Jean-Baptiste Joinet
Ouvrir la logique au monde[1]
54 pages

Avertissement :

L'article qui suit est paru en 2009, dans un volume collectif intitulé « Ouvrir la logique au monde – philosophie et mathématique de l'interaction » (dir. Jean-Baptiste Joinet & Samuel Tronçon), éditions Hermann.

En raison d'une erreur de l'éditeur lors de l'ultime compilation avant l'impression (non repérée au moment du « bon à tirer »), cet article dont le titre original était « Ouvrir la logique au monde » (donnant son titre au volume) est en fait paru sous le titre erroné « Introduction » et sans mention en début d'article du nom de son auteur, Jean-Baptiste Joinet.

Loin d'être une introduction présentant le volume et les articles qu'il contient (présentation réalisé dans le « préambule » du volume), il s'agit en fait d'un article à part entière visant à faire une synthèse des idées logico-philosophiques qui sont à l'origine des travaux du groupe interdisciplinaire LIGC (interfaces philosophie-sciences) depuis sa création en 2003.

---

[1] Article paru dans « Ouvrir la logique au monde – philosophie et mathématique de l'interaction » (dir. Jean-Baptiste Joinet & Samuel Tronçon), collection *Visions des sciences,* Éditions Hermann, Paris 2009, p.9-63 (*sous le titre erroné « Introduction »)*

Introduction

Quoiqu'ouvert sur des champs très variés de la science contemporaine (physique, biologie, linguistique, notamment), le groupe *Logique et Interaction : vers une Géométrie de la Cognition* (LIGC) reste marqué par ce qui, chronologiquement parlant, fut la source d'inspiration première de ses investigations philosophiques, à savoir les développements entrecroisés, parfois tumultueux mais souvent complices, de la logique mathématique et de l'informatique théorique tout au long du XX^{e} siècle – et tout particulièrement ceux postérieurs à ces années charnières que furent les années 1970.

Dans cet article introductif, mon objectif est d'expliquer en quoi l'ouverture pluridisciplinaire de LIGC, loin de simplement résulter du souci, par ailleurs louable, de promouvoir une philosophie plus vivante à l'écoute des sciences contemporaines, est en réalité le produit nécessaire de cette inspiration première, celle-ci justifiant et organisant la pluridisciplinarité sur la base d'une proposition *transdisciplinaire* originale.

A cette fin, je présenterai d'une part la révolution scientifique qui a marqué la logique mathématique au cours des trente dernières années sous l'effet de son dialogue avec l'informatique fondamentale, d'autre part je tenterai d'expliquer en quoi en affinant notre compréhension du lien de la logique aux formes abstraites du calcul et de l'interaction et donc, par transitivité, aux conditions mathématiques et physiques de ces dernières, à leur réalisation dans les phénomènes interactionnels

de nature diverses et à leurs limites, ces développements invitent à réévaluer en profondeur le statut épistémologique de la logique et le lien de la logique aux sciences.

Parce que les acteurs de cette « révolution scientifique » furent sans doute trop occupés à l'accomplir pour prendre le temps de la nommer, celle-ci n'a pas encore été à ce jour « labélisée ». Parmi diverses appellations qui ont été envisagées dans la période récente, notamment dans le cadre des activités du collectif LIGC, pour qualifier le paradigme émergent, celle de paradigme *interactionniste*[1] me semble efficace et essentiellement juste – et je continue aujourd'hui à défendre cette dénomination.

Que la transformation évoquée de la logique mérite d'être qualifiée de « révolution » ne va bien sûr pas de soi. Ce qui, à mes yeux, rend cette qualification légitime, c'est justement le fait que les transformations évoquées impliquent, nous le verrons, un changement radical de point de vue sur la logique elle-même, sur l'objet de la discipline, ses critères, son statut et – c'était l'objet même de la rencontre de Cerisy qui a donné naissance à ce volume – ses relations aux autres sciences.

Pour souligner la radicalité du changement accompli, l'invocation d'événements déclencheurs (en l'occurrence, mais j'y reviendrai, les *deux piliers* de cette recomposition que constituent, d'une part, la refondation sémantique opérée vers 1969 dans le contexte de l'informatique fondamentale, inspirée par les travaux de Dana Scott et Christopher Strachey, d'autre part la mise à jour à peu près simultanément de la correspondance de Curry-Howard), donc d'un *moment inaugural* de la rupture, a d'évidentes vertus.

Nul doute que l'invocation d'événements *fondateurs* doive être considérée, au plan historique, avec prudence. Simplification régulatrice, une telle invocation peut en partie brouiller les pistes de

[1] Terminologie justifiée page 26 du présent article. Cf. aussi mon éditorial du 28/02/2004, en ligne sur le site web du collectif LIGC, intitulé *Programme de l'interactionnisme* et repris pour l'essentiel dans la préface de [25].

l'inscription complexe du nouveau paradigme dans le temps, dans sa relation au passé d'une part – car la recomposition qui mène à l'instauration du paradigme interactionniste n'implique pas l'absolue nouveauté (certaines approches antérieures dans le cadre de l'histoire de la logique comme dans la tradition philosophique anticipent certains traits du nouveau paradigme) ; mais aussi dans son inscription effective dans la durée, postérieurement au « moment inaugural », en plusieurs étapes. En l'occurrence, ces « événements » furent très tôt considérés comme des repères clefs par la petite communauté des spécialistes des fondements logiques de l'informatique ; c'était en tout cas patent, dans le dernier tiers des années 1980.

Qu'ils aient été ou non inauguraux, on peut donc dire en tout cas de ces repères qu'ils jouèrent le rôle de catalyseurs de l'imagination scientifique, qu'ils furent les sources de l'accélération du renouvellement du questionnement (la recomposition des théories et la stabilisation des systèmes de représentation étant quant à elles intervenues plus tardivement).

Afin, dans un premier temps, de présenter à un public large ce nouveau paradigme vers lequel la logique a basculé et de faire comprendre la nature du changement opéré, je consacrerai une section, intitulée *la refondation sémantique*, au « premier pilier » de cette recomposition. Il s'agira de montrer comment les travaux en « sémantique dénotationnelle » des langages de programmation accomplis dans le contexte de l'informatique théorique impliquent une critique *radicale*, au sens propre, de la théorie dite « sémantique » proposée vers le milieu du XX^e^ siècle (A. Tarski) dans le cadre fourni à la logique mathématique par la théorie des ensembles, et plus précisément de sa pertinence en mathématiques en tant que théorie « sémantique ».

Dans une seconde section, intitulée *La Logique au delà du langage*, je propose une évaluation des conséquences de la « découverte » de la correspondance dite « de Curry-Howard » – le « second pilier » – quant au statut et aux critères du logique. Ces questions sont examinées et discutées à la lumière du rôle de la logique dans

la *dynamique interactionnelle* de l'évaluation du sens : simple norme extérieure d'apprivoisement d'une dynamique généralement sauvage (idée suggérée par la correspondance de Curry-Howard) ou norme intrinsèque, émergeant des propriétés nécessaires de l'interaction (comme le montre les derniers rebondissements du « programme de recherche », qui s'est développé depuis les années 1980, sous l'inspiration non exclusive mais déterminante des idées de Jean-Yves Girard).

La dernière partie vise à tirer les enseignements de ce qui précède quant à la nature ou au statut de la logique afin de mesurer les implications de cette nouvelle définition sur les liens de la logique aux sciences.

La refondation sémantique

Autour de 1966, Dana Scott et Christopher Strachey publient les premiers travaux de ce nouveau champ qu'on appelle encore aujourd'hui la « sémantique dénotationnelle »[2]. La recomposition dans laquelle s'est engagée la logique une quinzaine d'années plus tard et qui mènera au *paradigme interactionniste,* trouve dans ces investigations sémantiques – dans leur problématique plus que dans leurs résultats – l'une de ses sources principales. Si ces travaux en sémantique rompent de façon radicale avec l'idée de sémantique que la logique des deux premiers tiers du siècle avait léguée, et si, comme nous le verrons dans la section suivante, ils se sont ensuite diffusés, vers la fin des années 1970 et surtout dans les années 1980, vers la logique, c'est dans le cadre de l'*informatique théorique* à proprement parler qu'ils ont vu le jour.

[2] Pour une rapide exposition historique de l'émergence des travaux en « sémantique dénotationnelle » et une réévaluation des contributions de Christopher Strachey [35] dont peu sont publiées, on peut consulter [32], [34], [7] et [5].

Les retranchements du sens

Sans entrer dans une présentation technique du programme scientifique de la « sémantique dénotationnelle », on peut dire qu'il s'agit non pas tant d'une reformulation de la question sémantique elle-même (le nom même de l'entreprise : « sémantique dénotationnelle » indique assez la filiation frégéenne, autrement dit la reconnaissance de la distinction inaugurale entre *sens* et *dénotation* héritée de Frege), que comme une critique radicale de la réponse que, concernant les énoncés mathématiques, la logique formelle avait proposée dans la première moitié du siècle (sémantique ensembliste dite à la Tarski pour les langages du calcul des prédicats), réponse tarskienne qui, comme nous allons le voir maintenant, prend littéralement *à contresens* la relation du sens à la dénotation en mathématiques.
De même qu'on peut dire, en reprenant la distinction inaugurale introduite par Frege entre *sens* et *dénotation,* ici exclusivement appliquée au langage mathématique[3] que relativement à l'interprétation standard, l'entier 6 est la dénotation commune d'expressions mathématiques distinctes n'ayant pas le même sens (comme par exemple 2×3 et $4 + 2$), ou encore (pour reprendre cette fois l'expression « mode de donation » également proposée par Frege) que :

l'entier 6 *admet des modes de donation mathématiques*[4] *distincts*

de même, en poussant cette distinction dans ses retranchements, aimerait-on pouvoir dire par exemple que :

la fonction $\times$ *admet des modes de donation mathématiques distincts.*

En effet, non seulement de tels modes distincts de donation de la multiplication nous sont connus (il nous suffit de considérer

[3] Repoussant à plus tard (cf. page 21, notamment) des développements sur le statut ontologique des entiers naturels, des fonctions, etc ?

[4] L'adjectif « mathématiques » vise ici seulement à écarter les « modes de donation de l'objet » peu pertinents dans la perspective de cet article (comme par exemple : « le chiffre préféré de Jean »).

deux programmes informatiques réalisant des algorithmes distincts implémentant cette fonction), mais surtout, dans les faits, ce n'est qu'à travers eux que la multiplication nous est connue (la multiplication, comme graphe infini actuellement donné *in extenso* ne tient en tout cas dans aucune de nos bibliothèques).

Programmes et algorithmes

Avant même d'aborder les problèmes de fond que posent, dans le cadre d'une théorie sémantique du langage mathématique, toute tentative de prendre au sérieux la distinction frégéenne en la propageant ainsi au delà de la notion ensembliste de fonction, vers l'informatique, il importe de souligner que la distinction qui vient d'être faite entre « programme » et « algorithme » ajoute d'emblée une difficulté, qui pourrait bien n'être que superficielle, mais qui, néanmoins, vient très concrètement brouiller le jeu du sens et de la dénotation.

La question est celle de savoir si les deux distinctions induites (celle entre *programme* et *algorithme* d'une part, celle entre *algorithme* et *fonction* d'autre part) sont également pertinentes du point de vue de la distinction entre sens et dénotation. Admettant que deux programmes informatiques normaux distincts réalisent un même algorithme, nous pouvons à un premier niveau considérer que le « sens » de ces programmes est distinct (quoique tous deux dénotent le même algorithme[5]), ou à un second niveau, adopter une attitude « déflationniste » en considérant que la différence littérale entre nos deux programmes est superficielle, purement nominale, que la multiplicité des signifiants (au sens saussurien) n'implique pas en l'occurrence la multiplicité du sens, mais qu'elle résulte simplement et pour ainsi dire d'un *défaut de fabrication* de notre langage de programmation (mal conçu, trop bavard, puisqu'il nous permet de « dire » de deux manières différentes – quoiqu'isomorphes par

[5] Deux algorithmes pouvant alors apparaître à leur tour, comme modes de donation distincts d'une même fonction

delà les apparences – un « sens » unique, à savoir ce seul et même algorithme dénoté[6]), voire une limite congénitale de tout *langage* de programmation – les contraintes contingentes liées à la *textualité* du programme venant brouiller les traces d'une géométrie plus profonde cachée sous la surface du texte, dans sa structure[7].

Ce monde (doit-on dire idéal, doit-on dire réel ?) où les langages de programmation sont si épurés, si dégagés des contingences des langages de programmation particuliers, si proches des structures abstraites *a priori* du calcul, qu'ils finissent par ne plus être au sens habituel « des langages », c'est justement, nous le verrons, celui que la théorie logique des fondements de l'informatique, à travers les travaux qui sont abordés ou évoqués dans le présent article, a cherché à reconstruire – non sans succès, au demeurant[8], depuis les annéees 1970.

Misère du point de vue extensionnel ensembliste

Prendre au sérieux la distinction frégéenne entre sens et dénotation impose donc, nous l'avons vu, d'étendre la question du sens (telle qu'elle se pose par exemple pour une expression linguistique comme « 2×3 ») à ces autres expressions linguistiques que sont les programmes informatiques, ces voies d'accès aux algorithmes[9].

[6] Cette attitude « déflationniste », correcte dans son principe et communément adoptée dans ce champ de recherche, est en fait troublée dans le détail par l'instabilité de la notion d'algorithme. Quoique très usitée (et quoique des définitions standard en soit fréquemment proposées, ne serait-ce que dans les ouvrages pédagogiques), la notion d'*algorithme* reste en un sens mal définie : une grande partie des travaux contemporains d'informatique théorique visent d'ailleurs indirectement à formuler un cadre adéquat pour produire des critères d'*identité entre algorithmes*.

[7] Cf. le passage, p. 3 de [21], sur les « imperfections de la syntaxe » qui forment comme un voile sur les « symétries du sens ».

[8] Cf. les théorèmes de *full completeness* en ludique [16].

[9] Voies d'accès imparfaites certes, comme on vient de le remarquer, mais jusqu'à plus ample informé incontournables pour une première approche théorique de la question, en raison des qualités phénoménales du langage écrit : stabilité,

Étendre la question sémantique à ces textes que sont les programmes informatiques, peut-il pour autant signifier de leur appliquer la réponse « traditionnelle » élaborée par la logique à la fin du XIX^e siècle et au long de la première moitié du XX^e siècle ?

Dans le cadre de cette réponse « traditionnelle »[10], la « sémantique ensembliste » (et plus précisément pour notre exemple, la sémantique des termes du calcul des prédicats du premier ordre) le fait que les termes $2 + 4$ et 2×3 n'aient pas, dans l'interprétation standard, le même sens quoiqu'ils aient même dénotation, est pour ainsi dire défini comme le fait que les deux procédures de recherche (on n'ose dire de calcul) de la dénotation sont distinctes : dans le premier cas, c'est le graphe de l'addition qui doit être *parcouru* (à la recherche du triplet $(2, 4, 6)$), dans le second cas, c'est le graphe de la multiplication (à la recherche du triplet $(2, 3, 6)$).

Autrement dit, l'appartenance au graphe ne laisse aucune place à

caractère discret, structure. Si les programmes sont usuellement rédigés dans des *langages* de programmation, et en complément de la remarque faite plus haut sur la « géométrie » intérieure du programme à l'étroit dans sa forme textuelle, observons qu'il n'est pas clair que la *mise en œuvre* d'un algorithme implique un recours au langage *stricto sensu*. Pour prendre un exemple simple, je peux effectuer la division entière de ce tas de cailloux par n, en les distribuant un à un, circulairement, en n tas. Pour une discussion contemporaine de cette question de l'implémentation ou de l'incarnation non linguistique des algorithmes dans la nature, voir [10] (notamment le chapitre 5), les travaux de F. Bailly et G. Longo ([3] et [2]), sans oublier ceux de G. Dowek d'une part et de G. Longo et Th. Paul d'autre part dans le présent volume. Cette question fut par ailleurs au cœur du colloque intitulé « La thèse de Church : hier, aujourd'hui, demain » (`www-philo.univ-paris1.fr/Joinet/ChurchAccueil.html`) que j'ai organisé le 11 avril 2008 à la Sorbonne, dont la publication des actes est projetée.

[10] On attribue généralement à l'article de 1933 de Tarski, la définition de la notion de « vérité d'un énoncé du premier-ordre relativement à une interprétation ensembliste du langage ». Soit dit en passant, l'idée de la sémantique ensembliste est en 1935 manifestement suffisamment claire aux yeux de G. Gentzen pour qu'il souligne la correction des règles de la déduction naturelle en termes de préservation de la vérité, sans référence à l'article de Tarski et sans qu'elle lui paraisse nécessiter de plus amples développements...

l'idée que le triplet $(2, 3, 6)$ puisse être élément du graphe de l'addition *de différentes façons* (en fonction des algorithmes de calcul utilisés, autrement dit en fonction de la production concrète, dans le temps et l'espace, de 6 à partir de 2 et 3).

Le principe d'extensionalité (l'axiome selon lequel deux ensembles qui ont les mêmes élements n'en sont qu'un) empêche de distinguer les « fonctions » autrement que par le résultat de l'évaluation (quelles sorties, pour quelles entrées), autrement dit rend aveugle au phénomène calculatoire et à la multiplicité des *sens* qu'il recèle, ne laissant apercevoir que le minuscule épiphénomène de l'évaluation que constitue la trace extensionnelle des entrées et des sorties.

D'un autre côté, l'ontologie ensembliste est une telle corne d'abondance, qu'en première approximation, on pourrait être tenté, ayant pris acte de la précédente remarque, de réagir tout simplement en décidant de modifier la dénotation : d'y garder dorénavant trace du calcul, soit en somme de « coder le devenir dans l'être ». En effet, rien ne nous empêche *a priori* de « trouver » dans l'univers si peuplé des ensembles, un ensemble qui encodera, représentera cette évaluation discrète dont les traces (les étapes successives) étaient, disions-nous, invisibles tant qu'on s'en tenait aux graphes. Et rien ne nous empêche d'appeler *processus*, d'appeler *devenir*, l'ensemble ordonné des étapes comme bureaucratiquement ainsi consignées *in extenso*, quitte à finalement décréter que l'ensemble de ces suites d'étapes est la « sémantique » disons « opérationnelle » recherchée.

Cette option en quelque sorte « behavioriste », identifiant « l'action » d'un programme (lorsqu'il est plongé dans un contexte d'entrées particulières, et ce, pour tout contexte) à l'extension de ses effets (l'ensemble, « image après image », des états successifs, au long de l'évaluation, du texte que constitue le programme appliqué à ses entrées) serait cependant infertile au plan théorique. Non seulement parce que les comptes rendus bureaucratiques, inflationnistes par nature, mènent à la dévaluation sémantique. Mais surtout parce qu'elle reproduit l'erreur conceptuelle à la racine de l'approche

« traditionnelle » (erreur de conception quant à la nature même de la sémantique), à savoir de construire *à l'envers* la relation du sens et de la dénotation.

Sens et dénotation à contresens

Le propre de l'approche « sémantique » traditionnelle est en effet de fonder, à contresens, le sens sur la dénotation. Pour ce point de vue sémantique traditionnel, comme on l'a vu plus haut, la raison pour laquelle le sens de $2+4$ et celui de 2×3 sont distincts, par exemple, c'est que l'évaluation du sens de l'une et l'autre expressions (relativement à l'interprétation standard) passe par l'examen d'ensembles distincts, à savoir, respectivement, celui dénoté par $+$ et celui dénoté par $\times$: quoiqu'elle ne s'y réduise pas, l'« évaluation » du sens d'une expression suppose donc, pour cette approche sémantique, la donnée préalable de la dénotation des composants de cette expression. La dénotation est ici première, le sens second[11].

Fonder, à l'inverse, *la dénotation sur le sens*, c'est l'objectif du programme de refondation de la sémantique lancé vers la fin des années 1960, sous le nom de « sémantique dénotationnelle ». Prenant comme point de départ la dynamique calculatoire des programmes informatiques, il s'agissait de faire émerger à partir de la diversité de cette dynamique (mais cette fois comme produit second, abstrait), la dénotation que le sens recèle[12], autrement dit de retrouver la structure, les régularités et les invariants comportementaux individuant les algorithmes, qui alors seulement peuvent s'instituer comme véritable ontologie mathématique[13].

[11] Dans la suite, je qualifierai de *référentialistes*, les théories « sémantiques » fondant le sens sur la dénotation.

[12] « the sense contains the denotation » [21], p .3

[13] On appelle « sémantique dénotationnelle » des programmes, ce champ de travaux visant la détermination d'une dénotation recelée par le sens. Le corpus des travaux dans ce domaine, devenu très vaste avec le temps, est relativement peu connu, semble-t-il, des cercles philosophiques. La (désormais longue et riche) tradition des

En conclusion, prendre au sérieux la distinction frégéenne entre sens et dénotation suppose donc non seulement de propager cette distinction aux expressions linguistiques que sont les programmes, mais surtout de renoncer à seulement aménager la réponse (*i.e.* à étendre à ces expressions le traitement sémantique ensembliste « traditionnel ») au profit d'une reformulation de la question elle-même, bref d'entreprendre une refondation de la question sémantique, autour de l'objectif d'une construction de la dénotation *à partir du sens*, et non l'inverse.

Quand faire, c'est dire

En récusant le point de vue référentialiste, en abandonnant la fiction inopérante d'une dénotation constituant la donnée préalable sur laquelle la construction du sens pourrait s'appuyer, nous avons en particulier écarté, à propos de la dynamique de l'exécution des programmes, l'idée « behavioriste » d'un « "encodage" du devenir dans l'être ». Pour construire une alternative à la proposition sémantique référentialiste, il fallait donc reprendre la question à la racine, repartir du « devenir » (la dynamique de l'évaluation) et reconsidérer le parti pris *opérationnel* réalisé dans le contexte informatique par les programmes, ces expressions formulées dans des langages de programmation, « langages » nativement élaborés en vue de la *production d'effets* calculatoires.

Pour le présenter à l'aide d'une formule efficace, on peut dire que, du point de vue opérationnel, la détermination de la signification d'un terme (d'un programme) ne dépend plus, en tout cas en première instance, de la question « à quoi réfère-t-il ? », « que désigne-t-il ? », mais

travaux en sémantique dénotationnelle commence en 1969, avec les travaux de Dana Scott [33] qui proposa le premier modèle dénotationnel pour le lambda-calcul, un langage de programmation très épuré (mais fonctionnellement complet au sens de la thèse de Church : on y peut programmer toutes les fonctions récursives partielles, voir page 25 du présent article), conçu par Alonzo Church, quarante ans plus tôt, dans les débuts de la théorie du calculabilité, et dont nous allons reparler bientôt.

de la question préalable « que fait-il ? », « comment agit-il ? ».

Bien entendu, la manière dont cette « action » d'un programme se réalise effectivement dépend « dans la vraie vie »[14] de toute une série de facteurs concrets contingents qui brouillent le phénomène et qu'il fallut d'emblée commencer par écarter comme non pertinents pour aborder la question posée.

C'est la première raison pour laquelle ce fut d'emblée vers le langage de programmation le plus épuré (le lambda-calcul[15]) que la question fut d'abord posée. Mais outre sa structure particulièrement simple (il comprend, en tout et pour tout, trois « instructions de programmation »), le lambda-calcul offre un second avantage pour une théorie de l'opérationalité calculatoire.

Contexte et cotexte

Le lambda-calcul, en effet, est un « système de réécriture »[16]. Le trait des systèmes de réécriture qui nous importe, eu égard à la question sémantique pousuivie, peut être utilement présenté en recourant à la distinction (introduite par l'approche pragmatiste en philosophie du

[14] Celle où les langages de programmation ont été élaborés et perfectionnés au gré d'innovations ou de simplifications successives souvent désordonnées, mais en outre où leurs implémentations concrètes vont dépendre de facteurs physiques et systémiques extrêmement nombreux et hétérogènes.

[15] Cf. quelques mots de présentation page 19 et un bref développement page 25.

[16] Autrement dit, la définition du lambda-calcul commence par une double donnée : d'une part celle de la grammaire qui définit les termes (les programmes), d'autre part celle d'une règle de « réécriture » de ces termes en d'autres termes, qui joue le rôle de d'exécution, de l'évaluation. Le nom technique de cette relation de réécriture est la β-réduction (notée $\rightsquigarrow_\beta$). On parlera indifféremment de réduction, de normalisation, de réécriture, d'évaluation, d'exécution, de conversion, de dynamique.... Le lambda-calcul est un calcul de termes d'ordre supérieur (avec lieurs de variables). Il fut conçu par Church comme une « théorie naïve des fonctions » : une théorie où « tout est fonction ». Les termes du lambda-calcul sont aussi appelés « lambda-termes » ; l'ensemble des lambda-termes est noté Λ.

langage naturel) entre « contexte » et « cotexte »[17] : dans un système de réécriture, l'*action* des termes est toute entière définie comme *action sur le cotexte*[18], autrement dit non pas comme instruction communiquée à une machine, mais comme *action du langage sur le langage*[19].

L'exemple très « parlant » des entiers naturels du lambda-calcul (entiers de Church) illustre bien cette idée d'une sémantique fondée sur l'action sur le cotexte : l'entier n y est représenté par un terme (un programme) tel que, s'il est appliqué à un cotexte formé d'une fonction (en fait n'importe quel programme, mais qu'on peut utilement se représenter comme une fonction) et d'un argument (n'importe quel programme également), il *agira sur ce cotexte* en itérant n fois cette fonction sur cet argument. Avec des mots simples, on peut donc décrire l'*être actionnel* des entiers comme suit : les entiers sont des *itérateurs*. De la construction ensembliste des entiers comme ensembles

[17] Dans le cadre pragmatiste, le *contexte* englobe tout ce qui est extérieur au langage lors de l'acte d'énonciation « en situation », tandis que le *cotexte* est pour ainsi dire le complémentaire *textuel* de l'énoncé considéré (ou plus généralement le complémentaire du signifiant, au sens saussurien, en cas de discours non écrit). Pour une analyse synoptique des problèmes que posent la notion de contexte (et ses variantes) et la distinction contexte / cotexte dès qu'on la projette (hors du cadre mathématico-informatique) sur les théories du langage naturel et notamment des « actes de langage », voir l'article de J. Benoist [4]. Signalons également à cet égard les travaux de l'équipe *Prélude : vers une pragmatique théorique basée sur la ludique et les continuations* (« programme blanc » de l'Agence Nationale de la Recherche dirigé par Alain Lecomte ; `http ://anr-prelude.fr`) centrés sur l'analyse de l'interaction cotextuelle dans les langues naturelles. L'influence du pragmatisme sur la problématique sémantique en informatique, moins sensible aujourd'hui, semble l'avoir été dans les premières années de développement du domaine, cf. [37].

[18] Dans le vocabulaire technique de l'informatique toutefois, le « cotexte » est en fait appelé « contexte », voire « environnement ». Aussi pourra-t-il m'arriver, dans la suite du présent article, d'utiliser par inadvertance le mot « contexte » au sens de « cotexte »...

[19] Pour se prémunir contre les accusations de nominalisme qu'une telle proposition pourrait inspirer, insistons sur le fait que dans l'expression « action du langage sur le langage », le mot important est celui d'*action*. Pour le dire autrement, le slogan ici n'est pas « au commencement était le verbe », mais plutôt « im Anfang war die Tat », « l'action est première ».

particuliers « existant » dans l'univers ensembliste, à cette belle idée de l'entier comme acte pur[20], le contraste, on l'admettra, est saisissant.

Cohérence calculatoire

L'idée que l'action cotextuelle puisse être signifiante est encore bien sûr à ce niveau extrêmement vague. Pour faire saisir comment elle se décline avec succès dans le cas du lambda-calcul, il semble difficile de faire l'économie d'un survol au moins rapide de quelques unes de ses propriétés.

Tout d'abord il importe de voir que parmi les termes, les uns sont le point de départ de réductions infinies (« divergentes »), les autres au contraire ne peuvent jamais être indéfiniment réduits. Concernant ces derniers, leur évaluation finit toujours par produire un terme dans lequel il n'y a plus de règle de réduction à appliquer, plus rien à réduire, autrement dit finit par produire un *résultat*[21] (au demeurant alors unique[22]).

Cet espace des résultats constitue pour nous une sorte d'écran sur lequel les effets ultimes de l'action cotextuelle s'impriment au bout du compte (du moins pour tous les termes dont la réduction atteint cet espace) et où ils peuvent être saisis. Riche et varié (il y a une infinité de résultats distincts et parmi ceux-ci vivent en particulier ces entiers opérationnels que nous venons de présenter), cet espace des résultats se présente donc de prime abord comme un terrain favorable pour

[20] Les entiers de Church sont des lambda-termes *clos*, autrement dit leur « être dynamique » i.e. leur *rôle* d'itérateurs est intrinsèque, il ne dépend pas d'axiomes opérationnels (i.e. de règles de réécriture) supplémentaires qui viendraient leur conférer ce sens.

[21] Les résultats (termes dans lesquels il n'y a pas d'occasion d'appliquer une règle de réduction) sont appelés les « termes normaux ». Un terme normal, est dont un point terminal de la réduction (alias de la normalisation).

[22] La réduction du lambda-calcul est telle (propriété de confluence) que si la réduction d'un terme peut produire un résultat, celui-ci est unique (il ne dépend pas d'une stratégie particulière de réduction).

envisager les *différences d'effets*, « sur l'écran », de l'action cotextuelle, un terrain favorable donc pour aborder le sens par ses effets différentiels.

Tout le problème est alors de savoir si cette diversité, sans laquelle l'idée de sémantique dénotationnelle s'effondre, peut survivre aux identifications de termes engendrées, comme nous allons le voir à présent, par l'évaluation[23]. Pour faire saisir cette idée cruciale d'identification « forcée par le calcul », retournons un instant dans le cadre de pensée, spécieux mais familier, où la dénotation, antérieure au sens, vient le fonder, et observons qu'une réécriture qui ne préserverait pas la dénotation serait usuellement considérée comme *un calcul erroné* : dire qu'en effectuant le calcul correct de 2×2, je trouve la valeur 4, c'est dire que « depuis le début » ces deux expressions qui n'avaient pas le même sens, avaient même dénotation. Comme on l'a dit, l'orientation de la relation du sens à la dénotation est cependant ici renversée, et dès qu'on la rétablit dans son orientation naturelle, plutôt que de dire « j'appelle calcul, une transformation du sens qui préserve la dénotation », je dois dire « j'appelle dénotation, ce qui est préservé par la transformation du sens ».

L'évaluation de l'éventuel pouvoir dénotationnel de l'action cotextuelle commence donc par la considération de cette équivalence engendrée sur les termes par le calcul[24] (la plus petite relation d'équivalence sur les termes incluant l'évaluation[25]) et de la partition de l'ensemble des programmes qu'elle entraîne (le quotient).

[23] La question est : le calcul ne rend-il pas égaux trop de termes ? N'engendre-t-il pas, littéralement, la confusion ?

[24] Pour fixer les idées, on peut de nouveau reprendre nos exemples familiers : 2×2, $2 + 2$ et 4 ainsi sont tous « identifiés » par la dynamique, ils appartiennent à la même classe d'équivalence. Rappelons cependant que, quoique pédagogiquement utiles, ces exemples demeurent impropres : si ces termes étaient « rédigés » en lambda-calcul, ils ne comporteraient pas de symbole pour, notamment, la multiplication, mais l'expression complexe correspondant à *un programme particulier* pour la multiplication.

[25] En lambda-calcul, l'équivalence engendrée par le calcul, notée $\approx_\beta$, est appelée la β-équivalence.

À partir du moment où des termes distincts sont ainsi « identifiés » (qu'ils le soient « par le calcul » ou par quelque équivalence sur les termes qu'on serait amené à considérer pour une raison ou une autre, et qui identifierait plus de termes que le simple calcul ne le fait[26]), une condition minimale pour qu'une valeur sémantique puisse émerger de l'action sur le cotexte est que le calcul n'en vienne pas à identifier *trop* de termes, typiquement à identifier *tous* les termes, situation d'« incohérence calculatoire » qui annihilerait tout possibilité de discrimination dénotationnelle (en particulier, dans la mesure où, nous l'avons vu[27], les entiers eux-mêmes figurent en tant qu'itérateurs parmi les résultats, tous les entiers seraient identifiés par le calcul)[28].

Faculté de représentation

Dans le cas du lambda-calcul, non seulement cette condition minimale de cohérence calculatoire est satisfaite (il y a au moins deux termes que le calcul ne rend pas égaux), mais la « distinction » entre les entiers naturels (entre autres) est préservée (deux entiers distincts ne sont jamais égalisés par le calcul), en sorte que la *représentation* de fonctions des entiers dans les entiers devient envisageable.

[26] On pense à l'α-équivalence ($\approx_\alpha$), qui correspond à l'habituel renommage des variables liées (qu'on s'empresse de laisser tacite : les termes sont considérés à α-équivalence près) et à l'η-équivalence sur les termes ($\approx_\eta$), qui laisse l'extension invariante et s'avère requise pour que des isomorphismes calculatoires « naturels » (deux termes forment un isomophisme calculatoire, lorsque leur composition dans les deux directions converge vers l'identité) soient bien des isomorphismes (puisque, de fait, la convergence vers l'identité est réalisée dans ces cas seulement « à η-équivalence près ».)

[27] Cf. page 21.

[28] Dans le lambda-calcul (soit dit une dernière fois : tacitement quotienté par $\approx_\alpha$), on montre que l'équivalence $\approx_{\beta,\eta}$ sur Λ est maximale au sens où identifier deux termes normaux qui ne sont pas dans cette relation, mène au collapse dénotationnel des programmes. En définitive, compte-tenu de cette remarque, mais aussi de celle ci-dessus sur les isomorphismes calculatoires, le bon cadre pour questionner les propriétés dynamiques et dénotationnelles du lambda-calcul est donc le triplet : $(\Lambda, \rightsquigarrow_\beta, \approx_{\beta,\eta})$ ou encore, de façon moins redondante : $(\Lambda, \rightsquigarrow_\beta, \approx_\eta)$.

Chaque terme t du lambda-calcul tel que l'évaluation du terme formé en l'appliquant à n entiers termine *en produisant un entier* (ou diverge sinon) détermine le graphe ensembliste d'une fonction : il suffit pour cela d'oublier toutes les traces du calcul à l'exception du résultat lui-même (l'entier « calculé ») en tant qu'associé à une entrée particulière (tel n-uplet d'entiers). On dit dans ce cas que le terme t représente cette fonction[29]. Avec cette terminologie, le « pouvoir de représentation » du lambda-calcul évoqué plus haut (son caractère « universel » en tant que langage de programmation) s'exprime brièvement : toutes les fonctions récursives partielles y sont représentées par (au moins) un terme[30].

Pour terminer cette introduction express au lambda-calcul comme outil de sens et de dénotation, j'aimerais observer que la manière dont la faculté de représentation du lambda-calcul vient d'être exposée peut sembler contredire le principe de postériorité de la dénotation par rapport au sens. En effet, présentée comme ci-dessus, la sélection préalable d'un sous-ensemble de représentants clos normaux, les entiers de Church, pour « coder » les entiers naturels, semble reposer sur la donnée préalable des vrais entiers, *ex machina*, dont les entiers du lambda-calcul ne seraient que l'encodage. Sans m'étendre pour le moment davantage, j'aimerais souligner que des ensembles de termes comme l'ensemble des entiers de Church peuvent aussi bien être engendrés « de l'intérieur du calcul » à partir de ses propriétés (cf. plus généralement, la notion de « type de données » due à J.-L. Krivine) ; voir aussi la fin de la deuxième section du présent article et la section

[29] Terminologie standard mais peu heureuse puisque la dénotation ensembliste semble ici précéder le sens, alors qu'il faudrait plutôt dire que t est un programme qui *produit* ou *réalise* cette fonction.

[30] Les multiples termes représentant une même fonction, correspondent potentiellement à des algorithmes distincts. Quoique toutes les récursives partielles soient représentables moyennant ce mode particulier de représentation des entiers, certains algorithmes n'y sont réalisés par aucun terme (typiquement aucun terme ne réalise le « prédécesseur en une étape » sur les entiers de Church, mais on peut toutefois en trouver un, moyennant le choix d'un autre mode représentation des entiers).

du troisième chapitre de mon mémoire d'Habilitation à diriger des recherches consacrée à la complexité implicite, [23]). Il y a sans doute là une piste intéressante pour comprendre, et le cas échéant critiquer, des stratifications « ontologiques » à la Kronecker (ou d'autres moins radicales) qui semblent toujours très arbitraires ou dogmatiques.

Action et interaction

L'idée de mettre au départ de la sémantique, non pas la confrontation du langage avec une « ontologie » postulée, mais l'action homogène du langage sur le langage, est aussi, finalement, le fil qui de l'*action* mène à l'*interaction*. En effet, un sous-texte étant le cotexte de son cotexte, ce que nous avons décrit comme action d'un sous-texte sur son cotexte, devrait pouvoir en principe être décrit aussi bien comme action du cotexte sur ce sous-texte. Autrement dit l'*action* du terme sur son cotexte doit plutôt être conçue, au moins en principe, comme *interaction* entre le terme et son cotexte.

Il se trouve que cette idée, valable en principe, n'est pas directement réalisée dans le cadre du lambda-calcul. Tout d'abord le cotexte au sens littéral n'est généralement pas un terme *stricto sensu*, donc la notion de « plongement dans un cotexte », en tout état de cause, doit d'abord être précisée. Par ailleurs, comme le lambda-calcul est intrinsèquement dissymétrique (asymétrie « de type intuitionniste » entre entrées et sorties), réaliser dans ce cadre la dualité entre terme et cotexte signifie au mieux visibiliser, mais pas simultanément, les deux sens de cette dualité : selon un premier type de transformation, le contexte sera une fonction attendant le terme comme argument ; selon un second type de transformation[31], ces rôles seront inversés. Enfin, dans les systèmes de réécriture, l'individuation des processus en interaction est assez problématique (imaginons de composer deux termes rédigés avec des encres de couleur distinctes : la réécriture mêlera rapidement les deux

[31] En fait la trace sur les termes des techniques de ¬¬-traduction sur les types...

couleurs dans le code résultant[32], en sorte que la dualité primitive entre les deux agents de l'interaction, dans ce format, devient vite délicate à saisir). En résumé, dans le cadre de dispositifs plus abstraits de représentation de la dynamique (« machines abstraites », ludique) l'idée d'une évaluation comme *interaction* recouvre un sens plein. Et c'est elle qu'on retiendra dorénavant comme formant notre horizon[33].

Obstacles épistémologiques

Dans la mesure où, du coté de la philosophie, une grande partie des débats *épistémologiques* concernant les mathématiques (notamment à partir de la notion de « vérité » d'un énoncé proposée par Tarski) reposaient, au moment où les premiers travaux en sémantique dénotationnelle commencèrent à se diffuser de l'informatique vers la logique, sur le socle des propositions sémantiques de Tarski (ou ses variantes « "sémantiques" de mondes possibles »), une refondation épistémologique conséquente eût été naturelle.

Pourtant, si pour la communauté logicienne (en particulier, la communauté des théoriciens de la démonstration) l'importance théorique de la refondation sémantique (et de la découverte de la correspondance preuves-programmes présentée plus loin) s'est rapidement imposée avec une grande évidence (ne serait-ce qu'en raison de sa générativité théorique patente), force est de constater que les cercles institutionnels de l'histoire et de la philosophie de la logique et des mathématiques, dans l'ensemble, que ce soit en France ou à l'étranger, n'ont à ce jour toujours pas réagi : ni pris la mesure du caractère radical du changement de paradigme qui, il y a pourtant près d'une quarantaine d'années aujourd'hui, s'est progressivement imposé[34], ni pris conscience du prolongement de ce changement

[32] Image cinématographique en technicolor suggérée par Melliès (Paul-André).

[33] En particulier, pour connoter cet aspect particulier du point de vue *opérationnel*, je parlerai dorénavant parfois de point de vue *interactionnel*.

[34] Dans le contexte institutionnel philosophique, on peut certes trouver ici ou là de

dans les travaux les plus actuels, en théorie des fondements de l'informatique et des mathématiques.

On peut certes imputer en partie une telle « indifférence » à la perméabilité toujours plus réduite des frontières qui séparent, sur fond de division du travail intellectuel, les communautés scientifiques d'horizons divers (et tout particulièrement quand cette diversité oppose disciplines « techniques » et « philosophiques »), voire à la difficulté intrinsèque des champs de savoir pointus concernés (la logique mathématique et l'informatique fondamentale, discipline jeune à l'expansion rapide). En particulier, l'inexistence notamment en France d'une *philosophie de l'informatique* constituée, et la tendance avérée des philosophes à identifier l'informatique, fût-elle *théorique*, à une discipline aux enjeux plutôt techniques et appliqués, une science de l'ingénieur dénuée à leurs yeux de la noblesse théorique de la logique (cet ancien membre de la famille...) sont certainement des éléments déterminants à cet égard, dans la mesure où c'est justement du côté de l'informatique théorique que tout a commencé.

Toutefois, dans la mesure où l'école logique française s'avère avoir été, au moins à partir de la fin des années 1970, le principal foyer mondial de ces évolutions récentes (au moins pour le pan *Théorie de la démonstration*)[35] et où, dans la tradition française, les échanges entre

brèves évocations de la correspondance de Curry-Howard, mais une présentation systématique de la refondation sémantique et *a fortiori* de l'évaluation des enjeux épistémologiques de cette refondation semble inédite. Les travaux du groupe LIGC font bien sûr ici exception (cf. notamment [26], [36] et [22]). En Europe du nord, on doit également citer l'influence des travaux de Dag Prawitz en théorie de la démonstration (qui ont préparé la réception et la diffusion de la correspondance preuves-programmes dans la communauté des théoriciens de la démonstration) et de Per Martin-Löf.

[35] Notamment dans le cadre des équipes pionnières : l'*Équipe de logique mathématique* à l'université Paris 7, l'équipe *Logique de la programmation* à l'université Marseille 2, l'équipe *Preuves-Programmes-Systèmes* à l'université Paris 7 et diverses équipes de l'INRIA. Du côté proprement informatique, d'autres « foyers » ont bien sûr également joué très tôt un rôle de premier plan, notamment les écoles britannique (où émerge, semble-t-il, le projet de la sémantique dénotationnelle), italienne et

philosophie et logique mathématique sont assez intenses (au moins depuis Jean Cavaillès), une telle explication ne saurait suffire.

Comme J.-Y. Girard le souligne à de nombreuses reprises dans [15] et [16], quoiqu'avec d'autres mots, on peut voir ce silence et ce retard comme un effet de résistance de l'ancien paradigme, le produit d'une inertie active et signifiante : tout se passe en la matière, comme si, sur le terreau des théories logiques du siècle précédent, avait prospéré un système de préjugés finissant par former, en somme, une « idéologie scientifique » au sens où G. Canguilhem entendait cette expression (l'utilisation des résultats d'une science comme support ou renfort d'un système de préjugés) formant un obstacle épistémologique à la compréhension des changements intervenus, freinant la prise en compte philosophique des bouleversements advenus.

Sans trop s'étendre sur cette question forcément polémique, on peut en tout cas constater que le contresens profond sur la nature de la question sémantique en mathématiques qu'autorise une lecture « sémantique » de la notion d'interprétation du langage du calcul des prédicats[36] a eu pour effet collatéral d'hypertrophier la dimension descriptive[37], indexicale[38] (directement référentielle[39]) du langage mathématique. Si la référence directe a bien entendu sa place dans le cadre d'une approche pragmatique des actes de

suédoise d'informatique fondamentale...

[36] La « sémantique à la Tarski » est entièrement gagée, au plan mathématique, sur les *théorèmes de correction et de complétude* pour la logique du premier ordre et les succès de la *théorie des modèles* (devenue un outil incontournable en algèbre). Une fois dégagé le contresens (quant à la relation du sens à la dénotation) sur la base duquel ce dispositif est généralement qualifié de « sémantique », on doit en tirer une leçon : il n'a de « sémantique » que l'apparence et le nom.

[37] Idée tenace depuis Bolzano et Frege, toujours prête à la résurgence malgré le second Wittgenstein, d'une nature *descriptive* de la logique faisant face à « l'être-vrai » et à ses « lois » (Frege).

[38] Le signe en tant qu'il « indique, fait face à ».

[39] Caractère « indexical », (indicatif ?), « référence directe » : terminologie qui fait écho à la thèse défendue par S. Kripke selon laquelle les noms propres n'ont pas en somme d'autre sens que leur référence.

langage (où, l'inscription des locuteurs dans un environnement instaure l'indexicalité et impose une forme de réalisme pratique), dans le contexte d'une sémantique des énoncés mathématiques, elle n'est, au pied de la lettre, rien de plus que *la formalisation de la thèse de l'existence d'entités mathématiques indépendantes et stables* (dans l'univers des ensembles, réservoir inépuisable de dénotations), sans aucune portée *sémantique* particulière.

En conclusion sur ce point, il importe de souligner, pour lever tout malentendu, que remettre en question le *vérisme ensembliste* en logique ne signifie pas remettre en question la valeur de la théorie des ensembles (qui a du reste tout à fait sa place dans le cadre ouvert par la correspondance preuves-programmes[40]), ni celle de la notion standard de modèle[41], mais contester sa valeur *sémantique* et *ontologique*[42].

La logique au delà du langage

Si la refondation sémantique *venue de l'informatique* théorique (« sémantique dénotationelle » des programmes) est une source

[40] Cf. les résultats de Krivine sur les réalisateurs des axiomes de la théorie des ensembles.

[41] La disqualification du clivage traditionnel *syntaxe/sémantique* que produit le changement de point de vue en logique, ne résulte pas (ou pas nécessairement) d'une disqualification de la notion de modèle mais plutôt de la fusion de la construction des modèles et de l'analytisation des preuves – voir dans [19] et [18], les observations germinales et programmatiques de J.-Y. Girard sur le dépassement du clivage syntaxe / sémantique qui président à l'élaboration de la ludique [6], et sur lesquelles on revient plus loin ; voir aussi les analyses de G. Dowek dans [9] et de Patrick Dehornoy sur la « vérité » des axiomes de la théorie des ensembles et leur générativité méthodologique dans [8]) ; cf. enfin mes articles [26] et [27].

[42] Au delà des remarques parfois déjà anciennes sur l'absence de générativité ontologique de la sémantique ensembliste (du « décitationnisme » de W. Quine au déflationnisme ontologique du néo-pragmatisme), voir les analyses récentes de Gilles Dowek, *op. cit.*, sur les limites de la portée ontologique de la notion de modèle.

bien identifiée du futur paradigme interactionniste[43], on s'accorde à considérer que le déclencheur du basculement *de la logique* vers le nouveau paradigme est une observation faite en 1958 par Haskell Curry et complétée en 1965 par une seconde remarque due à William Tait. Cette double remarque (approximativement contemporaine de la refondation sémantique, mais essentiellement indépendante d'elle – même si l'histoire détaillée de cette période récente reste à faire) est passée à la postérité sous le nom de « correspondance de Curry-Howard » en raison de la synthèse de ces remarques[44] faite en 1969 par William Alvin Howard. Dit en peu de mots, il s'agit du constat de l'identité pure et simple de deux « dynamiques » jusqu'alors observées et étudiées dans des cadres théoriques et des communautés scientifiques relativement distinctes : l'informatique d'une part, la logique de l'autre.

Exécution des programmes et analytisation des preuves

La première de ces deux dynamiques est celle de l'évaluation calculatoire au sens informatique, autrement dit celle déjà évoquée plus haut de l'*exécution des programmes*. Pour être un peu plus précis, la remarque de Curry *et al.* concerne la dynamique de l'exécution des programmes informatiques rédigés dans ce langage de programmation épuré dû à Church que nous avons déjà présenté, le lambda-calcul (voir p. 19), et plus précisément dans un petit noyau de ce langage : le lambda-calcul *simplement typé*.

La seconde dynamique est celle de l'*analytisation des preuves*. Compte tenu de l'importance des usages du couple

[43] Ainsi c'est la décomposition de la *sémantique dénotationnelle* de l'implication intuitionniste en deux opérations plus primitives qui donne naissance à logique linéaire (présentée plus loin) qui figure au centre de ces évolutions [17].

[44] Sans doute faudrait-il ajouter à cette liste le nom de Nicolaas de Bruijn et d'autres encore (cf. les précisions données dans la section 1.4 de [11])

analytique / synthétique dans l'histoire de la philosophie, et pour éviter les malentendus, il importe de souligner que le niveau de langage auquel le concept d'« analytisation » s'applique n'est pas celui de l'énoncé en ses composants (niveau auquel on peut dès lors réserver l'usage plutôt du mot « analyse ») mais le niveau « suivant » d'*articulation du langage* qu'est le niveau du *texte*, ici le texte argumentatif : la preuve[45]. Une preuve analytique est une preuve dans laquelle tout énoncé présent est un sous-énoncé de la « surface extérieure de la preuve »[46] *i.e.* un sous-énoncé du théorème prouvé ou d'une hypothèse. Le terme « analytisation » est d'un usage plutôt minoritaire ; on utilise plus volontiers le terme technique moins connoté de « normalisation » des preuves, autrement dit la transformation des preuves en preuves « normalisées » dont l'analycité, par construction, est alors (sauf défaut marginal du système de représentation des preuves) nécessaire et flagrante.

Pour être un peu plus précis, côté « preuves », la remarque de Curry *et al.* concerne les preuves formelles en $\mathsf{DNI}_{\{\to\}}$, le fragment implicatif de la « Déduction naturelle intuitionniste » (un système qu'avait défini Gerhard Gentzen, quarante ans plus tôt, dans les débuts de la théorie de la démonstration) et le processus d'analytisation de ces preuves qu'avait défini le même Gentzen et dont il avait démontré la terminaison (donnant ce faisant une démonstration constructive de la cohérence de cette logique). En termes concis, cette remarque énonce que :

$$\left(\Lambda^{\mathsf{ST}}, \beta\text{-}\widetilde{\text{réduction}}\right) \simeq \left(\mathsf{DNI}_{\{\to\}}, \widetilde{\text{normalisation}}\right)$$

[45] En s'appuyant sur la terminologie des « niveaux d'articulation du langage » introduite par le linguiste André Martinet, on peut dire en simplifiant que le langage formel écrit du logicien articule deux niveaux : la preuve « articule » des énoncés, qui eux-mêmes « articulent » des symboles à valeur idéographiques (raison pour laquelle manque le troisième niveau des langues non idéographiques).

[46] Pour reprendre la jolie formule de Wittgenstein, p. 263 de [38]. En ces termes à peine métaphoriques, une preuve est analytique si on ne trouve rien à l'intérieur de la preuve, dans sa profondeur, qui ne soit déjà visible en surface.

Ce qu'on peut résumer en termes plus ordinaires et percutants par :

$$\left(\text{Programmes}\,,\ \overset{\text{exécution}}{\text{des programmes}}\right) \simeq \left(\text{Preuves}\,,\ \overset{\text{analytisation}}{\text{des preuves}}\right)$$

La correspondance de Curry-Howard a depuis lors été généralisée à des pans toujours plus larges de la logique et des mathématiques (à la logique d'ordre supérieur, à la logique classique, à un gros noyau des axiomes de la théorie des ensembles, ...), en sorte que les enjeux philosophiques dégagés ci-dessous ont une valeur tout à fait générale concernant la logique et les mathématiques (et non pas restreinte à tel ou tel fragment étroit de la logique). On évoque d'ailleurs désormais souvent cette version générale et ouverte de la correspondance sous le nom générique de « correspondance preuves-programmes ».

Logique	**"Informatique"** lambda-calcul	**Informatique** langages de programmation
Preuve	Terme typé	Programme
Formule	Type	Spécification
Conclusion	Type d'un terme	Spécification d'un programme
Théorème	Type des termes clos	Spécification complète d'un programme
Ensemble des hypothèses	Contexte	Environnement
Une hypothèse	Une variable libre typée	
Règle logique	Constructeur de terme	Instruction de programmation
Élimination des coupures	β-réduction	Exécution
Preuve normale	Terme normal	Résultat
Analytisabilité des preuves	Normalisabilité des termes	Convergence du calcul

TAB. 1 – Dictionnaire Trilingue preuves-programmes

Un nouveau regard sur les preuves

Pour le logicien, l'importance de la correspondance de Curry-Howard ne tient nullement à la difficulté qu'il y aurait eu à l'établir (techniquement parlant, il s'agit pour les spécialistes d'une remarque assez simple, localement assez mineure), mais au fait qu'elle a invité les théoriciens de la démonstration à porter systématiquement sur leur objet d'étude familier (les preuves formelles) un regard différent, extérieur, celui que l'informatique théorique avait développé sur les programmes et leur exécution.

Ce point de vue nouveau sur les preuves (élaboré au fil d'une autre tradition, dans une autre communauté scientifique poursuivant d'autres buts, usant de son propre langage et de son système de concepts vernaculaire – cf. le petit dictionnaire de la figure 0) provoquant un transfert de problématiques et d'imaginaire de cette branche de l'informatique théorique qu'est la *théorie du calcul* vers cette branche de la logique qu'est la *théorie de la démonstration*[47]. Laissant aux historiens des sciences le soin de tirer les leçons de ce « cas d'école » que constitue la correspondance de Curry-Howard pour la théorie des révolutions scientifiques, tentons de préciser de quelle manière elle modifie radicalement le statut épistémologique des preuves.

Nous avions vu plus haut comment, concernant les mathématiques, le texte informatique offre un régime de sens fondé sur *l'action sur le cotexte* (opérationalité cotextuelle) comme alternative au régime artificiellement fondé sur l'indexicalité (sur la référence directe), et comment le projet sémantique dénotationaliste visait à reconstruire la dénotation comme invariant du déroulement de cette action.

Établissant l'identité entre d'un côté un ensemble de programmes (et la dynamique de leur exécution) et de l'autre côté un ensemble de preuves (et la dynamique de leur analytisation), la correspondance

[47] Et réciproquement (même si dans le cadre de cet article, je m'intéresserai surtout à l'impact de ce transfert sur la logique, autrement dit, en retournant l'intitulé désormais canonique de « fondements logiques de l'informatique », à la « théorie des fondements informatiques de la logique »).

de Curry-Howard eut très tôt pour effet l'exportation de ce point de vue de l'opérationalité cotextuelle (« naturel » pour envisager les programmes) vers les preuves : une preuve après Curry-Howard n'est plus du tout conçue comme un enchaînement d'énoncés, elle n'est plus la « trace discursive d'un raisonnement », mais une structure, ouverte à l'interaction, susceptible d'agir sur le cotexte (à savoir les autres preuves avec lesquelles cette preuve est *composée*[48] au gré des textes mathématiques qui la réutilisent) et dont le sens se saisit d'abord comme scénario d'action sur ce cotexte. Le déroulé de cette action – la suite singulière d'événements que la preuve peut provoquer – dépend bien entendu des « règles » présentes et de la structure particulière de chacune d'entre elles. Sous ce jour, chaque (type de) règle est moins une règle d'*inférence* (règle de transition des énoncés vers les énoncés) qu'une règle d'*interaction*[49], règle déterminant une forme particulière d'interaction avec le cotexte.

Un nouveau regard sur la constructivité logique

Parmi les grands débats fondationnels qui mobilisèrent la première moitié du XX^e^ siècle et que la correspondance de Curry-Howard a profondément renouvelés, figure en bonne place le débat autour de la constructivité logique. Sur fond de naissance de la théorie de la

[48] Nous reviendrons plus loin sur cette notion de *composition*.

[49] Comme toujours dans le monde *recto-verso* des preuves-programmes, on peut aussi utiliser les mots de l'informatique pour décrire les phénomènes et donc dire que chaque règle est une *instruction de programmation* ; cette terminologie, littéralement tout à fait correcte et percutante (ce qu'atteste le grand nombre d'intuitions qu'elle a libérées) est toutefois en un sens régressive par rapport à l'idée de « règle d'interaction ». Cette dernière émerge plutôt « côté *recto* » (sans doute parce qu'en raison de l'observation déjà ancienne par G. Gentzen de symétries, dualités et autres régularités du côté logique – sur lesquelles nous reviendrons bientôt –, il est plus aisé d'extraire l'objet « preuve » du foisonnement des aspects contingents, afin de l'aborder *in abstracto*).

démonstration, la notion de « preuve constructive » repassa[50] sur le devant de la scène au début du siècle en raison, notamment, de la critique formulée par le mathématicien philosophe hollandais L. E. J. Brouwer quant à l'usage de certaines règles de démonstration hors du cadre finitiste (tiers-exclu, raisonnement par l'absurde, axiome du choix, etc) et plus généralement quant au statut accordé à la logique formelle et au langage dans le débat sur les fondements des mathématiques. La formalisation ultérieure de la « logique intuitionniste » par A. Heyting (comme sous-logique de la logique classique) permit de démêler deux aspects de l'intuitionnisme brouwerien : son anti-formalisme subjectiviste et son constructivisme. En neutralisant le premier, la formalisation rendit possible une analyse mathématique du second dans le cadre de la théorie de la démonstration, et au moment où la correspondance de Curry-Howard intervient, la jonction des théories de la constructivité logique avec l'informatique fondamentale (théorie de la récursivité) s'était non seulement déjà opérée, mais avait commencé au plan technique à être bien analysée[51].

Rappelons à grands traits la manière dont la problématique de la constructivité est envisagée au moment où la correspondance de Curry-Howard va intervenir. Les mots « logique constructive » étaient alors encore fréquemment employés comme synonyme de « logique intuitionniste » (i.e. la logique classique moins le raisonnement par

[50] Quoique la notion de preuve constructive soit inscrite au long cours dans l'histoire de la logique et de la philosophie des mathématiques, sa revivification au début du siècle dernier doit plus dans le détail à l'avénement de la théorie des algorithmes qu'à l'héritage kantien (l'activité mathématique comme construction de concepts), quelques nombreuses que soient par ailleurs les revendications de filiation de la part de divers protagonistes antagonistes de ces débats, Brouwer bien sûr, mais aussi Hilbert...

[51] On peut penser notamment au travail de Kreisel [28] (publié en 1958 et simplifié dans les années soixante par Friedman [12]) et à la fameuse interprétation dite « *Dialectica* » de Gödel, en termes de fonctionnelles récursives.

l'absurde ou ses équivalents)[52].

Pour l'essentiel ce qualificatif de « constructif » venait souligner une propriété spécifique des preuves analytiques intuitionnistes (propriété non satisfaite en logique classique) d'où découle les fameuses « Propriété de la disjonction » et « Propriété du témoin existentiel ». Cette dernière garantit, lorsqu'un énoncé existentiel de la forme $\exists x\, F[x]$ est prouvable sans hypothèses dans cette logique, que pour quelque terme t du langage (le « témoin » de cette existence, autrement dit le gage concret qui en « témoigne »), $F[\mathrm{t}]$ soit également prouvable (ce terme est facile à « extraire » de la preuve, à condition qu'elle soit analytique, puisqu'il suffit alors de le lire dans la prémisse de sa dernière règle ; cf. le terme $s^m(0)$, autrement dit l'entier m, dans la preuve de droite dans la figure .). Cette propriété, de même que la propriété de la disjonction (anticipation de la « complétude interne », cf. p. 50) devient généralement fausse dans le cadre d'une preuve (analytique intuitionniste) *avec hypothèses*. Mais elle reste satisfaite dans quelques situations, notablement[53] dans le cas où les hypothèses sont les axiomes de l'arithmétique du premier ordre (axiomes de Peano)[54].

Ces propriétés (et d'autres, analogues) convergeaient vers l'idée qu'une preuve en arithmétique intuitionniste *est* en un sens précis *un programme informatique* implémentant une fonction récursive des entiers dans les entiers, comme l'illustre la figure . qui représente la conversion d'une preuve intuitionniste non analytique d'un énoncé $\forall x \exists y\, B[x, y]$ de l'arithmétique (énoncé Π^0_2) instanciée sur une entrée donnée (l'entier de Peano n, autrement dit le n-ième successeur de 0) en une preuve analytique normalisée de cet énoncé, dont l'ultime

[52] D'une manière générale, le terme « constructivité » a souvent été et reste encore aujourd'hui utilisé à des niveaux de précision plutôt variables, l'adjectif *constructif* venant qualifier tantôt certaines règles de démonstration, tantôt les connecteurs (introduits par ces règles), tantôt les dérivations (composées de ces règles), tantôt les systèmes de démonstration (formés de ces règles), tantôt enfin les théories (clôture déductive d'axiomatiques relativement à la déductibilité déterminée par ces règles).

[53] Signalons aussi le cas où le contexte d'hypothèses forme une « théorie de Harrop ».

[54] Arithmétique intuitionniste alias « arithmétique de Heyting ».

règle est nécessairement une règle d'introduction du quantificateur existentiel, gagée sur un *témoin* de cette existence, ici l'entier $s^m(0)$.

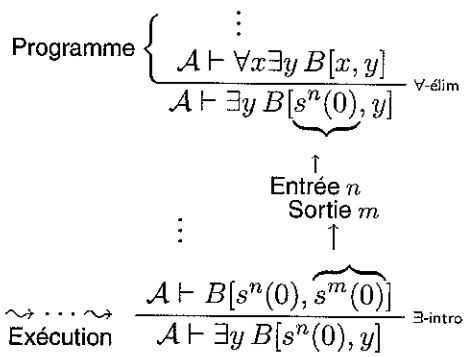


FIG. 1 – La preuve comme programme (antérieurement à Curry-Howard)

Quoique assez « moderne » dans sa façon d'approcher le lien entre *informatique* et *logique* puisqu'elle relie *analytisation des preuves* et *exécution des programmes*, cette approche reste en grande connivence avec l'ancien point de vue : les preuves sont certes l'*indice* de l'existence de l'algorithme, mais en définitive ce sont pour ainsi dire encore les énoncés qui parlent : les *données* (les entiers de Peano figurant dans les formules) sont ici en quelque sorte communiquées au programme *de l'extérieur*, puis, une fois l'exécution terminée, récupérées de même en sortie dans cette « surface extérieure de la preuve » que constituent les types. C'est donc ici aux types qu'échoit la fonction sémaphorique (ce sont les formules qui portent le signe voué à référence ; c'est au niveau des formules que celui-ci est récupéré, saisi).

Le contraste avec la vision « à la Curry-Howard » de la nature informatique des preuves est ici très net et met bien en évidence le clivage qu'a provoqué sur ce point précis la correspondance. Dans le cadre de cette dernière, au contraire, les données (entiers en entrée comme en sortie) sont réalisées directement dans la structure des preuves : elles *sont* ces preuves alias ces lambda-termes, et la fonction

sémaphorique leur revient toute entière (la « communication » entre l'intérieur de la preuve et sa « surface extérieure » étant par ailleurs assurée[55], en sorte que la conformité entre « ce que disent les preuves » intentionnellement – ce qu'elles font opérationnellement – et « ce que disent les formules dans les preuves extensionnellement » est assurée).

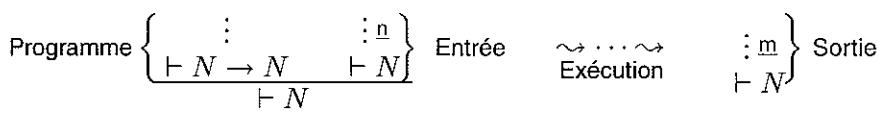


FIG. 2 – La preuve comme programme (après Curry-Howard)

La représentation des données, ou plutôt les données elles-mêmes, sont maintenant *internalisées* dans les preuves, incarnées par les preuves : celles-ci, littéralement, *sont* les « objets » (entiers, algorithmes) processuels.

Dans ce contexte, bien loin de se réduire aux traditionnelles « propriétés de la disjonction » et « du témoin existentiel », l'idée de « système logique constructif », a pris à partir des années 1980, le sens bien plus général de système logique « calculatoire », autrement dit apte à être utilisé et étudié comme un langage de programmation[56].

[55] Cf. les *théorèmes de représentation* pour les divers systèmes de « programmation par preuves » (typiquement, représentation de programmes équationnels par des preuves de totalité – cf. par exemple [29] pour l'arithmétique fonctionnelle du second ordre).

[56] Pour une discussion systématique du cahier des charges qu'un système logique doit satisfaire pour être ainsi qualifié de « calculatoire » (quelles propriétés de la dynamique des preuves, jugées pour elles-mêmes et pour leurs conséquences au niveau des algorithmes et des fonctions représentées), je renvoie au troisième chapitre de [23]. Quelques éléments de discussion sont toutefois déjà donnée à la fin de la présente section.

Sémantique des énoncés, sémantique des preuves

En exportant la *dimension opérationnelle du sens*, du monde des programmes vers celui des preuves, la correspondance transféra du même coup l'idée de « sémantique dénotationnelle » (née comme on l'a vu dans le contexte de l'entreprise de refondation sémantique venue de l'informatique) des programmes vers les preuves, inaugurant le développement d'une *sémantique des preuves* (par opposition à une *sémantique des énoncés*).

De la procédure (définie par G. Gentzen en 1935) permettant de transformer n'importe quelle preuve du fragment de la déduction naturelle considérée en une preuve progressivement plus analytique (jusqu'à parvenir à une version complètement analytique de cette preuve) ressort l'idée qu'un texte argumentatif donné, n'est pas tant « une preuve » (comme la terminologie le suggère) qu'« une preuve dans un certain état », plus ou moins analytique. En définitive, la dénotation de « la preuve » n'est donc pas tant le texte lui-même que j'ai sous les yeux, que ce qui au fur et à mesure de son analytisation, reste commun à toutes ses « versions ».

Observons que l'idée selon laquelle la question sémantique (sens et dénotation) doit être posée aux *preuves* avant de l'être aux *énoncés*, l'idée donc d'une *sémantique des preuves* (par opposition à une *sémantique des énoncés*) émerge au XX^e siècle, selon toute apparence à partir des années trente. On en trouve bien sûr des anticipations explicites dans le contexte de la philosophie de la logique de Ludwig Wittgenstein[57] et plus emblématiquement encore dans l'ainsi nommée « sémantique des preuves *à la Heyting* » (ou *à la Heyting-Kolmogorov*), qui consiste à promouvoir une description des preuves « comme fonctions », typiquement à voir une preuve de $A \rightarrow B$ comme un algorithme transformant toute preuve de A en une preuve de B (voir [21]). L'idée, en un sens, est également déjà au cœur de la philosophie intuitionniste

[57] Cf. par exemple : « Le résultat d'une preuve mathématique tire sa signification de sa preuve elle-même », p. 252 de [38].

des mathématiques de L. E. J. Brouwer, à ceci près bien sûr que ce dernier ne conçoit pas la question sémantique comme une question posée au langage (pas plus aux énoncés qu'aux preuves), mais à *la pensée* du mathématicien (le langage étant considéré par Brouwer comme un épiphénomène en mathématiques). Mais c'est surtout dans la philosophie du langage et des mathématiques de Michael Dummett que cette idée est investie d'un rôle central (à travers la thèse que le sens d'un énoncé est déterminé par les conditions d'assertabilité et d'usage de cet énoncé et donc, dans le cadre mathématique, des preuves de cet énoncé). Malgré la modernité relative des propositions de Michael Dummett (autour du statut des règles comme pourvoyeuses d'usages, donc opérationnellement *de sens* – dans la veine du second Wittgenstein, mais inspiré par les travaux précis de Dag Prawitz sur les règles de la déduction naturelle), ses thèses restent cependant largement tributaires d'une vision étroite de l'idée de constructivité logique (vision tendue héritée du rejet brouwerien du tiers-exclu et autres règles litigieuses, mais que, depuis lors, l'extension et la généralisation de la correspondance preuves-programmes à la logique classique et à la théorie des ensembles a fait éclater).

Si dans le contexte logico-informatique récent (postérieur à 1970), la tradition des travaux en « sémantique dénotationnelle des preuves » a prospéré et est en fait désormais fort longue (au point que, depuis plusieurs années déjà, dans cette communauté, le mot « sémantique », employé sans complément du nom ni qualificatif, y a précisément ce seul sens), la tradition philosophique a de son côté abordé ce problème de la « sémaphorie » (qui porte la signification : preuves ou formules ?) et en particulier débattu la question de savoir si, appliqué aux preuves, le mot « sémantique » restait approprié : défendre l'idée d'un sens des preuves, n'impliquerait-il pas de renoncer par ailleurs à celle d'un sens de l'énoncé ? En effet, *primo* : comment sinon conjuguer l'existence d'une multiplicité de démonstrations (donc a priori de significations) pour un même énoncé prouvé (*a priori* doué d'une signification unique) ? *Secundo* : comment concilier l'existence de

nombreux énoncés « faux » manifestement non équivalents du point de vue du sens, mais pour lesquels aucune démonstration ne peut exister ?

Dans la mesure où l'une et l'autre de ces objections ont trouvé dans les développements théoriques récents, dans le sillage de la correspondance preuves-programmes, une réponse tout à fait adéquate et d'ailleurs, avec le recul, très naturelle, il est inutile d'entrer ici dans le détail de ces débats. La première objection est neutralisée par la réponse « technique » apportée par les travaux de J.-L. Krivine (applications de la réalisabilité au problème de la spécification) analysant le sens d'un théorème donné comme comportement calculatoire commun, univoquement déterminé, de toutes les preuves de cet énoncé (ce qui supposait l'élaboration d'un cadre et d'un langage adéquats pour décrire ce « comportement », comme action sur l'environnement)[58].

Pour faire saisir comment la seconde objection est également contournée dans le prolongement de la correspondance de Curry-Howard, il importe de voir que non seulement les preuves (preuves correctes, si l'on veut, des théorèmes) mais aussi les *para-preuves*, les preuves incorrectes (« preuves » de non théorèmes) ont également un statut et un sens en tant qu'êtres dynamiques. Si c'est avec les propositions de J.-Y. Girard en Ludique [18] que cette idée sera systématisée, on peut d'ores et déjà en comprendre la substance à partir du lambda-calcul où elle affleure[59] et, au plan philosophique, en tirer un premier enseignement sur le statut des règles logiques.

[58] Que la description de ce comportement et le sens littéral (ensembliste) du théorème s'avèrent, dans tous les cas traités, apparemment incommensurables fait ici sans doute problème, mais ne me semble pas pouvoir être invoqué pour disqualifier la prétention à caractériser ce faisant la *sémantique* de l'énoncé prouvé.

[59] On rappelle que le lambda-calcul de Church est une théorie *naïve* des fonctions : *naïf* a ici aussi le sens qu'à ce mot lorsqu'il vient qualifier la théorie des ensembles de Cantor et son caractère contradictoire.

La logique ou les processus apprivoisés

Au pays *recto-verso* des preuves-programmes, on peut toujours choisir d'observer les preuves par l'autre bout de la correspondance : « du point de vue des programmes ». Rappelons tout d'abord que la correspondance de Curry-Howard établit l'identité des systèmes de démonstration concernés avec, toujours et seulement, des *fragments* du lambda-calcul[60].

Ces fragments sont obtenus en soumettant la grammaire de la construction des programmes (les termes du lambda-calcul), celle qui régule leur combinaison (notamment l'application d'un terme à quelque terme argument) à des contraintes supplémentaires, formulées dans une seconde grammaire, pour ainsi dire de second niveau : *la grammaire des types*[61], alias les formules (cf. la figure 2). Ces contraintes limitent a priori la construction des programmes et, de fait, réduisent leur nombre. Seuls « survivent » ceux qui satisfont ces contraintes : les programmes « typés » (alias les preuves, selon la correspondance de Curry-Howard).

Le fait important, ici, est que les contraintes de typage exercées sur les termes sont compatibles avec la dynamique ordinaire des termes[62] et donc avec le quotient induit par le calcul[63] (et en fait aussi avec celui où vivent les isomorphismes calculatoires, qui comprend aussi la « η-équivalence »[64]). Ce qu'on pourrait appeler « la bonne socialité

[60] On parle plus généralement de « correspondance preuves-programmes » pour qualifier sa généralisation à des extensions du lambda-calcul ou à d'autres formalismes.

[61] Le lambda-calcul typé réalise dans le cadre fonctionnel du lambda-calcul un principe de contrainte hiérarchisant, analogue à celui proposé dans sa *théorie des types* par Bertrand Russell pour rendre le cadre ensembliste moins « naïf ».

[62] Autrement dit la typabilité d'un terme (d'un type donné) est préservée au long de l'évaluation. Traduite dans le langage des preuves, l'idée de « conservation du type » signifie tout simplement que les avatars d'une preuve obtenus par analytisation sont toujours des preuves *du même théorème*.

[63] La β-équivalence.

[64] La notion d'isomorphisme calculatoire raffine la notion habituelle d'isomorphisme :

dynamique des processus » prévaut donc encore dans le fragment typé correspondant au système de déduction naturelle concerné, et le typage doit être vu non simplement comme découpant un sous-ensemble des termes (des programmes), mais au delà comme découpant *un fragment de la dynamique* (un sous-ensemble des évaluations, une sous-dynamique) équipant ce quotient qu'on avait décrit[65] comme le cadre adapté pour « questionner les propriétés dynamiques et dénotationnelles du lambda-calcul ». Notant Λ^{ST} l'ensemble des termes simplement typés, on peut le dire de façon compacte :

$$(\Lambda^{ST}, \rightsquigarrow_\beta, \approx_{\beta,\eta}) \text{ est un fragment calculatoire de } (\Lambda, \rightsquigarrow_\beta, \approx_{\beta,\eta})$$

Un premier point crucial est alors que la dynamique de ceux parmi les programmes qui sont « retenus par le typage » jouit de propriétés particulières, notamment la propriété d'universelle *terminaison* (composer entre eux ces programmes, ne donnera jamais lieu à une évaluation infinie) en sorte que le typage apparaît comme un dispositif de contrôle, de discipline d'une dynamique sinon vagabonde[66].

Pour autant, et c'est le second point crucial, cette discipline s'exerce sans préjudice pour le cœur du calcul nécessaire à la représentation[67] : les entiers (les itérateurs) notamment, restent présents, typables (y compris dans le système de typage sommaire qu'est le système des types simples), même si leur pouvoir d'itération (les fonctions susceptibles d'être itérées et donc la complexité de l'évaluation) reste lui-même « sous contrôle » (sous le contrôle des types).

cette dernière concerne des fonctions, la première les algorithmes. En termes de preuves, un isomorphisme calculatoire est une paire de preuves (alias programmes), l'une de $A \vdash B$, l'autre de $B \vdash A$, dont la composition dans les deux sens produit, par normalisation, l'*identité* dans le quotient considéré (i.e. produit une preuve « équivalente » à l'*axiome identité* correspondant).

[65] Voir page 24.

[66] Finitaire, la dynamique n'en demeure pas moins plus ou moins longue à terminer, plus ou moins fortement contrôlée, en fonction des systèmes de typage.

[67] Voir pages 21 et 24.

λ-calcul pur	**λ-calcul simplement typé**	**DNM$_{\{\to\}}$**
Variable $\frac{}{x \vdash x}$	Déclaration du type d'une variable $\frac{}{x : A \vdash x : A}$	Axiome identité $\frac{}{A \vdash A}$
Application $\frac{\Gamma \vdash t \qquad \Gamma' \vdash u}{\Gamma, \Gamma' \vdash (t)u}$	Application typée $\frac{\Gamma \vdash t : A \to B \qquad \Gamma' \vdash u : A}{\Gamma, \Gamma' \vdash (t)u : B}$	$\to$ -élim $\frac{\Gamma \vdash A \to B \qquad \Gamma' \vdash A}{\Gamma, \Gamma' \vdash B}$
λ-abstraction $\frac{\Gamma, x \vdash t}{\Gamma \vdash \lambda x\, t}$	λ-abstraction typée $\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x\, t : A \to B}$	$\to$ -intro $\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$
Jugements : $\Gamma \vdash t$ **Interpétation** si dérivable : t est un terme dont les variables libres sont parmi Γ	**Jugements :** $\Gamma \vdash t : A$ **Interpétation** si dérivable : t est un terme typable de type A quand ses variables libres sont du type déclaré dans Γ	**Jugements :** $\pi : \Gamma \vdash A$ **Interprétation** si dérivable : π est une preuve de A sous les hypothèses Γ

TAB. 2 – Les preuves comme programmes typés

Dans ce contexte, les règles logiques n'ont plus ce rôle de « police du discours correct » qu'une longue tradition tend à leur attribuer (quel que soit par ailleurs le type de fondement invoqué comme critère de cette « correction » : au pire dogmatique, au mieux normatif), mais celui d'une sorte de discipline des processus. Cette caractérisation de la logique comme théorie de l'« apprivoisement » de la dynamique[68] conduit à envisager avec un œil foncièrement différent le problème, central en philosophie de la logique, des critères de la correction logique.

Conditions nécessaires du logique

Le portrait dressé ci-dessus des règles logiques comme règles de typage venant *ex machina* limiter la construction des programmes et discipliner la dynamique (en imposant notamment sa terminaison) donne une idée générale du genre d'approche de cette question que la correspondance preuves-programmes permet de construire.

En effet, une méthodologie naturelle consiste alors à chercher à déterminer quelles propriétés de la dynamique des programmes typés (pour commencer, dans le cas favorable dévoilé par la correspondance de Curry-Howard poussée, disons, jusqu'à la déduction naturelle du second ordre, autrement dit le système F) résultent directement de cette « prise de contrôle » que réalise le typage sur la dynamique pure, et à « retourner » en quelque sorte tout ou partie de ces propriétés en une *définition du logique*[69].

[68] Cf. [27] et [26].

[69] C'est précisément cette approche du logique qui fonde le rejet, par la théorie de la démonstration contemporaine, des systèmes « logiques » auxquels un théorème d'élimination des coupures (autrement dit l'analytisabilité des preuves) fait défaut, sans même parler d'autres propriétés (l'analytisabilité formant le noyau vraiment minimal du cahier des charges). Cf. J.-Y. Girard attirant notre attention sur la nudité du roi, autrement dit sa dénonciation de la pléthore de publications en « logique formelle » qui, de ce point de vue, apparaissent comme des réponses vraies infiniment réitérables à des questions sans consistance ni enjeux, quoique parées des vêtements du formalisme mathématique, bien prompts à faire illusion [31].

Alors que les réponses traditionnellement apportées à la question des critères du logique[70] relèvent toujours et seulement de ce que J.-Y. Girard appelle le « premier sous-sol fondationnel »[71], l'approche suggérée par la correspondance preuves-programmes réoriente donc l'enquête vers le deuxième et le troisième sous-sols, avec comme point de départ les *propriétés de la dynamique*.

Observons d'emblée qu'au plan méthodologique, s'en tenir à prendre comme point de départ les propriétés qu'une discipline des types *particulière* (en l'occurrence, celle de la logique intuitionniste du second ordre) confèrent à un calcul pur *particulier* (en l'occurrence, le lambda-calcul), c'est prendre le risque d'aborder le phénomène

[70] Antérieurement à la révolution interactionnelle, la réponse la plus standard chez les logiciens à la question « Qu'ont les règles logiques de si spécial ? » était en somme : « elles préservent la vérité » (cette réponse reste aujourd'hui prédominante, me semble-t-il, dans la communauté des spécialistes de la *Théorie des modèles*). La « réponse » wittgensteinienne (celle du Wittgenstein du *retractatus*, selon le bon mot glissé par Vincent Danos) est plutôt un appel méthodologique assez général à changer les termes traditionnellement essentialistes de la question des critères, une invitation à descendre plus bas que le premier « sous-sol fondationnel » (voir note suivante), en abandonnant l'*être* au profit du *faire*. Parmi les propositions issues de la philosophie contemporaine, mêmes celles fondant les critères du logique sur l'interaction rationnelle (par exemple les propositions de Pascal Engel sur la logique classique comme norme protocolaire *définissant* la rationalité : protocole qui doit être imputé à l'agent épistémique porteur de croyances, dans l'interaction entre agents, parce qu'il le *constitue* en agent rationnel), relèvent seulement du *premier sous-sol* (lois logiques, vérité), voire à certains égards (ceci étant dit sans ironie, cf. mon article [26]) du *rez-de-chaussée* (conscience, intention).

[71] Voir, dans le présent volume, l'article de J.-Y. Girard. C'est dans [16], chapitre 7, que celui-ci introduit les trois niveaux fondationnels. Le *troisième sous-sol* est celui des algorithmes (même si nous sommes condamnés, de prime abord, à aborder ces derniers à travers des représentations potentiellement imparfaites et bavardes, à savoir les programmes dans un langage de programmation donné – alias les preuves dans un système de syntaxe logique donné), le *deuxième sous-sol* est celui des fonctions représentées, le *premier sous-sol* est celui des théorèmes, de la prouvabilité, de la vérité.

dynamique d'une façon trop empirique et donc potentiellement biaisée[72].

L'objectif naturel est donc plutôt de chercher à *élargir* le point de vue (par exemple en étendant la question à la logique classique et aux calculs purs sous-jacents, ou encore en étendant la logique par des règles compatibles avec une dynamique non déterministe), à *déplacer* le point de vue (par exemple en analysant la dynamique dans d'autres syntaxes logiques que la déduction naturelle, comme le calcul des séquents), mais surtout à l'*affiner* (comme le fera la logique linéaire qui, par une sorte de « zoom-avant » sur la dynamique, fera apparaître des distinctions et une organisation invisibles « à l'œil nu ») pour révéler les dimensions cachées de l'apprivoisement logique de la dynamique interactionnelle et poser le problème des critères dans un cadre enfin adéquat.

[72] En particulier, le fait que le lambda-calcul pur permette de représenter toutes les *fonctions* récursives partielles, n'implique pas que tout *algorithme* y ait sa place, en sorte que la quête d'un calcul processuel « pur » plus général fait sens *a priori*. Un autre exemple de biais (cette fois au sein du lambda-calcul) est celui du choix des représentants pour les données : certaines fonctions récursives représentables dans le lambda-calcul *sur les entiers de Church*, ne le sont pas via certains algorithmes (un exemple emblématique étant l'impossibilité démontrée d'obtenir le « prédécesseur en une étape », sur les entiers de Church). Si, parmi les « objets techniques » mobilisés vers ces objectifs, le lambda-calcul a longtemps été essentiel, c'est d'une part en raison du poids historique de la correspondance de Curry-Howard, mais aussi parce que ce fut longtemps le seul cadre disponible pour envisager les preuves comme sous-ensemble d'un « calcul pur », non apprivoisé par la logique. S'il reste aujourd'hui encore un point de référence incontournable, cette idée de calcul « sauvage », s'incarne désormais aussi dans des dispositifs plus larges ou plus fins comme les « réseaux purs » de Vincent Danos, les « réseaux d'interaction » d'Yves Lafont, le « calcul des séquents pur » de Hugo Herbelin ou la « ludique » de Girard. Qui plus est dans ce dernier cadre, comme on le verra à la fin de la présente section, les preuves ne sont pas la restriction typée d'un calcul pur ; ce sont les objets purs qui sont obtenus par *complétion* de l'espace des preuves, comme solution universelle à un problème d'orthogonalité.

On peut dire qu'à partir des années 80[73], cet objectif de reconstruire la logique *à partir de l'interaction calculatoire*[74] s'est constitué en un véritable *programme de recherche*[75] qui, au cours des deux dernières décennies, a mobilisé une grande partie de la communauté scientifique impliquée.

Avant de présenter dans la section suivante, la manière dont, au long des vingt dernières années, le « programme de recherche » sur l'approche logique de la dynamique calculatoire qui vient d'être esquissé s'est précisé, rendant étapes après étapes plus claires et saillantes les conditions *nécessaires* du logique, j'aimerais pour terminer cette section souligner que par rapport au questionnement sur les critères du logique, la quête de conditions nécessaires ne pouvait constituer plus qu'une *étape*.

[73] Sous l'impulsion déterminante de J.-Y. Girard, même si des recherches comparables ont été entreprises dans la foulée, dans d'autres parties du monde. Pour s'en tenir au prémices, c'est à J.-Y. Girard qu'on doit d'avoir mis, au centre de l'entreprise, comme point de départ, le *Calcul des séquents* et d'avoir élaboré l'un des outils majeurs sur lequel elle s'est appuyée, et s'appuie encore aujourd'hui : la *logique linéaire*. Parallèlement, en France également, mais aussi dans d'autres parties du monde, des entreprises liées mais inspirées par un projet différent, d'inspiration plus informatique, se sont développées (cf. le chantier ouvert en France par J.-L. Krivine, plus centré quant à lui autour du lambda-calcul et des méthodes de programmation par preuves dans l'arithmétique fonctionnelle du second ordre et ses extensions).

[74] D'où ce détournement de l'intitulé classique « Théorie des fondements logiques de l'informatique » que, dans [26], je proposais de retourner en l'intitulé complémentaire « Théorie des fondements informatiques de la logique », plus proche, me semble-t-il, de l'orientation réelle de cette direction de recherche. Avec le recul, « Théorie des fondements interactionnels de la logique » semble encore plus juste.

[75] Vers 1987, J.-Y. Girard a lancé sous le nom de « Géométrie de l'interaction » ce programme de recherche sur l'antériorité de la dynamique par rapport à la logique, avec comme intuition régulatrice que cette dynamique première relève de la *physis*. La dénomination « géométrie de l'interaction » est aussi utilisée pour désigner un sous-corpus de travaux techniques (dus à J.-Y. Girard, ou inspirés par ses travaux) formalisant ces idées dans un cadre d'analyse fonctionnelle.

L'internalisation

La vision pour ainsi dire « cybernétique » de la logique comme contrôle extérieur, *transcendant*, imposant des propriétés particulières à une interaction préexistante (*antérieure* à ce contrôle), qui fut méthodologiquement utile pour parvenir à dégager (à force d'avoir « élargi, déplacé et raffiné » le point de vue sur la dynamique, à force d'avoir repéré symétries, isomorphismes et quotients dans l'espace des preuves) un cadre favorable et général pour aborder *in abstracto* ces processus, s'est vu depuis lors remplacée ou au moins complétée par un point de vue qu'on pourrait par contraste qualifier d'*immanentiste*, selon lequel la norme régulatrice de la dynamique, *la logique*, émerge directement de l'interaction comme solution au problème de la *complétude interne*.

Le problème de la *correction* des règles logiques (sommes-nous sûrs qu'elles ne sont pas défectueuses, qu'elles ne nous mènent pas à l'erreur ?) et celui de leur *complétude* (avons-nous maintenant assez de règles ou bien nous en manque-t-il encore ?) est traditionnellement résolu en promouvant « en face » de l'univers des *preuves* d'un énoncé donné (les objets finitaires de la « syntaxe ») un univers de structures ensemblistes (les *modèles* de cet énoncé, objets généralement infinis de la « sémantique ») et en démontrant la dualité de ces deux univers : un énoncé prouvable grâce aux règles n'admet pas de réfutation, pas de contre-modèle (correction des règles) ; et si j'ai un énoncé non prouvable, ce n'est pas parce qu'il me manque une règle, mais bien parce qu'il est réfutable, parce qu'il en existe un contre-modèle (complétude). Mais alors, si nous voulons ces deux propriétés si naturelles et désirables, ou, moins poétiquement, si indispensables (la *correction* et la *complétude*), ne sommes-nous pas condamnés à reconnaître la nécessaire existence « en face » des preuves, « à l'extérieur » d'elles, de cet univers ensembliste où puiser l'interprétation de notre langage ?

L'analytisation des preuves offre justement une réponse négative

à cette question, du moins concernant la *correction* des règles (c'est précisément l'apport propre de G. Gentzen, à savoir une preuve, en quelque sorte *interne*, de leur *correction*). En effet, si nos règles sont telles que toute preuve est analytisable[76], il nous suffit alors de vérifier qu'il n'y a pas de preuve analytique d'une absurdité (or il suffit pour cela d'inspecter la « surface extérieure de la preuve », bornée, connue d'avance, puisque la preuve est analytique[77]).

Mais quid d'une *complétude interne* ? Tout semble ici conspirer, chaque pièce de la construction semblant soutenir l'autre : sans adouber une réalité extérieure aux preuves, sans pierre de touche à l'extérieur du langage et participant d'une autre nature, comment garantir qu'aucune règle n'a été omise ? Comment s'assurer de la complétude et quel sens peut même alors avoir ce mot : complet ? Mais par rapport à quoi ?

Dans ses travaux récents, J.-Y. Girard jette les bases d'une approche nouvelle de cette question (voir [19] et [18]). Au lieu d'adouber un monde *extérieur aux preuves* (l'au-delà statique des « lois de l'être vrai »[78]), il montre comment la dualité nécessaire n'a pas besoin de cet au-delà pour prospérer, qu'il suffit pour cela d'*élargir*, d'*enrichir* le monde des preuves de façon à y accueillir, en plus des preuves elles-mêmes, des para-preuves (des preuves fausses, des contre-preuves), contre-preuves dont les règles sont comme sélectionnées par la dynamique (elles ne portent pas préjudice à la socialité interactionnelle des règles et c'est comme solution à ce problème d'orthogonalité qu'elles émergent).

A l'ancien face à face des preuves et des modèles, à l'ancienne « dualité dualiste » (autrement dit une dualité entre hétérogènes), Girard substitue donc une « dualité moniste » (une expression paradoxale en apparence, mais non si l'on saisit son sens véritable de dualité *entre homogènes*, dualité *entre pairs*) accomplissant, selon

[76] I.e. peut être transformée en une preuve analytique du même énoncé.
[77] Cf. page 32.
[78] « Gesetze des Wahrseins », Frege.

ses propres termes, l'« abolition de la distinction entre *syntaxe* et *sémantique* »[79].

On peut reformuler cette nouvelle étape du programme de recherche girardien (en gardant à l'esprit l'idée développée dans la section précédente que les *règles de typage*[80] disciplinent la dynamique) en disant que l'ensemble des démonstrations est un fragment d'un espace plus large où vivent en bonne socialité interactive des processus que la logique vient « civiliser » et des processus, notamment infinitaires, qu'on voyait plus haut simplement comme des processus indociles, rebelles à ce contrôle, mais que, ayant « élargi notre horizon », nous verrons dorénavant comme fidèles à un contrôle, mais à un contrôle plus lâche, au contrôle d'une logique qu'on aurait tendance à qualifier d'erronée (où, par exemple, de l'atome propositionnel p, on infèrerait l'atome propositionnel q ou encore où, par exemple, p et ¬p seraient équivalents, comme dans les théories « naïves » évoquées plus haut...).

Selon cette approche contemporaine, les types apparaissent non plus comme limitant *ex machina* la construction des programmes et comme disciplinant leur dynamique, mais comme le produit d'une auto-discipline de l'interaction[81]. Dit d'une manière percutante : ce n'est plus le typage qui sélectionne une propriété de la dynamique, mais une propriété de la dynamique qui sélectionne le typage.

[79] Cette abolition est aussi une réfutation des positions essentialistes fondées sur le théorème de complétude. Voir la dernière partie de la première section du présent article.

[80] Dans le cadre de la correspondance preuves-programmes, on rappelle que les *règles de typage* ont également un autre nom : les *règles d'inférence logique*.

[81] En ludique, un *type* (un type interactif) est un ensemble d'agents (des néologismes comme *interagents* ou *interagissants* semblent un peu lourds, mais seraient plus adéquats, d'autant que hors du contexte *inter*-actif, le mot *agent* connote facilement l'*intention* et, à demi-mots, la *conscience*), identifiés à des stratégies, qui réagissent de la même manière à tous les tests, un ensemble de stratégies égal à son bi-dual, relativement au concept d'orthogonalité (c'est à dire l'idée que les stratégies qui se font face sont compatibles : le dialogue entre elles peut s'instaurer. Voir [16], principalement chapitres 13 et 14).

Le critère du logique devient alors enfin *intrinsèque* – au point d'ailleurs de faire place à l'idée, très déstabilisante pour un logicien né au XX^e siècle, d'une possible *réforme de la logique*, autrement dit des règles classiques elles-mêmes[82].

Conclusion

Parmi le petit nombre de concepts légués par la logique mathématique à la philosophie à l'aube du XX^e siècle, le couple *syntaxe-sémantique*[83] s'est longtemps vu attribuer un rôle régulateur de tout premier plan, non seulement pour les réflexions sur les fondements des mathématiques prenant pour point de départ les aspects structurels du discours mathématique, mais aussi comme cadre adéquat pour penser plus généralement la question du langage dans les sciences, et au delà, dans le contexte du positivisme logique, comme cadre d'analyse de la structure des théories scientifiques et outil de réflexion sur la méthodologie des sciences.

Ouvrir la logique au monde

Concernant les fondements des mathématiques, nous avons déjà mesuré dans la section précédente, la force de l'impact qu'a eu sur la question des critères du logique et leur fondement interactionnel, l'abolition de cette distinction *syntaxe / sémantique* (*preuves* et *modèles* d'un énoncé) au profit d'une dualité dynamique entre processus homogènes, de l'interaction desquels émergent les règles de la logique[84].

[82] Voir divers passages de [16], par exemple page 366.

[83] La dualité entre preuves d'un énoncé / modèles d'un énoncé, vue comme solution au problème de la correction et de la complétude de la logique ; voir la dernière partie de la deuxième section du présent article.

[84] Impact percutant au point de déstabiliser la logique classique elle-même : il revient à la complétude interne, loin des présupposés, de « choisir » les règles et notamment d'induire le niveau adéquat d'infini dynamique. Cf. la dernière partie de la deuxième

Au plan épistémologique, cette disparition de la distinction *syntaxe / sémantique*, véritable clef de voûte de toute la conception *représentationaliste* de la logique, prépondérante au siècle précédent (et bien souvent aujourd'hui encore), mais désormais sans tenue, oblige à repenser le rôle de la logique dans les sciences, à reconsidérer son statut d'outil de description structurelle ou de modélisation de protocoles méthodologiques, à renouer différemment les liens de la logique au monde[85].

Dès les premières évocations de la « Géométrie de l'interaction »[86] comme programme de recherche pour l'avenir, J.-Y. Girard revendiquait une intuition méthodologique tournée vers la physique[87]. L'ontologie processuelle[88] antérieure au typage logique et dont celui-ci

section du présent article. Voir aussi le chapitre 3 de [23].

[85] C'est le sens du beau titre, *Ouvrir la logique au monde*, qu'avait proposé Giuseppe Longo pour la journée organisée à l'Ecole Normale Supérieure de Paris, le 29 septembre 2006 par le *Groupe Histoire, Philosophie, Sciences* (anciennement *Collectif* HPS). Archives, notamment sonores, disponibles en ligne via `www.ens.fr/chps`.

[86] Le mot « géométrie » a ici une triple valeur : une valeur littérale tout d'abord, renvoyant à l'espace, emblématiquement illustrée par la syntaxe des « réseaux de preuves », dont la correction mobilise des critères non locaux et relève de l'homotopie des structures considérées (voir [17], [20] et l'abondante littérature sur cette question ; voir aussi le chapitre 3 de [23]) ; une valeur idiomatique plus directement technique, par ailleurs, qui renvoie aux approches « non commutatives » qui généralisent la théorie des ensembles ; enfin une valeur méthodologique générale, moins littérale - une invitation à la recherche des structures mathématiques profondes sous-jacentes à la dynamique du texte argumentatif visant à faire en sorte que la logique, déja séparée de longue date d'avec la psychologie, puisse aussi à présent cesser d'être vue comme étant peu ou prou une science du langage et, soit enfin instituée en science de la nature.

[87] Outre les publications qui relèvent de la géométrie de l'interaction au sens étroit, voir les intrusions récurrentes d'aspects quantiques dans le travail logique de J.-Y. Girard et notamment [13] et le chapitre 17 de [16]. Pour des considérations historiques sur la réflexion sur la naturalité en logique, voir [24]

[88] Expression commode, mais qui, à l'instar de « l'être du devenir » confine à l'oxymore parménido-héraclitéen. « Existence » (plutôt qu'« essence ») évite la confusion (et a été proposé au demeurant par J.-Y. Girard qui souligne, dans [14], la dimension « existentialiste » de l'interactionnisme logique).

émerge après coup, n'est en rien nébuleuse ou vague : les interactions élémentaires axiomatiquement définies qui engendrent les processus possibles sont parfaitement structurées et relèvent d'interprétations catégoriques. Par ailleurs, la normativité de la correction et de la complétude internes[89] est tributaire d'aspects concrets (espace, temps, non déterminisme, complexité, adressage, délocalisation...) qui rapportent ces interactions à la nature physique du traitement de l'information et aux modalités physiques de la synchronisation des processus[90].

En dehors de ce lien fondamental avec la physique, il reste à tirer les leçons philosophiques et épistémologiques de l'établissement, au cours de ces dernières années, de passerelles nouvelles entre la logique et, via le concept d'interaction calculatoire et communicationnelle, diverses disciplines scientifiques dans le champ desquelles des phénomènes interactionnels sont centraux, principalement la biologie (à partir au moins de 2001), la linguistique (notamment depuis 2005, cf. le projet Prélude déjà cité), voire l'économie (analyse de la structure de ce qu'on pourrait appeler l'interaction transactionnelle).

Concernant la biologie, il s'agit des investigations en « biologie moléculaire formelle » ou « bio-concurrence »[91], qui vise la reconstruction et la validation des mécanismes moléculaires concur-

[89] Voir les deux dernières parties de la deuxième section du présent article.

[90] Observant, tautologiquement, que la complexité des interactions dans la nature n'excèdent pas les limites du faisable, il serait juste de considérer que l'infini « tempéré » émergeant comme solution du problème de l'internalisation de la complétude n'est au juste que l'« infini enfin naturalisé ».

[91] Pour une bonne présentation synthétique de ce thème, on peut consulter les rapports quadriennaux de l'équipe *Preuves-Programmes-Systèmes* sur `www.pps.jussieu.fr` et les bibliographies incluses. Cette passerelle entre la biologie et l'informatique (et, par transitivité, avec la logique) constituait l'un des trois thèmes abordés lors du colloque *Le logique et le biologique* que j'ai organisé le 22 avril 2005 à la Sorbonne (les deux autres thèmes de la rencontre étant *Logique et cerveau* et *La métaphore logicielle en génétique*). La publication des actes est en cours aux *Publications de la Sorbonne* dans la collection *Logique, langage, sciences, philosophie* (archives : `www-philo.univ-paris1.fr/Joinet/LogiBio2005.htm`).

rents qui sous-tendent les activités de la cellule. L'horizon visé est, idéalement, un isomorphisme entre d'une part la dynamique du calcul dans les algèbres de processus[92], d'autre part la combinatoire de la biologie moléculaire. Comme j'eus l'occasion de le souligner dans l'exposé d'ouverture du colloque « Le logique et le biologique », un succès dans cette entreprise, se retourne d'emblée en une question : *la nature calcule-t-elle ?* ou de façon peut-être moins équivoque *y a-t-il du calcul dans la nature ?* – étant entendu que, dans le cadre du paradigme interactionniste, très éloigné de la vision traditionnelle du calcul[93] centrée autour de la notion de machine de Turing et de son programmateur, les approches du « calcul » développées, parce qu'elles envisagent ce dernier à travers une axiomatique des types d'interactions élementaires et locales, font l'économie de la machine centralisée, autrement dit : de la calculatrice, et donc, *a fortiori*, du calculateur...

Du côté de la linguistique, outre les applications désormais traditionnelles des outils de théorie de la démonstration de la logique linéaire non commutative ([1]) et de lambda-calcul typé en

[92] Langages élaborés par l'informatique théorique antérieurement à ces "applications" et tournés vers la « programmation concurrente », dans laquelle les aspects de communication et de synchronisation liés au caractère distribué du calcul sont prépondérants (penser au transfert éclaté et au calcul réparti d'une information sur internet). Dans ce contexte comme dans celui du lambda-calcul, un typage est l'expression de propriétés des processus. Nota : si les guillemets autour du mot "applications" sont ici de mise, c'est que cette liaison informatique-biologie est en réalité à double sens. Si la biologie peut bénéficier des approches informatiques en question pour modéliser des processus de communication intra-cellulaire complexes, l'informatique de son côté peut également tirer profit des interactions communicationnelles repérées par la biologie pour introduire de nouvelles primitives de programmation.

[93] Cf. les travaux du colloque d'histoire et philosophie de l'informatique organisé par mes soins en avril 2008 à la Sorbonne, intitulé « La thèse de Church : hier, aujourd'hui, demain », et qui portait notamment sur l'histoire des conceptions de la notion de calcul, avec une large place accordée aux approches contemporaines : calcul quantique, bio-calcul, calculabilité sur les nombres réels, concurrence... (voir note 9 page 16).

« linguistique computationnelle » (y compris les « continuations » utilisées pour échanger le rôle – fonction *versus* argument – du texte et du cotexte en vue de la résolution d'équations anaphoriques), le programme de recherche du réseau d'équipes *Prélude*[94] a précisément pour projet l'étude de l'interaction dialogique à travers des formalismes comme celui de la ludique.

A ma connaissance, l'utilisation effective en économie des théories logiques de l'interaction comme outils de modélisation de l'interaction transactionnelle n'existe même pas à l'état embryonnaire ; mais la fréquence du recours à la métaphore transactionnelle en logique linéaire[95] (logique centrée autour de la question des ressources et de leur consommation) et en ludique (la plupart des exposés d'introduction à la ludique commencent par des exemples élaborés de « scénarii transactionnels ») est le signe, me semble-t-il, d'une connivence qui mériterait d'être investie.

La notion d'interaction calculatoire et communicationnelle, centrale dans le paradigme logique émergent, dresse donc des ponts de la logique vers ces disciplines. Le type du lien de la logique aux sciences ici construit, diffère profondément de celui auquel nous avait accoutumé la longue tradition, renforcée au XX^e siècle par le positivisme logique[96], accordant à la logique le statut épistémologique d'un *organon*, outil linguistique méthodologique, descriptif ou heuristique. Cette dimension transdisciplinaire – voire unificatrice – de la notion d'interaction est l'un des thèmes pivots du collectif LIGC, qui d'une manière générale promeut une réflexion sur le statut du logique comme produit de l'interaction et le statut de la logique comme science de la nature.

[94] Cf. la note 17 page 21.

[95] Cf. l'« implication linéaire » expliquée aux enfants comme échange d'un bien acquis contre un bien cédé ; cf. aussi le « menu de Lafont » déjà cité.

[96] Voir cependant l'intéressante tentative d'Alexandre Miquel proposant une approche calculatoire de la théorie de la falsification de Popper [30] dans le cadre de la réalisabilité à la J.-L. Krivine.

Logique, langage et pensée

Le formalisme qui a prédominé en logique à partir du « tournant linguistique », dès la fin du XIX^e siècle, rêvait d'une possible objectivation de la rationalité mathématique dans le discours normé (« paradigme des preuves-comme-discours »). Il s'agissait en somme de démontrer la démarcation de la logique relativement à la psychologie, en expulsant de la première toute référence à la subjectivité au profit d'une dimension objective du sujet, à savoir les formes objectives de l'évaluation du discours que les techniques du calcul des prédicats naissant semblaient pouvoir incarner.

Les théorèmes de correction et de complétude[97] pour la logique des prédicats (pourtant limitée au cas restreint de la logique *du premier ordre*) qui constituaient (avec l'incomplétude) la base technique du « réalisme ensembliste » en philosophie des mathématiques, servirent rapidement de gage scientifique à la généralisation et la prolifération, hors du strict champ de la philosophie des mathématiques, d'approches représentationalistes, s'appuyant sur les techniques formelles de la logique, allant jusqu'à produire cette expression de « philosophie formelle » qui résonne tristement, comme une défaite.

Dans la vision représentationaliste de la logique, le sujet n'est pas à proprement parler absent du face-à-face du langage et de l'univers ensembliste. Il est d'abord présent comme *porteur* du lien référentiel du langage vers sa « réalisation ensembliste », même si son rôle de « sujet » est ici réduit à la portion congrue : simplement paramétrique, il est identifié à tel ou tel choix accompli (éventuellement convenu) d'une interprétation particulière du langage. Il est par ailleurs présent sous les traits du calculateur de valeurs de vérité. Le point de vue représentationaliste est donc finalement en connivence étroite avec une vision « cybernétique » du sujet cognitif et computationnel évaluant les énoncés en fonction des informations reçues du réel (peu importe

[97] Correction et complétude externes, cf. la fin de la deuxième section du présent article.

ici que la logique soit conçue comme structure transcendantale de l'évaluation « computationnelle » propre à l'appareil cognitif ou comme norme protocolaire de l'intersubjectivité).

L'un des traits les plus marquants de la dissolution par le « monisme »[98] interactionniste, de la dualité syntaxe-sémantique (cette dualité hétérogène entre preuves d'un énoncé et réalisations de cet énoncé) est sans doute la neutralisation du paradigme représentationaliste qu'elle induit.

Du point de vue interactionnel, le rapport de la logique au monde ne se pose en effet plus du tout dans les termes trinitaires d'un sujet logique (computationnel et cognitif) médiateur entre le langage et le monde, mais dans les termes binaires[99] de la relation qu'entretient la dynamique processuelle profonde à ses effets logiques « en surface ».

De cette vision de la logique très indifférente à la question de la conscience (*a priori* nécessaire en revanche à la « représentation »), peut-on conclure à la dissolution pure et simple de la subjectivité, à la disparition du sujet ? Ou bien faut-il voir au contraire dans la présence, au cœur du logique, de cette dynamique tantôt sensée, tantôt insensée, l'indice d'une sorte d'objectivation non pas des formes profondes de la démonstration, qu'elle soit discours ou structure, mais du raisonnement lui-même, le propre du sujet.

[98] Cf. la fin de la section précédente.

[99] Observons que pour autant l'aporie herméneutique (la confusion du langage et du monde) est ici évitée : il n'y a pas adhésion littérale du logique au réel.

Bibliographie

[1] Michele ABRUSCI et Paul RUET. "Non-commutative logic I : the multiplicative fragment". Dans : *Annals of Pure and Applied Logic* 101 (1) (1999). P. 29–64.

[2] Francis BAILLY et Giuseppe LONGO. "Causalités et symétries dans les sciences de la nature. Le continu et le discret mathématiques". Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 51–114. ISBN : 978-2-85944-584-3.

[3] Francis BAILLY et Giuseppe LONGO. *Mathématiques et sciences de la nature ; la singularité physique du vivant*. collection « Vision des sciences ». Paris : Éditions Hermann, 2006. ISBN : 978 27056 6630 9.

[4] Jocelyn BENOIST. *Entre acte et sens. La théorie phénoménologique de la signification*. collection « Problèmes et controverses ». Paris : Vrin, 2002.

[5] Martin CAMPBELL-KELLY. "Christopher Strachey, 1916-1975, A Biographical Note". Dans : *Annals of the History of Computing* 7.1 (jan. 1985). P. 19–42.

[6] Pierre-Louis CURIEN. *Introduction to linear logic and ludics (Part II)*. Manuscrit en ligne sur `www.pps.jussieu.fr/~curien`, 2004.

[7] Olivier DANVY et Carolyn L. TALCOTT, éds. *Special Issue in Memory of Christopher Strachey*. T. 13, numbers 1 and 2. Higher-Order and Symbolic Computation. Kluwer Academic Publishers, 2000.

[8] Patrick DEHORNOY. "Au-delà du forcing : la notion de vérité essentielle en théorie des ensembles". Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 147–167. ISBN : 978-2-85944-584-3.

[9] Gilles DOWEK. "La notion de modèle suppose-t-elle une conception réaliste de la vérité mathématique ?" Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 171–192. ISBN : 978-2-85944-584-3.

[10] Gilles DOWEK. *Les métamorphoses du calcul*. Paris : Le Pommier, 2007.

[11] Gilles DOWEK. *Théorie des types (notes de cours)*. Manuscrit, 1999.

[12] Harvey FRIEDMAN. "Classically and intuitionistically provably recursive functions". Dans : *Higher Set Theory*. Éd. par D. S. SCOTT et G. H. MULLER. Lecture Notes in Mathematics 699. Paris : Springer Verlag, 1978. P. 21–28.

[13] Jean-Yves GIRARD. "Between logic and quantic : a tract". Dans : *Linear logic in computer science*. Éd. par Thomas EHRHARD et al. London Mathematical Society Lecture Notes 316. London : Cambridge University Press, 2004. P. 346–381.

[14] Jean-Yves GIRARD. "La logique comme géométrie du cognitif". Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 15–29. ISBN : 978-2-85944-584-3.

[15] Jean-Yves GIRARD. *Le point aveugle*. T. 1. Visions des sciences. Paris : Hermann, 2006.

[16] Jean-Yves GIRARD. *Le point aveugle*. T. 2. Visions des sciences. Paris : Hermann, 2007.

[17] Jean-Yves GIRARD. "Linear logic". Dans : *Theoretical Computer Science* 50 (1) (1987). P. 1–102. ISSN : 0304-3975. DOI : `http://dx.doi.org/10.1016/0304-3975(87)90045-4`.

[18] Jean-Yves GIRARD. "Locus Solum". Dans : *Mathematical Structures in Computer Science* 3 (11) (2001). P. 301–506.

[19] Jean-Yves GIRARD. "On the meaning of logical rules (I) : syntax vs. semantics". Dans : *Computational Logic*. Éd. par U. BERGER et H. SCHWICHTENBERG. NATO series F 165. Springer Verlag, 1999. P. 215–272.

[20] Jean-Yves GIRARD. "Proof-nets : the parallel syntax for proof theory". Dans : *Logic and Algebra*. Lecture Notes in Pure and Applied Mathematics 180. New York : Marcel Dekker publisher, 1996.

[21] Jean-Yves GIRARD, Yves LAFONT et Paul TAYLOR. *Proofs and types*. Paris : Cambridge University Press, 1989.

[22] Jean-Baptiste JOINET, éd. *Logique, dynamique et cognition*. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. ISBN : 978-2-85944-584-3.

[23] Jean-Baptiste JOINET. *Logique et interaction*. Spécialité : philosophie de la logique et de l'informatique. Soutenue le 12 décembre 2007. Paris. 2007.

[24] Jean-Baptiste JOINET. "Nature et logique de G. Gentzen à J.-Y. Girard". Dans : *De scientia* 1 (à paraître). Revue semestrielle de « Pensée des sciences », groupe de recherche franco-italien en histoire et philosophie des sciences.

[25] Jean-Baptiste JOINET. "Programme de l'interactionisme logique (préface)". Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 5–8. ISBN : 978-2-85944-584-3.

[26] Jean-Baptiste JOINET. "Proofs, reasoning and the metamorphosis of logic". Dans : *Advances in Natural deduction*. Éd. par L.-C. PEREIRA, E. HERMANN-HAUESLER et V. DE PAIVA. D'après les travaux de la conférence "Natural Deduction" organisée en 2001 par L.-C. Pereira, à la Pontifícia Universidade Católica, Rio de Janeiro, Brésil. à paraître aux éditions Springer.

[27] Jean-Baptiste JOINET. "Sur le temps logique". Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 31–49. ISBN : 978-2-85944-584-3.

[28] Georg KREISEL. "Mathematical Significance of Consistency Proofs". Dans : *Journal of Symbolic Logic* 23 (2) (1958). P. 155–182.

[29] Jean-Louis KRIVINE. *Lambda-calcul, types et modèles*. coll. Etudes et Recherches en Informatique. Paris : Dunod, 1997.

[30] Alexandre MIQUEL. *L'effectivité expérimentale de la preuve mathématique*. Manuscrit communiqué par l'auteur au printemps 2007, 14 pages, 2007.

[31] Yves-Jean RINGARD. *Mustard watches : an integrated approach to time and food*. Rap. tech. Prépublications de l'équipe de logique (*hors série*). Paris : Université Paris 7, 1990. P. 1–4.

[32] Dana SCOTT. "An appreciation of Christopher Strachey and his work (Foreword)". Dans : *[34]*. P. xv–xxx.

[33] Dana SCOTT. "Outline of a mathematical theory of computation". Dans : *Proceedings of the 4th Annual Princeton Conference on Information Sciences and Systems (1969)*. Princeton University, 1970. P. 169–176.

[34] Joseph E. STOY. *Denotational Semantics : the Scott-Strachey Approach to Programming Language Theory*. MIT Press Series in Computer Science. Cambridge (Massachusetts, USA) : MIT Press, 1977. ISBN : 0-262-69076-4.

[35] Christopher STRACHEY. "Toward a Formal Semantics". Dans : *Formal Language Description - Languages for Computer Programming*. Éd. par T.B. STEEL. Amsterdam : North-Holland, 1966. P. 198–220. ISBN : 0-262-69076-4.

[36] Samuel TRONÇON. "Interaction et signification". Dans : *Logique, dynamique et cognition*. Éd. par Jean-Baptiste JOINET. collection « Logique, langage, sciences, philosophie ». Paris : Publications de la Sorbonne, 2007. P. 119–145. ISBN : 978-2-85944-584-3.

[37] C. P. WADSWORTH. "The semantics and pragmatics of the λ-calculus (Ph. D)". Thèse de doct. Royaume-Uni : Oxford University, 1971.

[38] Ludwig WITTGENSTEIN. *Philosophie pour mathématiciens : les cours de Cambridge 1932-1933* (traduc. E. Rigal). Trans Europ Repress (TER), 1992.

Visions des sciences

Ouvrir la logique au monde

Philosophie et Mathématique de l'interaction

sous la direction de **Jean-Baptiste Joinet et Samuel Tronçon**

TRAVAUX DU GROUPE ***LIGC***

Hermann
CERISY

Ouvrir la logique au monde

Philosophie et Mathématique de l'interaction

TRAVAUX DU GROUPE ***LIGC***



Cet ouvrage rassemble une partie des exposés donnés lors de l'Ecole thématique du CNRS « Logique, Sciences et Philosophie », organisée par Jean-Baptiste Joinet, qui s'est tenue au Centre Culturel International de Cerisy-la-Salle, en 2006, à l'initiative du groupe « **L**ogique et **I**nteraction : **G**éométrie de la **C**ognition » (**LIGC**).

La première partie, intitulée « Physique et intuition », réunit des contributions concernant l'intelligibilité de la nature. C'est d'abord la logique qui est convoquée, par l'interrogation sur la dualité entre subjectif et intersubjectif (J.-Y. Girard) à travers les apports récents de la ludique. Sont analysés ensuite les liens fondamentaux entre le formalisme mathématique et l'intuition en physique (T. Paul, M. Bitbol). Pour finir, la mathématique formelle est entrevue depuis la phénoménologie (V. Gérard), notamment dans le dialogue entre théorie des ensembles et ontologie.

La deuxième partie, « Nature et calcul : de la logique au sujet », introduit de manière critique la conception de la nature-calcul, en abordant d'un côté les limitations de la thèse de séparabilité entre systèmes calculables et systèmes chaotiques (G. Dowek), et de l'autre la différence entre déterminisme et prédictibilité en tant qu'elle questionne la thèse de Church physique (G. Longo, T. Paul).

La troisième partie, consacrée aux « Processus et primitives de cognition », aborde les questions de signification et d'ontologie. Les deux premiers chapitres s'articulent autour de l'hypothèse localiste, en s' attachant à l'ancrage cognitif des concepts mathématiques (B. Teissier), puis à la signification spatiale de la négation et de l'implication logiques (P. Livet). Les deux derniers chapitres abordent sur un plan philosophique les jalons théoriques qui permettent d'envisager une sémiotique cognitive (S. Tronçon), et les apports de l'intuitionnisme philosophique à une sémantique des constructions et des catégories (A. Benmakhlouf).

En clôture, F. Nicolas revisite la logique musicale à l'aune du « tournant géométrique » survenu dans le champ de la logique mathématique et dont l'ensemble de ces travaux tente de définir les contours.

www.editions-hermann.fr

ISBN 978 2 7056 6830 3

9 782705 668303

Hermann

CERISY

34 euros