et les autres, les connecteurs propositionnels par exemple : la suite initiale et la récriture d'un combinateur se trouvent l'équivalent de la table de vérité d'un connecteur, elles détermiment des propriétés sémantiques, mais elles le font à l'intérieur même du langage formel (à l'aide des termes de L) sans présupposer le renvoi à un champ d'interprétation qui lui soit extérieur (et bien que la logique combinatoire puisse aussi interpréter ses termes dans un tel champ). Nulle part il n'apparaît mieux que chercher une application d'un langage formel (une classe d 'objets concernés) c'est au fond en chercher une deuxième, ou plutôt particulariser celle qui se confond avec la construction elle-même du langage et qui rapporte déjà aux symboles des objets, même s'ils sont indéterminés.

même symbole peut être opérateur ou opérande suivant sa position et non seulement suivant sa forme typographique[10].

On appellera d'une façon générale *combinateur* tout opérateur qui transforme une certaine suite (dite *initiale*) de termes d'un langage donné en une combinaison applicative quelconque de termes de ce langage (dite *récriture* ou *résultante*), sans restriction sur les éléments figurant dans cette combinaison et en écrivant en métavariables tous les termes autres que le combinateur. Il s'ensuit qu'une combinaison de combinateurs est un combinateur. Dès lors, on définira inductivement la propriété d'être un combinateur de L de la manière suivante :

- **I, K, W, C, B** sont des combinateurs de L
- si x et y sont des combinateurs de L, $(x*y)$ est un combinateur de L
- rien d'autre n'est un combinateur de L.

$(x*y)$ s'écrira comme d'habitude xy. D'autre part "rien d'autre n'est un combinateur de L" s'entend au nom près. On peut parler du combinateur **T**, par exemple, comme d'un combinateur de L si **T** désigne **CI**. Nous utiliserons $X,Y,Z,X',...,Z',X'',....$ comme métavariables de combinateurs quelconques (atomiques ou moléculaires). **KW(IB)** par exemple est un combinateur de L, il peut être désigné par X. On appellera *combinateur proprement dit* un combinateur X dont la règle de récriture est de la forme :

$$Xx_1 \dots x_n \rightarrow y_1 \dots y_m ,$$

où $y_1,...,y_m$ sont des combinaisons de $x_1,...,x_n$ (la récriture ne comporte pas d'élément absent de la suite initiale). Ainsi, $\mathbf{A}xy \rightarrow xyz$, $\mathbf{A}'xy \rightarrow A'yx$ ne seraient pas des combinateurs proprement dits. Ces opérateurs, en effet, ne combinent pas, au sens usuel du mot, les éléments sur lesquels ils portent, ou pas seulement. De même, un opérateur comme :

$\mathbf{A}''xyzw \rightarrow yw$ si w est le même élément que x

$\mathbf{A}''xyzw \rightarrow zw$ si w n'est pas le même élément que x

qui comporte deux règles de récriture et qui discrimine l'effet obtenu selon la nature de certains éléments de la suite, ne serait pas non plus un combinateur proprement dit. Il ne s'ensuit pas, d'ailleurs, que ces opérateurs n'ont pas d'intérêt. **A''**, par exemple, a été utilisé par John T.

10 En logique non combinatoire il y a des opérateurs comme ~, et ils ne sont jamais alors opérandes, et des opérandes comme p , et elles ne sont jamais alors opérateurs, la forme géométrique du symbole marquant s'il s'agit d'un opérateur ou d'une opérande : $\sim\sim p$, par exemple n'est pas $(\sim\sim)\, p$, l'opérateur ~ ne porte pas sur un opérateur ~ devenu son opérande, mais $\sim(\sim p)$, où ~p est une opérande, comme q peut l'être, simplement opérande indiquée comme de valeur inverse de celle de p. En logique combinatoire, au contraire, on a par exemple $\mathbf{WW}a$: le premier **W** est un opérateur dont le second **W** est opérande. On obtient $\mathbf{W}aa$ où le **W** qui était opérande devient opérateur. La distinction traditionnelle entre des agents et des patients s'atténue considérablement car, s'il faut l'appliquer pour qu'une entité exerce une action, n'importe quelle entité suivant sa place agit ou pâtit.
L'hylémorphisme d'Aristote (dont les logiques usuelles demeurent tributaires) se représente la formalisation logique et la prise de forme physico-technique sur le même schème : l'imposition active d'un moule (les constantes logiques, pour l'essentiel, dans le premier cas) à une matière, donnée malléable et passive (les variables ou l'argile) qui reçoit la forme. En logique combinatoire, au contraire, comme dans l'opération physico-technique mieux décrite (dans laquelle tour à tour le moule et l'argile supportent puis exercent des forces), un même élément peut être tranformé puis transformateur ; voir G. Simondon, *L'individu et sa genèse physico-biologique*, Paris, P.U.F., 1974, p. 27 et sq. (*forme et matière*).

Kearns sous le nom de *discriminateur* pour élaborer une forme originale de langage formel, mais non proprement combinatoire[11].

On appellera *combinateur régulier* un combinateur X dont la règle de récriture est de la forme :

$$X x_1\, x_2 \,....\, x_n \rightarrow x_1\, y_1 \,...\, y_m \,,$$

où $y_1,...,y_m$ sont des combinaisons de $x_2,...,x_n$ (c'est-à-dire laisse le premier argument invariant) ; les combinateurs réguliers sont des combinateurs proprement dits. Les combinateurs élémentaires sont réguliers, et sont donc des combinateurs proprement dits.

Par les règles (*Comb*), on ne peut pas récrire d'expressions comme $a(\mathbf{I}b)$,$\mathbf{K}abc$. Puisqu'il y a un terme devant $\mathbf{I}b$, la règle (**I**) ne s'applique pas ; puisqu'il y a un terme surnuméraire après $\mathbf{K}ab$, la règle (**K**) ne s'applique pas. On peut toutefois estimer naturel d'en donner une récriture qui conserve sans modification les termes qui précédent ou qui suivent l'expression à laquelle, en leur absence, les règles de récriture s'appliqueraient. Nous ajouterons donc les règles de récriture suivantes :

(*MG*) si x commence par un combinateur et si $x \rightarrow y$, alors quel que soit z, $xz \rightarrow yz$ (règle dite *de monotonie gauche*)

(*MD*) si x commence par un combinateur et si $x \rightarrow y$, alors quel que soit z, $zx \rightarrow zy$ (règle dite *de monotonie droite*)

On a par exemple : $\mathbf{K}abc \rightarrow ac$ car $\mathbf{K}ab \rightarrow a$ par (**K**), donc $\mathbf{K}abc \rightarrow ac$ par (*MG*) ; $a(\mathbf{I}b) \rightarrow ab$ car $\mathbf{I}b \rightarrow b$ par (**I**), donc ab par (*MD*). En revanche, $a\mathbf{I}b$, c'est-à-dire $(a\mathbf{I})b$, ne peut pas se récrire, l'expression ne comprenant aucun terme commençant par un combinateur. Soit $\mathbf{K}(\mathbf{W}ab)$, qui ne se récrit pas par (**K**), on a $\mathbf{W}ab \rightarrow abb$, donc par (*MD*) $\mathbf{K}(abb)$, qui ne se récrit plus, et non $\mathbf{K}abb$, puisque **K s'applique** à $\mathbf{W}ab$ et donc aussi à la récriture de $\mathbf{W}ab$ comme un tout. Nous sous-entendrons souvent l'usage de (*MG*) et de (*MD*)[12]. Nous poserons désormais :

$x \rightarrow y$ est une formule vraie si et seulement si y s'obtient de x par les règles (*Comb*), (*MG*), (*MD*).

On constate qu'il y a dans L des expressions bien formées, possédant des combinateurs pourvus d'une suite suffisante et qui cependant ne se réduisent pas à une expression sans combinateur. Il en va ainsi pour **WWW**, et quelle que soit la suite d'indéterminées qu'on lui rapporte car $\mathbf{WWW}a... \rightarrow \mathbf{WWW}a...$ par (**W**), les règles (*MG*), (*MD*) étant ici sans utilité ou sans emploi. Le rejet de ces expressions (par exemple au moyen d'une action sur les règles de formation) est interdit par le principe dit *de la formalisation complète*. Curry entend par

---

11 John T. Kearns, *Combinatory logic with discriminators*, *The Journal of Symbolic Logic*, vol. 34, number 4, dec. 1969, p. 561-575. L'intérêt qu'il y a à travailler avec des combinateurs proprement dits est comparable à celui qu'on trouve en physique à travailler sur des systèmes clos. On cherche à éviter, autant que cela est possible, les irruptions parce qu'elles sont des coups de force pour la raison.

12 Au vu de la formulation de (*MG*) et de (*MD*), on pourrait considérer que ces règles font perdre à L son caractère proprement combinatoire puisque z semble faire irruption dans la récriture. On voit dans nos exemples que ce n'est pas le cas : quand (*MG*) intervient dans la récriture de $\mathbf{K}abc$, il n'introduit pas c , il se borne à le conserver. De même (*MD*) n'introduit pas a devant $\mathbf{I}b$, il permet d'employer (**I**) malgré a, supposé déjà présent.

là l'exigence de ne pas exclure par des mesures à notre convenance des expressions prétendument dénuées de sens dont les composants en sont pourvus (**W***ab*....,**WW***a*..., ne faisant pas difficulté), la compréhension intuitive de l'homme ne devant pas être la mesure de toutes choses en mathématiques, et cela d'autant plus que **WWW...W** n'introduit pas d'inconsistance dans le langage formel[13].

Si on dit qu'une expression *est en forme normale* quand elle ne comporte pas de combinateur, et qu'elle *a une forme normale* quand elle peut être récrite en une expression qui ne comporte pas de combinateur, **W***ab* a une forme normale qui est *abb*, **WWW...** n'a pas de forme normale. Il faut distinguer plusieurs cas d'expressions bien formées n'ayant pas de forme normale :

(1) les expressions comme **WWW** qui se réduisent à elles-mêmes un nombre infini de fois : **WWW** → **WWW** → **WWW** ...

(2) les expressions qui se réduisent à un nombre infini de combinaisons, comme **WI(BW(WI))***a*, puisque **WI(BW(WI))***a* → **I(BW(WI))(BW(WI))***a* → **BW(WI)(BW(WI))***a* → **W(WI(BW(WI)))***a* → **WI(BW(WI))***aa*, etc.

(3) les expressions de forme $Xx_1 \ldots x_n$ dans lesquelles $x_1 \ldots x_n$ ne fournit pas une suite suffisante d'arguments, comme **K***a*, **BW***a*.

(4) les expressions qui se réduisent à une expression qui ne se réduit plus elle-même, comme **IBW***a* qui se récrit **BW***a*, expression qui ne se récrit pas.

(5) les expressions qui ne commencent pas par un combinateur et pour lesquelles (*MD*) n'est pas applicable, comme *a***I***b*, ou qui se réduisent à une telle expression, comme **CIK***abc*, puisque **CIK***abc* → **I***a***K***bc* → *a***K***bc*. On notera que **CIK** n'est donc pas un combinateur proprement dit, il correspond à un combinateur qui transforme *abc* en *a***K***bc*, bien que **C**, **I**, et **K** soient des combinateurs proprement dits. La logique combinatoire proprement dite recèle, par conséquent, des opérateurs qui ne sont pas des combinateurs proprement dits.

Seuls les deux premiers cas et la conséquence du dernier font vraiment difficulté, sans doute, parce qu'ils font apparaître des résultats inattendus, mais Curry refuse, en somme, de subordonner la réalité mathématique à ce qu'on pouvait en attendre[14].

D'autre part, nous sommes toujours libres d'introduire de nouveaux combinateurs, même s'ils se révèlent définissables par les anciens, pourvu qu'ils ne se confondent pas aux symboles près avec eux, et dès lors qu'ils sont mieux appropriés aux applications qu'on se propose d'en faire. Il peut être intéressant de faire usage de cette liberté dans les sciences de l'homme car leurs besoins ne sont pas exactement ceux des sciences mathématiques qui ont été à l'origine de la logique combinatoire. Nous ne présenterons, toutefois, que trois combinateurs encore classiques, **S**, **Φ**, **Ψ** (nommés *compositeurs non élémentaires*) adjoints aux éléments de E dans L :

nom de la règle	formulation de la règle
(S)	$\mathbf{S}xyz \to x\,z\,(y\,z)$
(Φ)	$\mathbf{\Phi}\,x\,y\,z\,w \to x\,(y\,w)\,(z\,w)$
(Ψ)	$\mathbf{\Psi}\,xyzw \to x(yz)\,(yw)$

[13]Curry s'oppose ainsi à un courant important : celui de la logique et de la philosophie russelliennes du non sens.

[14]La position de Curry ne supprime pas toute différence entre les termes qui ont une forme normale et ceux qui n'en n'ont pas ; même les seconds ont un sens (large, comme objets de pensée), mais les premiers en ont, de surcroît, un autre (plus restreint, comme contenu manifeste obtenu par l'effectuation).

Nous supposerons que les mesures prises antérieurement sont adaptées à cette extension (qui gardera le nom de L).

Il arrive qu'un combinateur ait sur une suite donnée le même effet qu'un autre. On a par exemple :

W ab → abb ; **BWC**ab → **W**(**C**a)b → **C**abb → abb.Nous dirons que pour deux combinateurs quelconques X,Y :

$X = Y$ est une formule vraie (ou une loi de L ; ou que X et Y sont égaux) si et seulement si pour toute suite σ telle que à la fois $X\sigma$ et $Y\sigma$ peuvent se récrire, alors $X\sigma$ et $Y\sigma$ se récrivent en obtenant ultimement la même combinaison.

Cette propriété est réfutée si pour la plus courte des suites de termes $x, y,...,$ permettant d'appliquer X et Y, disons σ_1, les deux combinaisons obtenues diffèrent. Elle est établie dans le cas contraire car pour les suites plus longues que σ_1, X et Y obtiendront encore, grâce à (MG), la même combinaison. Ainsi à **I** il faut au moins x, à **BI** il faut au moins xy, donc σ_1 est xy. Il vient : **I**xy → xy; **BI**xy → **I**(xy) → xy.

L'identité des combinaisons obtenues se maintient pour les suites plus longues, xyz,$xyzw$, etc. par (MG). Donc **BI** = **I**. Les lois combinatoires sont bien des lois logiques au sens usuel du mot : des formules universellement valides. Il faut remarquer que les règles de récriture n'établissent pas les égalités elles-mêmes $X = Y$.

Si on peut écrire $X = Y$ une loi de L, bien qu'on ait appliqué X puis Y à une certaine suite, c'est qu'il est inutile d'exprimer celle-ci car (1) elle doit être quelconque, dès qu'elle est suffisante, les noms des arguments n'important pas, (2) elle doit être la même pour X et pour Y. Toutefois, il est clair que c'est l'ensemble des mesures prises par la logique combinatoire qui conduit à cette élimination, notamment la définition positionnelle des opérateurs et des opérandes, les règles de monotonie, etc..

La manière dont l'égalité de X et de Y a été caractérisée est évidemment extensionnelle : on identifie les combinateurs ayant sur les mêmes suites le même effet. D'autres manières de caractériser l'égalité sont possibles, et plusieurs ont été examinées. Les déterminations non extensionnelles (*intensionnelles*, comme on dit) sont nombreuses, en entendant par là tout traitement pour lequel un critère plus sévère d'égalité est donné. Ce critère est choisi en général comme ajoutant des conditions à celles du critère extensionnel car si les entités n'ont pas le même effet on peut par là les distinguer. Le critère est choisi aussi, en général, pour correspondre au "sens" des combinateurs X et Y, selon une théorie préalable de leur synonymie.

b. *Algèbre des combinateurs.*

Nous introduirons aussi une opération "." (dite *produit*) formant des expressions en toute rigueur hors système mais qui pourront abréger des termes bien formés de L, et telle que $(X.Y)x_1\, x_2 \dots x_n$ consiste à appliquer X à $x_1\, x_2 \dots x_n$ puis Y à la combinaison ainsi obtenue. Quand X est régulier, $(X.Y) =_{df} \mathbf{B}XY$.

Dans ce cas, en effet :

$$Xx_1\, x_2 \dots x_n \to x_1\, y_1 \dots y_m$$

$$Yx_1\, y_1 \dots y_m \to z_1\, z_2 \dots z_p$$

Or, $\mathbf{B}XYx_1\,x_2\,...\,x_n \rightarrow X(Yx_1)x_2\,...\,x_n \rightarrow Yx_1y_1\,...\,y_m \rightarrow z_1\,z_2\,...\,z_p$. Ainsi $(\mathbf{C}.\mathbf{K})abc$ se récrit : $\mathbf{C}abc \rightarrow acb$, $\mathbf{K}acb \rightarrow ab$; $\mathbf{BCK}abc \rightarrow \mathbf{C}(\mathbf{K}a)bc \rightarrow \mathbf{K}abc \rightarrow ab$, tandis que $\mathbf{CK}abc \rightarrow \mathbf{K}bac \rightarrow bc$. Quand X n'est pas régulier, un artifice, exposé plus loin, permet de trouver un terme bien formé de L qui correspond à $(X.Y)$. De là, on peut définir par induction la puissance d'un combinateur quelconque X, en posant : $X^1 =_{df} X$; $X^{n+1} =_{df} (X^n\,.\,X)$. On obtient donc : $X^1 = X$; $X^2 = (X.X)$; $X^3 = ((X.X).X)$, ou par associativité $(X.X.X)$, etc.. Nous appellerons *combinateur à puissance* tout combinateur X^n où $n > 1$. Si X est régulier, on a donc aussi :

$$X^1 = X \;;\; X^2 = \mathbf{B}XX \;;\; X^3 = \mathbf{B}(\mathbf{B}XX)X \text{ etc..}$$

On peut établir l'effet de chaque puissance sur chaque combinateur. Si on se borne à l'étude des combinateurs élémentaires, il apparaît deux types de cas : ou bien le combinateur à puissance X^n n'a pas besoin d'un nombre d'arguments supérieur à celui de X pour donner lieu à récriture (c'est le cas de $\mathbf{I}^n$, $\mathbf{C}^n$, $\mathbf{W}^n$) ou bien il a besoin d'un nombre d'arguments supérieur (c'est le cas de $\mathbf{K}^n$, $\mathbf{B}^n$). Il vient, en effet :

	$\mathbf{I}x \rightarrow x$		$\mathbf{C}xyz \rightarrow xzy$		$\mathbf{W}xy \rightarrow xyy$
$\mathbf{I}^2$	$\mathbf{I}x \rightarrow x$	$\mathbf{C}^2$	$\mathbf{C}xzy \rightarrow xyz$	$\mathbf{W}^2$	$\mathbf{W}xyy \rightarrow xyyy$
$\mathbf{I}^3$	$\mathbf{I}x \rightarrow x$	$\mathbf{C}^3$	$\mathbf{C}xyz \rightarrow xzy$	$\mathbf{W}^3$	$\mathbf{W}xyyy \rightarrow xyyyy$

$\mathbf{C}^3$ par exemple, est le combinateur qui, appliqué à xyz obtient (comme le ferait $\mathbf{C}$) la combinaison xzy. On voit qu'on a plus généralement : $\mathbf{C}^n = \mathbf{I}$, si n est pair, $\mathbf{C}^n = \mathbf{C}$, si n est impair. $\mathbf{W}^n$ écrit le 2e argument de $\mathbf{W}$, puis le répète n fois, c'est-à-dire le reproduit $n + 1$ fois. Il est facile d'établir le nombre d'arguments et l'effet de $\mathbf{K}^n$ et de $\mathbf{B}^n$:

	$\mathbf{K}xyz \rightarrow xz$	$\mathbf{K}xyzw \rightarrow xzw$		$\mathbf{B}xyzw \rightarrow x(yz)w$		$\mathbf{B}xyzwx' \rightarrow x(yz)wx'$
$\mathbf{K}^2$	$\mathbf{K}xy \rightarrow x$	$\mathbf{K}xzw \rightarrow xw$	$\mathbf{B}^2$	$\mathbf{B}x(yz)w \rightarrow x(yzw)$		$\mathbf{B}x(yz)wx' \rightarrow x(yzw)x'$
	$\mathbf{K}^3$	$\mathbf{K}xw \rightarrow x$			$\mathbf{B}^3$	$\mathbf{B}x(yzw)x' \rightarrow x(yzwx')$

$\mathbf{K}^n$ exige $n + 1$ arguments et les supprime tous sauf le premier. $\mathbf{B}^n$ exige $n + 2$ arguments et les associe tous, sauf le premier, par une paire de parenthèses. On a ainsi de manière générale : $\mathbf{B}^n\,x_1x_2...\,x_{n+2} \rightarrow x_1(x_2\,...\,x_{n+2})$. Il est évident que les résultats qui précèdent sont démontrables par récurrence.

$\mathbf{B}^n$ reporte plus à droite l'action de tout combinateur régulier X différent de $\mathbf{I}$ qu'il précède. Un combinateur de forme $\mathbf{B}^nX$ (où $n \geq 1$) sera dit *combinateur à distance* : $\mathbf{B}^n$ provoque une action différée de X, et différée de n arguments à droite. En effet, si dans la formule ci-dessus $\mathbf{B}^nx_1x_2..x_{n+2} \rightarrow x_1\,(x_2\,...\,x_{n+2})$ on place un combinateur X régulier différent de $\mathbf{I}$ avant x_1 (pour obtenir $\mathbf{B}^nX$), et en appliquant $\mathbf{B}^n$ à une suite lui donnant $n + 2$ arguments, il vient : $\mathbf{B}^nXx_1...\,x_{n+1}... \rightarrow X(x_1...\,x_{n+1})...$ Comme X est régulier par hypothèse, il agira à partir de x_{n+2} ; si donc X, employé seul, agissait au rang 2, il agira, précédé de $\mathbf{B}^n$ au rang $n + 2$; s'il agissait au rang 3, il agira, précédé de $\mathbf{B}^n$, au rang $n + 3$, etc. Si donc X agissait au rang m, il agira au rang $n + m$.

Or, les combinateurs élémentaires différents de $\mathbf{I}$ agissent au rang 2, comme $\mathbf{K}$, $\mathbf{W}$, ou aux rangs 2 et 3, comme $\mathbf{C}$, $\mathbf{B}$. Précédés de $\mathbf{B}^n$, ils agiront donc au rang $n+ 2$, ou aux rangs

$n + 2$, $n + 3$. En pratique, pour éliminer (respectivement, dupliquer) le k-ème élément d'une suite, on lui applique $\mathbf{B}^{k-2}\mathbf{K}$ (respectivement, $\mathbf{B}^{k-2}\mathbf{W}$), pour permuter (respectivement, composer) le k-ème élément d'une suite et le suivant, on lui applique $\mathbf{B}^{k-2}\mathbf{C}$ (respectivement, $\mathbf{B}^{k-2}\mathbf{B}$). Si X est **I**, on a sur le même modèle : $\mathbf{B}^n\mathbf{I}x_1 \ldots x_{n+1} \rightarrow \mathbf{I}(x_1 \ldots x_{n+1}) \rightarrow x_1 \ldots x_{n+1}$. $\mathbf{B}^n\mathbf{I}$ reproduit sans modification les termes $x_1 \ldots x_{n+1}$ de la suite initiale, c'est-à-dire $n+1$ éléments. $\mathbf{B}^n\mathbf{I}$ et **I** diffèrent par le nombre des éléments de leur suite initiale la plus courte. On a raisonné jusqu'ici sur les combinateurs à distance de forme $\mathbf{B}^nX$, mais on peut avoir plus généralement des combinateurs, dits aussi *à distance*, de forme $\mathbf{B}^nX^m$, pour X régulier (ce qui implique X^m régulier), $n, m \geq 1$.

Inversement, si X est un combinateur élémentaire, différent de **I**, l'insertion de **I** entre X et la suite provoque ce qu'on peut nommer *l'action rapprochée* de X au premier élément de la suite (le combinateur X**I** est non régulier). Placé devant une suite qui lui suffit, **KI** supprime, **WI** duplique le premier élément de la suite, **CI** permute le premier et le deuxième élément de la suite. **BI** et **II** ont le même effet que **I**, l'action n'est donc pas rapprochée.

Comme on connaît l'effet des combinateurs $X^n (n \geq 1)$, une règle de récriture est associable à chacun. Soit par exemple : $(\mathbf{K}^2)$ $\mathbf{K}^2xyz \rightarrow x$; $(\mathbf{B}^3)$ $\mathbf{B}^3xyzwx' \rightarrow x(yzwx')$. Dès lors, pour effectuer une expression de la forme $\mathbf{B}^nX^m$ ....., ($n,m \geq 1$, X régulier différent de **I**), on applique la règle $(\mathbf{B}^n)$, en rapportant à $\mathbf{B}^n$ ses $n+2$ arguments (dont X^m est le premier) et on obtient X^m ..., puis on applique la règle (X^m). Ainsi : $\mathbf{B}^3\mathbf{K}^2abcdef \rightarrow \mathbf{K}^2(abcd)ef \rightarrow abcdf$; $\mathbf{B}^2\mathbf{C}abcde \rightarrow \mathbf{C}(abc)de \rightarrow abced$. On écourte souvent ces récritures en supposant des règles associées à des combinateurs moléculaires : $\mathbf{B}^2\mathbf{C}abcde \rightarrow abced$, *par* $(\mathbf{B}^2\mathbf{C})$.

Certains combinateurs à puissance et à distance généralisant les combinateurs élémentaires et leurs effets, on appellera *identificateurs* les combinateurs $\mathbf{I}^m$, $\mathbf{B}^n\mathbf{I}^m$; *éliminateurs* les combinateurs $\mathbf{K}^m$, $\mathbf{B}^n\mathbf{K}^m$; *duplicateurs* les combinateurs $\mathbf{W}^m$, $\mathbf{B}^n\mathbf{W}^m$; *permutateurs* les combinateurs $\mathbf{C}^m$, $\mathbf{B}^n\mathbf{C}^m$; *compositeurs* les combinateurs $\mathbf{B}^m$, $\mathbf{B}^n\mathbf{B}^m$ $(m,n \geq 1)$. Notons, toutefois, que les combinateurs non élémentaires peuvent avoir aussi des effets permutateurs, des effets duplicateurs, des effets compositeurs, etc. (**S** a d'ailleurs ces trois effets). Nous écrirons $\mathfrak{I}$, $\mathfrak{K}$, $\mathfrak{W}$, $\mathfrak{C}$, $\mathfrak{B}$ tout produit, éventuellement à un seul facteur, respectivement d'identificateurs, d'éliminateurs, de duplicateurs, de permutateurs, de compositeurs. On a par exemple : $\mathfrak{I}$ pour **I**, **BI**, (**I.BI**) ; $\mathfrak{K}$ pour **K**, $\mathbf{K}^2$, $((\mathbf{K}.\mathbf{K}^3).\mathbf{B}^4\mathbf{K}^2)$; $(\mathbf{I}.\mathbf{BI}.\mathbf{K}^2)$ est un produit d'identificateurs et d'éliminateurs. On va montrer qu'on peut obtenir une combinaison applicative quelconque $\mathfrak{X}$ par un produit de $\mathfrak{I}$, $\mathfrak{K}$, $\mathfrak{W}$, $\mathfrak{C}$, $\mathfrak{B}$, puis on donnera une formulation proprement combinatoire à ce produit.

Si $\mathfrak{X}$ comporte une majuscule au moins, on remplace $\mathfrak{X}$ par une combinaison pure, c'est-à-dire d'indéterminées, $ab\mathbf{I}$, par exemple, devenant abc, $\mathbf{K}(\mathbf{WK})$ devenant $a(ba)$. La combinaison pure obtenue par un produit, il suffira de remplacer, là où il y a lieu, dans la suite et dans la récriture, les indéterminées par les majuscules de la combinaison originale pour obtenir $\mathfrak{X}$. On ne fait pas appel à une règle de substitution, mais on exploite seulement le sens du mot *indéterminée*. Le problème se réduit, dès lors, à former toutes les combinaisons possibles d'indéterminées.

Soit $\dot{x}_1, \ldots, \dot{x}_n$ des indéterminées différentes appartenant à L, soit $\mathfrak{X}$ $[\dot{x}_1, \ldots, \dot{x}_n]$ une combinaison applicative quelconque formée à partir de quelques unes des indéterminées ou de toutes les indéterminées $\dot{x}_1, \ldots, \dot{x}_n$, alors s'il existe un combinateur de L, disons X', tel que

$X'\dot{x}_1 \ldots \dot{x}_n \rightarrow \mathfrak{X}[\dot{x}_1,\ldots,\dot{x}_n]$ L est dit <u>*combinatoirement complet*</u>. Autrement dit, il faut montrer qu'on peut à un rang quelconque des éléments de la suite $\dot{x}_1 \ldots \dot{x}_n$ effectuer chacune des cinq transformations qui obtiennent une combinaison applicative : identifier, éliminer, dupliquer, permuter, composer. On notera d'abord que chacun des cinq combinateurs élémentaires, sauf **C**, opère sur un seul élément, comme **I, K, W**, voire plusieurs fois, ou quand il s'agit de **B** sur deux éléments nécessairement contigus (on ne peut associer ni un seul élément, ni, sans modification antérieure, deux éléments distants), voire plusieurs fois, c'est-à-dire sur les résultats déjà acquis. Or, on vient d'établir que ces transformations peuvent être effectuées à tout rang par un combinateur, élémentaire ou à puissance ou à distance ou à action rapprochée. Au contraire des quatre transformations précédentes, il y a un sens à affecter aussi des éléments non contigus par un permutateur, par exemple *a* et *c* dans *abc*. On remarquera que cela est possible en effectuant plusieurs permutations sur des éléments contigus, par exemple **(CI.C.CI)** récrit *abc* en *cba*. On peut donc aussi permuter deux éléments situés à un rang quelconque. En conséquence, un produit utilisant autant que de besoin des combinateurs $\mathfrak{J}$, $\mathfrak{K}$, $\mathfrak{W}$, $\mathfrak{C}$,$\mathfrak{B}$ obtiendra $\mathfrak{X}$ On peut donner à ce produit la forme suivante (dite *du produit normal*) : ($\mathfrak{J}$.$\mathfrak{K}$.$\mathfrak{W}$.$\mathfrak{C}$.$\mathfrak{B}$), autrement dit en imposant aux facteurs du produit d'être dans l'ordre donné (dit *normal*) avec un facteur au plus de chaque espèce, le premier facteur, c'est-à-dire **I**, pouvant être sous-entendu et certains autres absents, faute d'utilité.

Soit par exemple à obtenir la combinaison *a(bb)c* ; il suffit de se donner la suite *abc*. Il vient : **W***abc* $\rightarrow$ *abbc*, **B***abbc* $\rightarrow$ *a(bb)c*. Le produit est donc **(W.B)**. Mais il se trouve également que sont imposées à la fois la suite et la combinaison ; soit par exemple à obtenir la combinaison *a(cc)d* à partir de la suite *abcd*. Il vient : **K***abcd* $\rightarrow$ *acd*, **W***acd* $\rightarrow$ *accd*,**B***accd* $\rightarrow$ *a(cc)d*. Le produit est donc **(K.W.B)**. On n'a évidemment pas besoin d'éliminateurs si la suite est choisie en vertu de la combinaison qu'on veut obtenir, mais dans les applications il arrive souvent qu'on ait à obtenir plusieurs combinaisons sur un même ensemble de données, donc sur une même suite, qui comporte des éléments nécessaires pour une combinaison et non pour une autre.

Une fois trouvé le produit qui permet d'obtenir $\mathfrak{X}$, de deux choses l'une : ou bien pour tout terme *(X.Y)* du produit, *X* est régulier, comme dans les exemples du paragraphe précédent, ou bien ce n'est pas le cas, comme dans l'exemple suivant : soit à obtenir *aba* ; il vient **WI***ab* $\rightarrow$ **I***aab* $\rightarrow$ *aab*, **C***aab* $\rightarrow$ *aba* ; le produit est **(WI.C)**, et **WI** n'est pas régulier. Dans la première éventualité, le produit possède une expression combinatoire qui résulte de l'usage, éventuellement répété, de la formule $(X.Y) =_{df}$ **B***XY*.

	produit	expression combinatoire	
à un facteur	*X*	*X*	
à deux facteurs	*(X.Y)*	**B***XY*	expression dite *en* **B**
à trois facteurs	*(X.Y.Z)*	**B(B***XY***)***Z*	
	etc.		

Le combinateur *X'* qui figure dans la définition donnée de la complétude combinatoire est donc *X*, **B***XY*, **B(B***XY***)***Z*, etc.. Ainsi **(K.W.B)** devient **B(BKW)B** ; on vérifie que **B(BKW)B***abcd* $\rightarrow$ **BKW(B***a***)***bcd* $\rightarrow$ **K(W(B***a***))***bcd* $\rightarrow$ **W(B***a***)***cd* $\rightarrow$ **B***accd* $\rightarrow$ *a(cc)d*. Dans l'éventualité où pour un terme *(X.Y)* du produit, *X* n'est pas régulier, on ajoute devant la suite initiale (imposée ou non) et devant $\mathfrak{X}$ une indéterminée qui n'est pas dans $\mathfrak{X}$, disons $\dot{y}$; soit $\mathfrak{X}'$ la combinaison ainsi modifiée. On a donc : $\dot{y}\dot{x}_1 \ldots \dot{x}_n \rightarrow \dot{y}\mathfrak{X}$, où $\mathfrak{X}$ est une combinaison de $\dot{x}_1, \ldots, \dot{x}_n$. $\mathfrak{X}'$ peut s'obtenir par un produit de combinateurs réguliers puisqu'aucune récriture n'aura à affecter le premier élément, donc par une expression en **B**. Soit X'' ce produit ;

or, si	$X''\ \dot{y}\dot{x}_1 \ldots \dot{x}_n \rightarrow \dot{y}\mathfrak{X}$
alors	$X''\ \mathbf{I}\dot{x}_1 \ldots \dot{x}_n \rightarrow \mathbf{I}\mathfrak{X} \rightarrow \mathfrak{X}$

Comme X'' peut être formulé par une expression en **B**, cette expression appliquée à **I** et à la suite originale obtiendra $\mathfrak{X}$. Pour obtenir *aba* de *ab*, on pose donc *cab* → *caba*, on cherche le produit adéquat : **W***cab* → *caab* , **BC***caab* → *caba*, on obtient (**W.BC**), on conclut que *aba* s'obtiendra de *ab* par (**W.BC**)**I** ; en effet : **WI***ab* → **I***aab*, **BCI***aab* → **C**(**I***a*)*ab* → **I***aba* → *aba*, soit par l'expression en **B** correspondante : **BW**(**BC**)**I***ab* → **W**(**BCI**)*ab* → **BCI***aab* → **C**(**I***a*)*ab* → **I***aba* → *aba*. On ajoutera qu'on peut se contenter de la formulation du produit puisqu'on a fait la preuve qu'il existe une expression en **B** correspondante, que l'intérêt du produit est notamment de faciliter l'obtention des combinaisons et la preuve de complétude, qu'on connaît une autre formulation combinatoire que l'expression en **B**, souvent plus courte, mais souvent aussi plus difficile à trouver, dont le principe est le suivant : on se donne la combinaison à obtenir puis on cherche le combinateur et la suite de termes par lesquels elle peut être directement obtenue ; cette expression trouvée, on procède de la même manière à son égard, et ainsi de proche en proche jusqu'à la suite initiale convenable. On y parvient en utilisant (heuristiquement) les règles de récriture de manière converse, dans le sens inverse des flèches, c'est-à-dire comme des procédés d'expansion et non plus de réduction. En pratique, on commence par écrire les expressions intermédiaires, s'il en faut, entre la suite initiale, qu'on place à gauche, et la combinaison à obtenir, qu'on place à droite, puis on travaille comme indiqué, en "remontant" de droite à gauche. Soit par exemple à obtenir *ba* de *abc* ; on écrit : *abc* → *ab* → *ba*, on cherche un combinateur qui récrit *ab* en *ba*, il vient par exemple **CI** ; on cherche ensuite un combinateur X tel que $Xabc$ → **CI***ab*. Un éliminateur, et tel que **CI***ab* se conserve, c'est-à-dire agissant au cinquième rang, donne une solution. X est donc $\mathbf{B}^3\mathbf{KCI}$, ou, en mettant **CI** entre parenthèses pour gagner un rang, $\mathbf{B}^2\mathbf{K}(\mathbf{CI})$. On a bien : $\mathbf{B}^2\mathbf{K}(\mathbf{CI})abc$ → **CI***ab* → *ba*.

I, K, W, C, B forme donc un ensemble qui est combinatoirement complet, et qu'on dira *prototype*, mais **K, W, C, B** aussi car **I** peut être défini et donc remplacé par **WK**. On dit en effet que $X =_{df} Y$ si et seulement si le définissant Y est une combinaison de combinateurs donnés comme primitifs et qui possède la même règle de récriture que X, le défini (la même suite initiale la plus courte et la même récriture) ; ainsi **I***x* → *x* ; **WK***x* → **K***xx* → *x*. Les deux combinateurs **K,S** forment un système complet, tous les combinateurs du système prototype (autres que **K**) étant définissables par **K** et **S** : $\mathbf{I} =_{df} \mathbf{SKK}$, car **I***x* → *x*, **SKK***x* → **K***x*(**K***x*) → *x* ; $\mathbf{W} =_{df} \mathbf{SS}(\mathbf{SK})$, car **W***xy* → *xyy*, **SS**(**SK**)*xy* → **S***x*(**SK***x*)*y* → *xy* (**SK***xy*) → *xy* (**K***y*(*xy*)) → *xyy* ; $\mathbf{B} =_{df} \mathbf{S}(\mathbf{KS})\mathbf{K}$;
$\mathbf{C} =_{df} \mathbf{S}((\mathbf{S}(\mathbf{KS})\ \mathbf{K})\ (\mathbf{S}\ (\mathbf{KS})\ \mathbf{K})\ \mathbf{S})\ (\mathbf{KK})$.

c. Logique combinatoire et formation des mots.

Avec l'un des systèmes complets, il est possible de former un mot d'une langue naturelle (ou quelque autre unité) par un produit de combinateurs, donc aussi par un terme de L, à partir des lettres (ou des éléments) qui composent son étymon, si celui-ci comporte toutes les lettres du mot (ou tous les éléments de l'unité), après avoir doté l'ensemble E de L de toutes les lettres (ou de tous les éléments) utilisés. On engendrera par exemple *bon* de *bonus* par $(\mathbf{B}^2\mathbf{B}.\mathbf{B}^2\mathbf{K})$.

En effet : $\mathbf{B}^2\mathbf{B}\,bonus \rightarrow bon(us)$, $\mathbf{B}^2\mathbf{K}\,bon(us) \rightarrow bon$. Rappelons qu'on obtient exactement $((b * o) * n)$, mais $(b * o)$ exprimant que *bo* fait un tout, $((b * o) * n)$ que *bon* fait un tout, c'est bien là ce que marque l'écriture usuelle du mot.

La logique combinatoire ne suffit pas, évidemment, à obtenir, à partir d'une formation linguistique donnée, les seules formations linguistiques qui en sont issues étymologiquement. Pas plus qu'une autre théorie mathématique, fût-elle l'arithmétique, elle ne peut interdire d'effectuer des traitements inopportuns ou de s'adresser à des données qu'on peut juger trop faibles, d'être ici par exemple l'étude de certaines attractions paronymiques (et bien que ces dernières fassent aussi évoluer la langue). On objectera peut-être que si l'histoire des mots légitime d'obtenir *bon* de *bonus*, *âpre* de *asper*, on n'a plus besoin d'utiliser les combinateurs puisque c'est une transformation linguistique qu'il faut invoquer ; la disparition du *s*, par exemple, dans le français *âpre* ne provient pas, dira-t-on, de l'emploi d'un éliminateur sur le mot latin *asper* (en négligeant l'accent), ou du moins il n'est pas éclairant de le dire, elle s'explique par le fait que s'il y a deux consonnes intervocaliques, la première subit (ou au moins subit fréquemment) une dissimilation qui peut être totale. Toutefois, l'impression qu'on a affaire à de simples exercices est susceptible de provenir d'un usage encore fruste des combinateurs. On a pu montrer, en effet, que des opérations logiques comme la négation, l'implication, des opérations arithmétiques comme l'adddition, la multiplication, etc. sont exprimables par les seuls combinateurs, comme on le verra ci-dessous. Il n'est pas exclu que ce soit aussi le cas pour les transformations diachroniques de la linguistique et qu'elles parviendront à s'exprimer de manière moins rudimentaire dans le langage des combinateurs.

III. INTERPRETATIONS DE L.

D'une manière générale, les symboles de L sont aptes à recevoir de nombreuses interprétations. De même que des fonctions combinatoires, ou combinateurs, sont associées aux éléments de E écrits en majuscules, on associe souvent des fonctions (non combinatoires) et des arguments de fonction aux éléments de E écrits en minuscules. On conviendra hors système d'utiliser *f*, *g*... pour les fonctions non combinatoires et *a*, *b*,... pour les arguments. Le tout formé par $*$ et ses deux arguments sera en général celui que forment des expressions comme $f * a$, $\mathbf{I} * a$ (fa , $\mathbf{I}a$, en écriture simplifiée), c'est-à-dire correspondant à l'application d'une fonction unaire à son argument. La logique combinatoire s'exprime ainsi plus directement dans le langage des fonctions que dans celui des ensembles, et en comprenant la fonction comme une opération produisant un certain effet, non comme une simple correspondance, ou, si l'on préfère, avec une notion d'application conçue comme procédé d'obtention d'un objet par action sur un autre, non comme ensemble de couples[15].

Pour faire comprendre l'intérêt des combinateurs dans le traitement des fonctions, reformulons quelques règles de récriture sur *f*, *g*, fonctions, avec *x*, *y*, z, arguments de ces fonctions. Appliqué à une fonction *f* à deux arguments *x*,y, **C** obtient la converse : $\mathbf{C}fxy \rightarrow fyx$; $\mathbf{C}f$ désignera la converse de *f*. Appliqué à une fonction à plus de deux arguments, par (*MG*), **C** en permute les deux premiers : $\mathbf{C}fxyz \rightarrow fyxz$. On voit ainsi l'importance des combinateurs réguliers : ils effectuent des modifications utiles sur les arguments d'une fonction,

[15] On dit souvent que dans les sciences le concept de fonction supplante le concept de cause, et on veut dire par là qu'on reconnaît des liaisons entre phénomènes sans affirmer que les uns sont les producteurs des autres. Cela toutefois n'est vrai que si le concept de fonction lui-même est conçu de manière phénoméniste, c'est-à-dire au fond ensembliste. Cela n'est pas vrai au sens ancien du mot *fonction* que la logique combinatoire remet en honneur, et il y a sans doute sur le concept mathématique de fonction les mêmes options à prendre que celles qui opposent les phénoménistes et les rationalistes sur le concept de cause.

sans affecter la fonction elle-même qu'ils conservent. Appliqué à deux fonctions f, g, **B** forme $f(gx)$, c'est-à-dire le produit de composition classiquement noté $g \circ f$. Soit deux fonctions f, g telles que f est à deux arguments et g à un argument, alors $\mathbf{S}fgx \to fx(gx)$ substitue au deuxième argument de f une expression gx.

D'une manière très proche, la logique combinatoire permet de former des prédicats complexes à partir de prédicats donnés comme simples : à partir du prédicat *f, regarde*, par exemple, on formera le prédicat **W***f, se regarde*. $\mathbf{W}fx \to fxx$.

Il arrive d'ailleurs souvent que pour appliquer la logique combinatoire par exemple sous la forme donnée par L, il faille ajouter certaines constantes du domaine d'application, ici dans E, après avoir reformulé en notation combinatoire les expressions de ce domaine (c'est-à-dire préfixée, avec *(AG)*, etc.). Soit le système L_a appliquant L à une théorie mathématique utilisant les constantes d'addition, de multiplication, respectivemnt A et M en notation préfixée, et les constantes numériques *1*, *2* ; on peut s'exprimer sans variable dans L_a :

	Formulations		
usuelles	préfixées	combinatoires	preuves
x^2	Mxx	$\mathbf{W}M$	$\mathbf{W}Mx \to Mxx$
$x + x$	Axx	$\mathbf{W}A$	$\mathbf{W}Ax \to Axx$
$x + 1$	$Ax1$	$\mathbf{C}A1$	$\mathbf{C}A1x \to Ax1$
$x^2 + 2x$	$A(Mxx)(M2x)$	$\mathbf{\Phi}A(\mathbf{W}M)(M2)$	$\mathbf{\Phi}A(\mathbf{W}M)(M2)x \to A(\mathbf{W}Mx)(M2x) \to A(Mxx)(M2x)$

(les variables doivent être placées en fin d'expression pour pouvoir être éliminées)[16].

IV. ARITHMETIQUE COMBINATOIRE.

Toutefois, en procédant ainsi, on obtient seulement des représentations semi-combinatoires puisqu'on garde des constantes du domaine d'interprétation. L'objectif ultime est d'obtenir des représentations totalement combinatoires, c'est-à-dire d'éliminer les constantes non combinatoires en les exprimant par des combinateurs. Cela a été fait, notamment, pour + et × (A,M), ou plus exactement pour des opérations $\underline{+}$, $\underline{\times}$, ($\underline{A}$,$\underline{M}$), analogues aux opérations + et × sur les entiers mais définies sur des objets exprimables par la logique combinatoire et eux-

16 Ce qu'obtient l'élimination des variables, l'intérêt technique excepté, pose au philosophe des questions capitales. Si on peut écrire, par exemple, **C***r*, la converse d'une relation sans lui donner aucun argument, bien que l'effectuation en ait besoin, est-ce parce qu'il faut qu'une relation "se matérialise en quelque sorte pour qu'elle devienne visible mais n'est elle-même que dans la suppression de toutes ses matérialisations", comme le dit J. Ladrière (in *L'explication dans les sc.*, Flammarion, 1973, p. 54), ou bien est-ce parce qu'elle est elle-même dans l'une quelconque de ses matérialisations ? de ce qu'une réalité ne dépend pas des données qu'on lui soumet parce qu'elle convient toujours à d'autres, s'ensuit-il qu'elle existe ou subsiste, même d'une façon indicible, indépendamment de toutes ces données, si ce n'est pas là une simple manière de parler ? n'être ni dans l'une ni dans l'autre seulement, est-ce se trouver en dehors de toutes ? n'est-ce pas plutôt être à l'intérieur de chacune ? on peut d'ailleurs admettre qu'il n'y a qu'un seul couple de matérialisations possibles pour *r* ou **C***r*, une fois négligé ce qui différencie entre eux d'une manière non pertinente pour l'objectif, les couples possibles d'arguments.

mêmes analogues aux entiers. On assimile d'abord chaque entier *0, 1, 2, ...* à un combinateur, dit *numérique*, qu'on notera **0, 1, 2,....** , caractérisés par les règles de récriture suivantes :

$\mathbf{0}xy \to y$	**0** applique zéro fois x à y , il reste y
$\mathbf{1}xy \to xy$	**1** applique une fois x à ce que produit **0**
$\mathbf{2}xy \to x(xy)$	**2** applique une fois x à ce que produit **1**
$\mathbf{3}xy \to x\,(x(xy))$	**3** applique une fois x à ce que produit **2**
soit $(\mathbf{n} + 1)\, xy \to x(\mathbf{n}xy))$	

On peut alors définir *successeur de* par un combinateur, à savoir **SB**. En effet, le successeur de $\mathbf{n}xy$ est $x(\mathbf{n}xy)$ par hypothèse ; or, $\mathbf{SBn}xy \to \mathbf{B}x(\mathbf{n}x)y \to x(\mathbf{n}xy)$; **0** étant **KI** car $\mathbf{KI}xy \to \mathbf{I}y \to y$, **1** sera **SB(KI)**, **2** sera **SB(SB(KI))**, etc. On démontre, entre autres, que $\underline{A}$ est **ΦB**, que $\underline{M}$ est **B**[17].

A son point ultime d'aboutissement, la logique combinatoire doit devenir, cependant, un système axiomatique, soit de la théorie pure des combinateurs, soit appliqué à tel ou tel domaine. On va donner ici une idée du premier.

V. ELEMENTS D'AXIOMATIQUE.

On pose comme axiomes certaines égalités comme **BI = I** qu'on sait par ailleurs être des lois. On ajoute des règles de déduction pour lesquelles on possède la preuve qu'appliquées à des lois, elles obtiennent des formules qui sont des lois, par exemple :

$RD_1 \quad x = y \;\; : \;\; y = x$	$RD_2 \quad x = y\, ,\, y = z \;\; : \;\; x = z$
$RD_3 \quad x = y \;\; : \;\; xz = yz$	$RD_4 \quad x = y \;\; : \;\; zx = zy$

A propos de RD_1, la preuve est triviale : si $x = y$ est une loi, $y = x$ est une loi (si x obtient la même récriture que y, y obtient la même récriture que x). On démontre alors des théorèmes comme **BI = BI**, **BI(KW) = I(KW)** :

(1) **BI = I**	axiome	(1) **BI = I**	axiome
(2) **I = BI**	(1) par RD_1	(2) **BI(KW) = I(KW)**	(1) par RD_3
(3) **BI = BI**	(1) (2) par RD_2		

On sait produire une axiomatique complète et consistante pour l'ensemble des lois de L. Dans une telle présentation, les variables sont éliminées des axiomes et des règles comme dans la théorie intuitive des combinateurs, ou, dans le système lui-même, par hypothèse. Mais elles sont

---

[17] L'entreprise rappelle à certains égards le "logicisme" de Whitehead et Russell (telle théorie mathématique peut être reformulée, ce qui ne veut pas dire découverte, en utilisant les seuls moyens de la logique, de la logique combinatoire présentement, et même si une mathématique forte est requise pour élaborer la métathéorie). On notera que la logique combinatoire est loin de se borner à supprimer les variables. Elle entend aussi, et principalement, supprimer les constantes usuelles, logiques et mathématiques, au profit de ces constantes particulières que sont les combinateurs, et en raison des propriétés immédiatement compréhensibles de ceux-ci, sans doute parce qu'elles sont moins ontologiques que praxéologiques. Il semble plus facile de dire ce qu'est éliminer, dupliquer, permuter, etc. que d'analyser par exemple le mode d'existence d'une fonction récursive.

éliminées des théorèmes en un sens beaucoup plus fort puisqu'on engendre directement par les règles de déduction du système des formules sans variables.

Si on prolonge ces investigations en élaborant un système axiomatique de logique combinatoire appliquée, et sans constante non combinatoire, on atteint sans doute le niveau le plus élevé d'une application des techniques qu'on vient de présenter. Cette dernière approche - très ambitieuse - reste au moins un idéal du programme défendu par la logique combinatoire.

VI. THEORIE DE LA FONCTIONNALITE.

Pour terminer nous donnerons un aperçu de la théorie de la fonctionnalité qui constitue une entreprise bien distincte de celle qui précède puisqu'il s'agit d'effectuer la classification des entités en catégories, alors qu'elles ont été jusqu'ici aussi peu différenciées que possible, traitées seulement dans leurs propriétés les plus générales d'objets. La méthode consiste à se donner des catégories primitives (on dit aussi des *types primitifs*, mais sans dépendance à l'égard de la théorie russellienne) puis à engendrer toutes les autres catégories d'un domaine à partir d'elles, grâce à un opérateur F (qui n'est pas un combinateur) tel que :

(1) si α et β sont des catégories, $F\alpha\beta$ est une catégorie
(2) si $xy \in \alpha$ et si $y \in \beta$ alors $x \in F\beta\alpha$

xy étant une expression écrite en langage applicatif, α, β des métavariables de catégories. On peut aussi formuler (2) par $\dfrac{xy \in \alpha \qquad y \in \beta}{x \in F\beta\alpha}$ ou par $xy \in \alpha \begin{array}{l} \nearrow y \in \beta \\ \searrow x \in F\beta\alpha \end{array}$

ou encore par $xy \in \alpha, y \in \beta \vdash x \in F\beta\alpha$ que nous utiliserons. Nous appellerons (F) cette règle. On comprendra $x \in F\beta\alpha$ comme signifiant *x appartient à la catégorie $F\beta\alpha$ des objets dont chacun appliqué à un objet de la catégorie β détermine un objet qui appartient à la catégorie α*. En bref, x transforme un β en un α.$F\beta\alpha$ exprime le caractère fonctionnel d'une expression, c'est-à-dire son type de fonction. Ainsi un opérateur unaire dont l'opérande appartient à la catégorie β et la résultante ou la valeur à la catégorie α a le caractère fonctionnel $F\beta\alpha$, appartient à la catégorie $F\beta\alpha$. Inversement, si on connaît le caractère fonctionnel de x et la catégorie à laquelle appartient y, on peut trouver la catégorie à laquelle appartiennent xy par la règle suivante : $x \in F\beta\alpha$; $y \in \beta \vdash xy \in \alpha$. Au lieu de formules comme $x \in \alpha$, on écrit aussi $\alpha x, \alpha : x$, et on peut lire ces trois formules *x est un α*.

Soit donc la catégorie s des propositions, supposée donnée, dans le calcul des propositions. On sait que $Np \in s$ et que $p \in s$. On peut donc par (F) déterminer à quelle catégorie appartient N. Il vient : $Np \in s$, $p \in s \vdash N \in Fss$. Autrement dit, N appartient à la catégorie Fss, c'est-à-dire à la classe des fonctions (de vérité) dont l'argument unique et l'expression toute entière appartiennent à la même classe. On obtient d'une manière comparable la catégorie des fonctions de vérité binaires, par exemple en raisonnant sur D, et en lisant Dpq selon (AG) comme $(Dp)q$, par un double emploi de (F) : $Dpq \in s, q \in s \vdash Dp \in Fss$, $p \in s \vdash D \in FsFss$. Un opérateur propositionnel traité usuellement comme binaire, tel que D, peut être traité (selon la méthode de Schönfinkel) comme un opérateur unaire appliqué à une proposition comme p, c'est-à-dire Dp, qui , appliqué à une proposition comme q, produit une proposition, Dpq. Soit à déterminer la catégorie de **I**, en posant qu'une expression et sa récriture appartiennent à la même catégorie, disons u : $\mathbf{I}a \in u$, $a \in u \vdash \mathbf{I} \in Fuu$; Fuu est aussi la catégorie de $\mathbf{I}^2$, $\mathbf{I}^3$, etc. mais non celle de **K** puisqu'on aurait : $\mathbf{K}ab \in u$, $b \in u' \vdash \mathbf{K}a \in Fu'u$, $a \in u \vdash \mathbf{K} \in FuFu'u$. Bien que **I** et **K** soient l'un et l'autre des combinateurs, on peut donc, à certains égards, les attribuer à des catégories différentes. On comprend qu'on puisse affiner de la même manière (ou bouleverser) les catégories grammaticales de verbe, d'adverbe, d'adjectif, etc. dont la généralité, parfois souhaitable, masque aussi des différences importantes et commet des confusions. Or, la détermination formelle des catégories sémantiques, est nécessaire pour qu'on puisse traiter strictement du sens d'un énoncé. Elle intéressa à la fois Husserl et les néo-positivistes. Elle doit sans doute intervenir dans les travaux sur la traduction automatique. Pour cette question, comme pour beaucoup d'autres, la logique combinatoire apparaît au carrefour d'investigations mathématiques, linguistiques et philosophiques.