# Conversion en logique et dans quelques domaines appliqués

## *mots, gènes et programmes*

**Alain LECOMTE**
**Groupe « Philosophie, Langage et Cognition » - Grenoble**

**Résumé**
*La « conversion » est un concept familier en mathématiques, dont il semble a priori qu'il n'y ait pas beaucoup à dire, or ce concept mathématique intervient de plus en plus dans des domaines qui lui étaient autrefois étrangers : en linguistique et en biologie. Il apparaît donc nécessaire de l'interroger. En chemin, nous le découvrons au cœur de mécanismes fondamentaux à l'œuvre dans les langages modernes de programmation sous la dénomination de « λ-conversion ». Ce dernier type de conversion possède son pendant logique dans les processus de transformations de preuves qui font passer de démonstrations abstraites vers des démonstrations concrètes. Or, n'est-ce pas justement au moyen de cette « élimination de l'abstrait » que fonctionnent les processus producteurs de certains objets finis et concrets de notre monde : qu'il s'agisse de phrases, assignables à des arbres d'analyse, ou qu'il s'agisse de séquences de protéines générées par un code ADN ?*

## 1 La définition de la conversion d'un point de vue mathématique

Le mot « conversion », qu'évoque-t-il a priori pour le mathématicien ou le logicien ? A première vue, des trivialités. Ainsi, dans les mathématiques les plus élémentaires avons-nous tous à un moment ou à un autre de notre scolarité opéré des *conversions* d'une base de numération dans une autre. Un nombre étant donné en base b il s'agissait de le convertir en base b'. Evidemment, il y avait un algorithme pour cela, reposant sur celui de la division euclidienne. D'où notre première observation : la conversion est ce qui fait passer de la représentation d'une chose (ici un nombre) dans un système donné à sa représentation dans un autre système, et notre deuxième observation : ce passage s'effectue au moyen d'un algorithme.

Pourquoi effectue-t-on de telles « conversions » ? Premièrement pour nous assurer qu'une chose ne dépend pas du mode de représentation que l'on a choisi : souvenons-nous des travaux des pédagogues des mathématiques des années soixante, Zarpad Dienes et Jean Piaget qui insistaient sur la nécessité de montrer aux élèves la construction du nombre en tant qu'invariant (par rapport aux systèmes de représentation), et deuxièmement, et de manière plus utilitaire, parce qu'il y a des systèmes plus commodes que d'autres pour faire certaines opérations. Par exemple si on considère la manière dont s'effectuent les calculs dans un ordinateur, on sait que cela requiert la conversion en base binaire, si on manipule des octets on aura intérêt à les représenter par des hexadécimaux et ainsi de suite. On connaît aussi les racines présumées anthropologiques des bases de numération les plus courantes (10, 12, 20, 60…).

Afin d'être plus précis cependant, nous dirons même que certains calculs peuvent être *vraiment* effectués au prix d'une conversion, en ce cas, la conversion n'est plus une commodité mais devient une vraie nécessité, découlant des moyens théoriques et/ou opérationnels dont nous disposons. Les Egyptiens, d'après le papyrus de Rhindt, exigeaient une représentation en base deux pour faire opérationnellement leurs calculs. Même si vous pouvez évidemment entrer vos nombres sous une forme décimale dans votre ordinateur, celui-ci va *devoir* les convertir en base deux pour faire *opérationnellement* les calculs que vous lui

demandez. Notre troisième observation est donc qu'il y a parfois dans la conversion un passage obligé pour effectuer des opérations concrètes bien définies.
Cette dernière observation nous permet d'entrer dans le vif du sujet, la différence entre calculs « abstraits » et calculs « concrets ». Bien sûr, « 1 + 2 + 3 = 6 », mais que dit cette écriture ? Elle dit simplement que si vous prenez les entités qu'il est convenu de noter au moyen des signes respectifs « 1 », « 2 » et « 3 » et si vous voulez faire entre elles l'opération appelée « somme », alors, vous pouvez être sûrs que cela va vous donner l'entité qu'il est convenu de noter au moyen du signe « 6 ». Est-ce un « calcul » à proprement parler ? Pas vraiment. En tout cas, il ne s'agit pas d'un calcul « concret » dans la mesure où en supposant l'existence d'un agent (humain ou machine) qui ne connaîtrait pas « les règles » de l'addition sous forme des tables que nous avons apprises, il ne lui serait pas possible d'effectuer ce calcul à partir de la seule donnée des représentations de l'opération et de ses arguments. C'est, pour nous, un calcul « abstrait » parce que, lorsque nous l'effectuons, nous faisons appel à un savoir abstrait, antérieurement acquis, nous ne retournons pas aux justifications initiales qui nous ont convaincu de la justesse de ce savoir et qui pourraient nous permettre de retrouver le sens de l'addition au cas où nous aurions oublié nos connaissances des tables.

## *2 Calculs abstraits, calculs concrets*

Qu'est-ce qu'un calcul « concret » ? Nous ne pourrons répondre à cette question que lorsque nous aurons donné aux nombres et aux opérations portant sur eux une représentation matérielle permettant une manipulation rigoureuse. D'autre part, évidemment, les définitions s'arrêtent toujours à un moment donné sur des termes primitifs, autrement dit des atomes, des signes qui ne seront pas définis. Cela veut dire qu'on peut se placer à différents niveaux d'explicitation des calculs. En supposant connues les entités notées « 1 », « 2 », « 3 », on pourra dire : le calcul concret, c'est celui qui permet de calculer la somme de trois nombres à partir de la définition de celle de deux nombres et de l'admission de l'axiome d'associativité : on se contentera de :

$$1 + 2 + 3 = (1 + 2) + 3$$

on saura faire 1 + 2, et on saura faire 3 + 3 donc on saura faire 1 + 2 + 3. Mais on peut évidemment souhaiter aller à un niveau plus bas.
Pour cela, admettons que nous ayons ce que les logiciens appellent *un système formel*, quelque chose de très simple, du genre de ce qu'on peut trouver dans le fameux livre de Hofstatdter, « Bach, Gödel, Escher » (Hofstadter, 1979, trad. franç. 1985), qui comporte la donnée *d'un alphabet de signes* (de simples marques écrites sur une page pour lesquelles on suppose seulement une relation d'*équiformité* : on sait reconnaître l'identité de deux signes au fait qu'ils ont « même forme »), de *règles de formation* d'expressions dites « bien formées » (*ebf*), d'un *axiome*, sélectionné parmi ces ebfs, et de *règles de déduction*, qui se limitent à transformer une ou plusieurs ebfs données en une ebf résultante (qui relèvent donc d'un pur travail de réécriture) :

alphabet : {|, +, =}

expressions bien formées : tous les mots sur l'alphabet contenant un et un seul + et un et un seul =, avec le + avant le =. (ex: ||||+|||=|||, |+=, |+||||=, ...)

axiome : +=

règles de dérivation :

R1 : pour toute ebf $\phi$, de $\phi$ on peut déduire: $|\phi|$

R2 : pour toute ebf $\phi = \alpha + \alpha' = \alpha''$, de $\phi$ on peut déduire $\alpha + \alpha'| = \alpha''|$

Dans un tel système, l'addition « 1 + 2 » se fait en dérivant, à partir de l'axiome une chaîne de préfixe « |+||= ». Notons que cette dérivation est *en même temps* un *calcul* et une **preuve**. C'est même une preuve *effective*, en ce qu'elle est effectuée dans un système formel, et donc entièrement formalisée : toutes ses étapes sont explicitées et se traduisent par l'application d'une des deux règles données dans le système. Une machine dotée de ces règles saurait faire toutes les additions d'entiers qu'on lui présenterait. Simplement, les résultats obtenus, du genre : « ||||||||||||||||||||+|||||||||||||||=|||||||||||||||||||||||||||||||||| » nécessiteraient pour nous une « conversion » dans un autre système symbolique nous permettant d'appréhender les résultats sous une forme plus compacte et moins difficile à lire.
Il est nécessaire d'ajouter ici qu'un tel système fonctionne bien parce que les représentations manipulées sont *matérielles*, ce qui autorise la traduction d'une opération en une simple manipulation de symboles. C'est la grande découverte de Turing d'avoir compris que toute opération « calculable » se ramenait à un dispositif élémentaire consistant en un ruban illimité dans les deux directions et une tête de lecture-écriture pouvant se déplacer dans les deux directions par rapport au ruban. La machine de Turing « reconnaît » un symbole à sa forme matérielle, elle concrétise l'idée selon laquelle il n'y a pas de mathématiques, pas de calcul, sans poser en principe premier *l'identité du signe à lui-même* : ce qui se note « a » demeure identique dans toutes les instances de répétition du symbole (modulo d'éventuelles variations accidentelles de forme, voir plus haut la relation d'équiformité).
Grâce au petit système formel précédent, nous avons pu voir qu'un calcul ne se différenciait pas fondamentalement d'une preuve. *La dérivation de la somme est en même temps preuve qu'elle est correcte*, cela se vérifie en contrôlant la correction de chaque pas. Si nous laissons le soin à une machine de faire un tel enchaînement de pas, il n'y aura pas de raison pour « qu'elle se trompe » au cours d'une application de règle. Autrement dit, le mécanisme du système assure à la fois l'exécutabilité et la rigueur. Nous saisissons au passage l'étrange issue où nous mène la poursuite de la rigueur mathématique : pour être assuré de celle-ci, le meilleur moyen serait de se décharger sur une machine (même une machine idéale comme l'est la machine de Turing), de sorte que les calculs et les preuves vraiment concrets ne dépendent plus de nous, soient exécutés dans un extérieur « machinique »[1].

Un calcul « concret » est donc une preuve effective. Cette notion suppose bien entendu qu'il y en ait de *non effectives*, autrement dit de plus abstraites. Cela se réfère en effet à la possibilité qu'on a de se situer à des niveaux plus élevés de la représentation symbolique (ici la notion de niveau s'entend dans le même sens qu'en informatique lorsqu'on parle de langage de plus ou moins haut niveau). On peut « calculer » aussi 1 + 2 + 3 en faisant appel à la propriété suivante[2] :

$$(1) \quad \forall n \sum_{0}^{n} i = \frac{n(n+1)}{2}$$

[1] Giuseppe Longo et Francis Bailly voient une « attitude schizophrène » dans ce mouvement qui, disent-ils « laisse cette fonctionnalité de l'homme, son intelligence, en dehors de lui-même, de son propre corps, de son cerveau, de son vécu, de son expérience active du monde », (Bailly & Longo, 2004)

[2] cet exemple et la discussion qui s'en suit viennent d'un exposé de Jean-Baptiste Joinet, lors de la journée « Le logique et le biologique » du 22 avril 2005, à la Sorbonne.

laquelle propriété est démontrable au moyen du principe de récurrence. Quel rapport avec la preuve précédente de 1 + 2 = 3, par exemple ? Eh bien, celui-ci que, en un sens, la preuve effective de 1 + 2 = 3 que nous avons faite à l'instant (elle ou son équivalent dans un autre système) *doit bien se trouver quelque part au sein de la grande preuve du résultat* (1) ! Autrement dit, nous avons, en démontrant (1), utilisé un mécanisme de preuve puissant, encapsulé dans notre axiomatique des nombres entiers, mais si nous « démontions » ce mécanisme, nous pourrions trouver en lui *une infinité* de petites preuves comme celle de 1+2 = 3 ! C'est d'ailleurs pour cela que le principe de récurrence a été « inventé » : pour nous faire faire l'économie d'un nombre infini de preuves ! Si les mathématiques sont correctes (et elles le sont !), on devrait pouvoir retrouver ces preuves infinitésimales au sein de la « méta » preuve. Nous allons donc dans ce qui suit, dans un premier temps, développer la notion de preuve ainsi que la distinction qui s'opère entre plusieurs niveaux de preuves, certaines étant plus « basiques » ou plus « effectives » que d'autres qui les contiennent de manière implicite. La conversion, au sens logique, c'est le processus par lequel on retrouve les preuves effectives à partir des preuves plus abstraites. Un passage de la description aux processus concrets en quelque sorte.

## *3 La convertibilité des preuves*

### 3.1 Le calcul des séquents

Puisque nous parlons de preuve, alors parlons **de logique**, en admettant, du moins en première approximation, que la logique est bien « la science des preuves ». Où, quand, comment se trouve exprimé en logique le rapport entre preuve non effective et preuve effective ? La réponse est simple à donner : elle nous ramène aux années trente et aux recherches de Gentzen sur les systèmes formels, notamment dans le but d'établir certains résultats comme la complétude de la logique des prédicats du premier ordre. Gentzen invente à ce moment-là une présentation de la logique sous forme de *calcul des séquents*.

Pour faire bref, un *séquent* est une relation de déduction qui peut exister entre deux séries de formules, l'une qui figure dans l'antécédent (et doit être comprise comme une conjonction) et l'autre qui figure dans le conséquent (et doit être comprise comme une disjonction). Par exemple :

$$(2) \quad a \vee b, a \Rightarrow c, b \Rightarrow d \vdash c, d$$

est un séquent prouvable. Il signifie qu'on peut déduire de « a ou b », de « si a alors c » et de « si b alors d », la disjonction « c ou d ». Parmi les règles du calcul des séquents figurent deux règles remarquables : l'une par sa simplicité, qu'on appelle axiome d'identité (elle n'a pas besoin de prémisse pour être prouvée) :

$$(\mathrm{Id}) \qquad A \vdash A$$

l'autre par sa commodité :

$$(\mathrm{Cut}) \quad \frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

On notera que l'une est en quelque sorte l'envers de l'autre : si nous appelons « contextes » les suites de formules incluses dans les lettres majuscules grecques et si nous mettons un

temps ces contextes entre parenthèses, et si d'autre part, nous considérons les formules de gauche comme ayant une certaine polarité ( - ) et celles de droite comme en ayant une autre (+), nous voyons que (Id) établit un lien d'un A de polarité négative vers un A de polarité positive alors que (Cut) établit un lien (souterrain !) entre un A de polarité négative vers un A de polarité positive. D'une manière moins abstraite et moins esthétique, notons que la règle dite « Cut » (« coupure » en Français…) est tout simplement la règle qui permet d'introduire un lemme dans une démonstration. Si nous savons que la formule A est vraie sous certaines hypothèses (dans un certain contexte), alors nous pouvons l'utiliser pour prouver autre chose, et dans ce cas, on pourra dire qu'on a obtenu la preuve de cette autre chose dans l'union des contextes (on « oublie » A). Le résultat remarquable de Gentzen (le « Hauptsatz », ou « théorème d'élimination des coupures », démontré en 1934) est que, dans ce système, toute preuve obtenue *avec* coupure peut être transformée en une preuve du même résultat obtenue *sans* coupure. La preuve du Hauptsatz est constructive, au sens où elle donne un algorithme pour construire la preuve sans coupure à partir de la preuve avec coupure[3].
Autrement dit, partant d'une preuve « générale », non effective[4] (ou seulement partiellement effective), il est possible en appliquant cet algorithme d'obtenir une preuve totalement effective, ce que nous avons appelé aussi une preuve « concrète ». On peut dire que l'élimination des coupures est une « élimination de l'abstraction » dans les preuves.
Ce théorème d'élimination des coupures a son correspondant dans d'autres systèmes de présentation de la logique, et notamment en déduction naturelle, sous la forme de théorèmes de *normalisation*.
Qui plus est, il a été découvert par la suite (dans les années cinquante – soixante), que les règles appliquées en logique intuitionniste trouvaient leurs exacts correspondants dans un calcul qui avait été inventé par Church également dans les années trente et dont on avait démontré l'équivalence avec les machines de Turing (d'où la fameuse « thèse de Turing – Church ») : le *λ-calcul*. Cette correspondance, dite « de Curry-Howard », s'étend à l'élimination des coupures : l'élimination des coupures dans une preuve est exactement la même chose que le processus de normalisation d'un lambda-terme, lequel est fondé sur ce que, justement, on appelle : la ***λ*-conversion** (nous voici enfin revenus au thème du séminaire !), dont la définition est donnée ci après (d'après Hindley, Lercher et Seldin, 1972).

## 3.2 Lambda-convertibilité

On dit que X est **λ-convertible** en Y (« X |> Y ») si et seulement si cet énoncé peut être déduit dans le système formel suivant :

Axiomes :

a. λy. Z |> λv. [v/y]Z si y n'est pas lié dans Z et v ne figure pas déjà dans Z
b. (λx. M) N |> [N/x]M
c. M |> M

Règles :

d. X |> X' ⇒ ZX |> ZX'
e. X |> X' ⇒ XZ |> X'Z
f. X |> X' ⇒ λx. X |> λx. X'
g. X |> Y et Y |> Z ⇒ X |> Z

[3] Voir en particulier pour une démonstration (Girard, Lafont, Taylor, 1989)
[4] ?analytique

Or, l'équivalence entre λ-calcul et machines de Turing nous a mis sur la voie : les expressions du λ-calcul (les λ-termes) *sont* des programmes et la normalisation des λ-termes correspond exactement à *l'exécution* de ces programmes. Nous obtenons donc un étonnant parallélisme entre les notions de formules et de preuves (« analytiques » ou non) d'un côté et celles de programmes et de calculs (« effectifs » ou non) de l'autre. L'ensemble de ces résultats a bouleversé dans les deux dernières décennies notre conception classique de la logique (celle d'Aristote certes, mais tout aussi bien celle de Frege), pour installer à la place une conception dynamique qui ne met plus au cœur de ses intérêts la notion statique d'argument, mais celle de *preuve*, avec toutes les transformations qu'elles peuvent subir. Ceci conduit Jean-Baptiste Joinet[5] à dire : « aujourd'hui, en logique, les preuves sont les effets de bord d'un processus dynamique : la conversion ».

## 3.3 Logique et processus

La démonstration du théorème d'élimination des coupures n'est possible que parce que les règles ont une certaine forme, en particulier parce que les règles d'introduction en partie gauche ou en partie droite d'un même symbole ont une importante caractéristique de symétrie : cela introduit une contrainte forte sur l'ensemble des systèmes logiques possibles (ou « utiles »). Dans l'algorithme d'élimination, une règle de logique n'est plus vue comme un simple passage d'un énoncé à un autre, elle devient une « action élémentaire définie par ses capacités d'interaction avec les autres règles » (J. B. Joinet, ibid.).

D'autre part, on demande à une logique d'être « convergente » au sens suivant : il peut arriver que dans la dynamique de transformation des preuves, plusieurs ordres d'application des opérations soient possibles. On souhaite évidemment que le résultat du processus ne dépende pas de l'ordre choisi (on dit aussi que le processus de normalisation, ou de conversion, doit être *confluent*). Or cela n'est pas toujours le cas ! et en particulier, cela ne peut pas être le cas en logique classique ! D'où l'intérêt portée à des logiques dites non classiques, mais qui du point de vue de ce critère se comportent mieux que les logiques classiques : logique intuitionniste, logique linéaire notamment. Ceci dit, la manière dont la logique classique diverge n'est pas très grave : la divergence peut être maîtrisée.

D'autre part, mais ce serait trop long d'expliciter ici ce point, toutes ces logiques peuvent encore être vues sous un autre angle, que Girard et al. qualifient de « géométrie de l'interaction », (Girard, 1987, Lafont, 1995) celui d'un ensemble d'interactions entre des processus qui s'échangent de l'information (réseaux de preuves etc.).

Cela a pour conséquence que certains chercheurs d'aujourd'hui, changeant complètement l'objet de la logique (qui, jusque là était une sorte de « théorie de la vérité » ou une « théorie des inférences correctes ») font de celle-ci *une science formelle des processus informationnels convergents*.

La logique n'est plus vue comme un ensemble de cadres et de recettes permettant à un « calculateur » ou à un « démonstrateur » d'accomplir sa mission : elle a pour vocation ambitieuse de prendre comme objets les processus (de production et d'échange d'information) tels qu'ils se trouvent, oserait-on dire « dans la nature », sans qu'il soit nécessaire de remonter à leurs auteurs (si tant est que cette notion ait un sens) pour en décrire les mécanismes à condition qu'ils soient « raisonnables » ou « maîtrisables ». Les objets obtenus sont des *preuves analytiques locales*. On ne vise pas à rendre analytique *l'ensemble* d'un processus : la logique a pour but d'enfermer l'infini des processus dans du fini maîtrisable.

[5] J. B. Joinet, 2005, communication orale à la journée « le logique et le biologique », 22 avril 2005, Paris-Sorbonne

Si nous faisons le bilan de cette première partie, en établissant un lien avec le thème général de la conversion, nous voyons que :

- la logique est posée comme une science générale formelle des processus informationnels (convergents !),
- elle étudie la dynamique d'interaction entre de tels processus, laquelle est basée sur un ensemble de principes de conversion simples comme *la λ-conversion*,
- la conversion étant la base de cette dynamique, nous sommes libres à tout instant de la faire agir ou non. Lorsque nous la faisons agir, nous obtenons des objets symboliques analytiques concrets, tels des « preuves concrètes », ou d'une manière peut-être plus parlante : des formes symboliques traduisant l'effectuation de certains programmes,
- dans ce tableau d'ensemble, ce qui est au centre, ce ne sont pas les preuves concrètes (donc pas les exécutions de programmes), mais la dynamique de la conversion.

# *4 Des processus ailleurs qu'en logique*

## 4.1 Biologie

### 4.1.1 L'information biologique

Notre deuxième partie concernera la biologie et la linguistique. Pourquoi envisager ces disciplines dans un tel contexte ? Pour la raison, et là est notre thèse, qu'ils s'insèrent dans un contexte où la notion de *processus informationnel* (convergent) est reine.
Partons de la biologie. On sait que les notions de *code*, d'*information* et de *transfert d'information* y jouent un rôle central. Dans un petit livre publié en 1944, « Qu'est-ce que la vie ? », Erwin Schrödinger introduisait la notion de code, alors même que l'informatique n'existait pas encore. Evoquant le rôle des chromosomes, il écrivait : « ce sont ces chromosomes […] qui contiennent sous la forme d'une *espèce de code*, le modèle intégral du développement futur de l'individu et de son fonctionnement dans l'état adulte »[6]. Plus loin, il précisait : « en donnant à la structure des fibres chromosomiques le nom de code, nous entendons signifier que l'esprit omniscient conçu un jour par Laplace, et pour qui tout rapport causal serait immédiatement déchiffrable, pourrait immédiatement déduire de cette structure si l'œuf, placé dans des conditions convenables, se développerait en coq noir ou en poule tachetée, en mouche ou plante de maïs, rhododendron, scarabée, souris ou femme ». Commentant ce petit livre à l'époque contemporaine (1986), Antoine Danchin met l'accent sur ce qu'il y a de profondément neuf dans la démarche du physicien autrichien : « ce qui est central dans l'argumentation de Schrödinger, et que comprendront certains de ses lecteurs seulement, est que le modèle sur lequel sera construit l'organisme, le « moule intérieur » de Buffon est un objet physique, *lui-même manipulé par les règles qu'il spécifie* », mais, ajoute Danchin, « ce que Schrödinger n'a pas encore compris, et ce que découvriront les véritables créateurs de la Biologie Moléculaire, c'est que ce double rôle, de matrice et de substrat, ne peut prendre de sens qu'à la suite d'une transposition d'un élément de sens – le gène – en un autre élément de sens – le produit actif du gène. Il faut au moins deux niveaux pour que le code (au sens de programme) puisse jouer son rôle créateur de forme. Et aujourd'hui le mot

[6] « Qu'est-ce que la vie ? de la physique à la biologie », E. Schrödinger, trad. Léon Keffler, ed. du Seuil, coll. Points, n°S94, p.71

de code a perdu le sens initial de programme que lui donnait Schrödinger, pour prendre celui d'intermédiaire symbolique entre le gène et le produit du gène ». On trouve également chez Schrödinger déjà la métaphore du gène comme texte écrit avec un alphabet de base (il prend l'exemple de l'alphabet Morse), le produit du gène devenant alors le résultat d'une *traduction* du premier texte. Transposition, traduction, conversion : autant de mots voisins pour signifier à notre avis ce qui demeure le plus fondamental, à savoir le fait que les processus vitaux mettent en jeu une pluralité de niveaux (au moins deux) avec des mécanismes de passage de l'un à l'autre qui s'effectuent selon certaines règles.

### 4.1.2 Réplication et unicité

Tous les livres élémentaires de biologie le disent[7], « la propriété fondamentale de la vie, celle qui différencie les organismes vivants du monde inanimé, c'est *la reproduction* » (Dujon, 2005). On peut ajouter qu'il s'agit le plus généralement d'une reproduction « conforme » (une souris fille ressemble à une souris mère) mais néanmoins compatible avec *l'évolution*. La biologie contemporaine montre que ces deux aspects ne sont pas contradictoires en s'appuyant sur les mécanismes moléculaires de l'hérédité. Au centre de ces mécanismes figurent les notions de *transcription* et de *réplication* : Dujon (op. cit.) cite un biologiste (Michael Yarmolinsky) pour qui « l'évolution des gènes et des génomes n'est finalement qu'un petit problème de « répliconnerie ». On considère alors l'information comme *« l'invariant » dans des processus biologiques qui visent à la réplication*. La vie reproduit du similaire, mais cette notion de similarité elle-même ne peut être définie en dehors d'une référence à la notion d'information. « Similaire » (et non « identique ») s'entend dans le sens justement où une même « forme » se trouve reproduite dans la copie par rapport à l'original. La réplication, comme d'ailleurs la mutation, sont subordonnées à *l'information*. Laquelle, qui plus est, semble être un prélude à la notion *d'individuation*, qui joue un rôle si important dans le monde du vivant. Dans le monde physique, en effet, deux morceaux de matière peuvent être rigoureusement identiques, avec leurs atomes et molécules interchangeables alors qu'on sait qu'il n'en est pas de même pour le vivant, où chaque organisme est *unique*.

C'est, selon certains chercheurs, le surgissement dans l'univers d'un « pôle informationnel » qui serait responsable de ce caractère d'unicité. Vincent Danos, un logicien profondément investi dans une collaboration avec les biologistes, fait ainsi remarquer que si on extrayait du vivant le mode de fonctionnement de l'ADN, alors la vie s'arrêterait car les protéines ne seraient plus produites : le vivant se dissoudrait sans laisser de traces, si ce n'est des atomes de carbone. Le cycle biochimique n'est pas suffisant pour l'explication de la vie : se superpose à lui un « cycle informationnel ». L'existence de ce dernier permettrait de comprendre la raison d'être de la duplication elle-même, et donc de la vie : l'information est en effet un capital bien fragile (non stockable dans l'espace de manière tangible) et pour perdurer, elle n'a guère d'autre solution que la réplication permanente, laquelle inclut la dimension temporelle (« plus on est de l'information, plus on doit se dupliquer ! »), d'où l'ADN et sa structure en hélice comme solution unique.

### 4.1.3 Le programme génétique et la conversion

L'accent mis sur la notion d'information, l'hypothèse de l'existence de processus informationnels à la source même de la vie expliquent qu'aujourd'hui des logiciens se lancent dans la collaboration avec des biologistes pour modéliser les mécanismes cellulaires, en trouvant en eux des similarités avec le fonctionnement des machines de Turing à propos notamment de la notion de *transcription*. La notion d'interaction intervenue récemment en logique est également mise à profit. On citera parmi les principaux biologistes engagés dans

[7] cf. le récent « Comment évoluent nos gènes » de Bernard Dujon, 2005

ce mouvement Antoine Danchin, qui intitule une de ses conférences : « The cell as a living computer »[8]. Dans cette conférence, on trouve une caractérisation profonde des différentes disciplines scientifiques du point de vue des notions fondamentales qu'elles mobilisent :

« - Physique : matière, énergie, temps…
- Biologie : Physique + information, codage, contrôle…
- Arithmétique : chaînes d'entiers, récursivité, codage…
- Informatique : arithmétique + programme + machine… »

Danchin établit un parallèle entre la construction d'une machine et celle d'une cellule : « comme dans le cas de la construction d'une machine, dans celui de la construction d'une cellule, on a besoin d'un livre de recettes… cela demande ensuite qu'on soit capable de changer le texte de la recette en quelque chose de concret : ceci consiste dans le « transfert d'information ». Dans une cellule, ce transfert d'information est assuré par le programme génétique ». On remarquera ici la similitude avec ce que nous avons vu dans la première partie : la dynamique d'élimination des coupures (ou de normalisation, ou de λ-conversion, comme on voudra) avait pour but de changer le texte d'une preuve (mais une preuve « non analytique », comportant certains lemmes) en le texte d'une autre preuve (mais complètement « analytique », une preuve concrète), de la même manière, le programme génétique est ici supposé s'appliquer pour changer le texte de l'ADN (la « recette ») en quelque chose de concret : la production de protéines bien définies.

## 4.2 Langage

### 4.2.1 Pourquoi le langage ?

Venons-en maintenant au langage. La réflexion sur l'origine du langage a redémarré ces dernières années (après avoir été littéralement bannie de la communauté linguistique par une décision de congrès de linguistes du XIX$^{ème}$ siècle), mais elle n'a pas encore fourni beaucoup de certitudes. A peine commençons-nous à penser que ce n'est pas seulement à des fins de *communication* que le langage humain, avec tout ce qu'il comporte de complexité (phonologie, morphologie, syntaxe…) est apparu dans l'espèce. La communication existe à tous les niveaux de la vie et en particulier dans toutes les espèces animales. En un sens lié à la seule opérationnalité, les abeilles communiquent mieux que nous. Notre langage est trop complexe justement pour que nous puissions bien « communiquer », nous en faisons l'expérience chaque jour ! La critique de la notion de « code » en linguistique n'est également plus à faire. Alors pourquoi le langage humain ? quel « avantage » a-t-il pu bien fournir à l'espèce qui l'a développé dans le cours de l'évolution ? Nous n'allons pas entrer ici dans un débat actif entre les partisans du « tout politique » (le langage sert à des joutes oratoires qui permettent de sélectionner le meilleur « chef ») et du « tout séduction » (les beaux parleurs ont toujours l'avantage auprès des femmes et de ce fait, se reproduisent mieux que les autres !). Nous suggérerons simplement qu'il puisse y avoir une hiérarchie entre *trois* « mondes » (au lieu de deux), l'un de ces trois mondes n'étant pas abordé dans la littérature biologique comprenant les écrits de Danchin :

- le physique : matière, énergie, temps
- le biologique : information, codage, contrôle
- le linguistique : pertinence, inférence, dialogue

[8] (consultable à l'adresse : http://www.pasteur.fr/recherche/unites/REG/lectures/Stockholm1_04.pdf).

ou, plus brièvement : si le biologique se définit comme « le physique + l'information », le linguistique ajoute à l'information *la pertinence*.
Sur quoi étayer cette hypothèse d'un troisième niveau ? Si nous retenons l'idée que le biologique est solidaire d'un « pôle informationnel » qui ne peut se survivre que par la reproduction et qui est responsable d'une certaine individuation (au niveau biologique), nous suggèrerons en parallèle que la *pertinence*, en quoi nous reconnaissons la marque du linguistique, ne peut se perpétuer que par la *continuation* des discours (dans un flux de langage qui ne se tarit jamais depuis les origines) et qui serait responsable de ce processus tout aussi fondamental pour les humains que l'individuation pour les êtres biologiques : la *socialisation*, et partant évidemment, de l'achèvement de leur individuation, au sein du *social*.
En résumé, la notion de programme génétique n'est plus une simple métaphore à partir du moment où on peut décrire les mécanismes cellulaires comme des « ordinateurs vivants ». De même, les mécanismes de l'interprétation sémantique, comme nous le verrons plus loin, sont similaires à ceux que l'on trouve en matière d'évaluation de programmes, ce qui rend l'interprétation de la phrase analogue à une évaluation de forme symbolique. Cependant, il ne s'agit jamais de « tout » évaluer, mais seulement, dans un flux permanent d'informations, les fragments *pertinents*.
Si la vie trouve son origine dans la nécessité de maintenir un cycle informationnel, on peut suggérer que le langage (et tout ce qu'il draine avec lui en matière de symboles et donc de culture et d'organisation de la société) trouve la sienne dans la nécessité de maintenir *un flux de pertinence dans l'information*, tâche qui s'effectue principalement par le dialogue et plus généralement par ce que Wittgenstein appelait des *jeux*. La logique et les mathématiques s'intéressent alors aux structures formelles que cela met en œuvre. De même que la vie trouve une solution dans une structure formelle : celle de l'hélice de Crick et Watson, l'hominisation en trouverait une, voire plusieurs, dans la structure formelle liée à l'activité langagière (dialogique, conversationnelle etc.). Dans les deux cas, ce que nous entendons par « structure formelle » réside dans l'identification de quelques invariants qui assurent la reproduction[9], la réplication et la diffusion.

### 4.2.2 La dynamique de la conversion dans le langage

En tentant l'analogie entre le langage et le biologique, nous retrouverons *l'idée de dynamique de la conversion* (la même qui nous a déjà fait passer de la preuve non analytique à la preuve analytique, du texte du programme à son exécution, du texte de l'ADN à la production de protéines et qui pourrait aussi nous faire passer de la recette de la crêpe à la crêpe…).
Des linguistes et psycho-linguistes (Thomas Bever et David Townsend, 2002) ont mis en évidence l'existence de deux phases dans la compréhension d'un discours ou d'un texte (le discours étant plutôt oral et le texte écrit) : une analyse initiale d'un sens vraisemblable et une reconstitution de la structure syntaxique dérivationnelle complète cohérente avec la forme et le sens. Ils relient cette théorie de la compréhension à une théorie semblable en ce qui concerne la perception : lorsque nous cherchons nos lunettes dans l'appartement, disent-ils, et que finalement nous les trouvons, nous les avons en réalité « vues » deux fois, la première est passagère, « mais elle laisse le clair souvenir d'un évènement qui paraît pré-perceptif », la seconde est bien sûr au moment où nous les trouvons, et elle nous paraît être la première parce que la vraie première a été immédiatement enfouie dans notre inconscient. Il en irait semblablement avec le langage : le tout-venant du discours et du texte est écouté ou lu sans nécessairement d'attention particulière, comme un balayage furtif exercé par notre ouïe ou

[9] On notera que c'est le même mot, « reproduction » qui est utilisé aussi bien par les biologistes en ce qui concerne la vie, que par les sociologues (cf. Bourdieu) en ce qui concerne la société.

notre regard de lecteur, et ce n'est qu'en certains moments bien localisés (lorsque nous ne sommes pas sûrs de bien comprendre quelque chose ou bien lorsque nous cherchons une information précise et que nous voulons confronter ce qu'on lit ou ce qu'on entend avec la requête que nous avons en tête) que nous procédons à l'analyse détaillée (« dérivationnelle » pour reprendre le qualificatif très chomskyen utilisé par les auteurs) d'un fragment du discours proféré ou du texte lu. Autrement dit, là aussi, nous aurions une différenciation entre deux niveaux tout aussi différents que le sont le texte de l'ADN et sa conversion en une suite d'actions biologiques déterminées, mais se situant entre celui des instructions générales évoquées par les mots et les phrases mais pas nécessairement évaluées et la mise en activation de ces instructions afin de produire un « texte concret », c'est-à-dire une évaluation « mentale » authentique. Celle-ci peut se traduire alors par la reconnaissance d'un accord avec une requête ou par une séquence d'actes : cas d'un ordre par exemple, où là, l'opération est très similaire à ce qui se passe dans la cellule, ou bien encore par la production par soi-même d'autres phrases et d'autres mots, dans une situation de dialogue, et dans le but de poursuivre le flux langagier, selon un processus après tout similaire… à une réplication.

### 4.2.3 Du langage… aux programmes

Le rapprochement opéré dans la section précédente est d'autant plus fondé qu'il existe de nombreux travaux de linguistique formelle qui utilisent justement les outils présentés dans notre première partie, à savoir l'isomorphisme de Curry-Howard, pour produire non pas des « programmes » mais des « représentations sémantiques » (mais pouvons-nous vraiment établir une séparation tranchée entre les deux ?). De plus, des informaticiens (P. de Groote, 2001) et des linguistes informaticiens (C. Barker, 2002) ont montré qu'il y avait un parallélisme étroit entre certains styles de programmation développés dans les langages fonctionnels (LISP, Scheme…) et les mécanismes d'évaluation en vigueur dans la langue. Essayons de voir de quoi il s'agit.

C'est le philosophe et logicien Richard Montague qui a posé les bases d'une sémantique formelle des langues naturelles reposant sur un homomorphisme entre structures syntaxiques et structures sémantiques (cf. Montague, 1974). La grammaire de Montague consiste en un ensemble fini de règles de syntaxe (souvent formulées il est vrai dans un formalisme quelque peu *ad hoc*) assorties de contreparties sémantiques sous la forme d'instructions particulières visant à obtenir la signification d'un constituant à partir de celles de ses parties immédiates. La sémantique de Montague est à la fois *compositionnelle* : on y retient le principe de Frege selon lequel la signification d'une expression linguistique est fonction des significations de ses composants, et *dénotationnelle* : elle vise à attribuer aux phrases des valeurs de vérité et, ce faisant, les constituants intermédiaires, par exemple les syntagmes nominaux et les syntagmes verbaux, se traduisent en termes ensemblistes. On peut évidemment critiquer une conception de la sémantique pour laquelle on en arrive toujours à dire que le sens d'un énoncé comme « Pierre embrasse Marie » est donné par un univers contenant au moins deux éléments a et b, référents de « Pierre » et de « Marie » et d'une relation binaire R sur cet univers interprétant le verbe « embrasse » de sorte que « Pierre embrasse Marie » soit vrai si et seulement si le couple (a, b) appartient à l'extension de la relation R (!), pourtant l'essentiel n'est pas là, mais réside plutôt dans le fait que la représentation logique utilisée (logique du premier ordre ou bien logique intensionnelle) a une puissance d'expression suffisante pour permettre d'analyser des phénomènes relevant authentiquement de notre manière d'interpréter les énoncés comme : les références pronominales (le fait que les pronoms du langage se comportent *grosso modo* comme des variables liées dans un langage logique), la portée des expressions quantifiantes (comme *tous les villageois, au moins un villageois*, par extension : *la plupart des villageois, plus d'un villageois sur deux* etc.), les ambiguïtés de lecture reposant sur des interprétations *de re* ou *de dicto*, le rôle des items de polarité négative etc.

Dans les années quatre-vingt la sémantique de Montague a été intégrée au modèle génératif hérité de Noam Chomsky, dans le cadre d'un modèle de grammaire appelé « Grammaire Syntagmatique Généralisée » élaboré par G. Gazdar, E. Klein, S. Pullum et I. Sag, donnant ainsi une formulation plus harmonieuse de la grammaire de Montague (Gazdar, Klein, Pullum, Sag, 1985). Dans une telle grammaire, nous avons des règles de réécriture comme : $S \rightarrow SN\ SV$, complétées avec des instructions du genre : « pour obtenir la sémantique du constituant S, appliquer la sémantique du SV à la sémantique du SN ». La notion d'application ici mentionnée appelle immédiatement l'idée que les représentations sémantiques sont des *entités fonctionnelles*. Nous retrouvons ainsi le λ-calcul évoqué plus haut. Par exemple, dans le cas particulièrement simple de cette règle, on doit imaginer que la représentation sémantique du SV est un λ-terme, comme : $\lambda x.\ \textit{écoute\_le\_chant\_des\_étoiles}(x)$, de même que celle du SN, qui pourrait être ici : **pierre**. La représentation de la phrase « Pierre écoute le chant des étoiles » se construit alors par application du premier λ-terme au deuxième : $(\lambda x.\ \textit{écoute\_le\_chant\_des\_étoiles}(x), \textbf{pierre})$ et les règles de conversion nous donnent : $\textit{écoute\_le\_chant\_des\_étoiles}(\textbf{pierre})$. Les choses ne sont pas si simples, puisque si nous voulons obtenir une représentation de « un promeneur écoute le chant des étoiles », il est peu probable que nous nous satisfassions d'une représentation comme : $\textit{écoute\_le\_chant\_des\_étoiles}(\textbf{un\_promeneur})$. Obtenir une formule comme $\exists x\ (promeneur(x)\ \&\ \textit{écoute\_le\_chant\_des\_étoiles}(x))$ va nécessiter un niveau d'abstraction supplémentaire nous conduisant à considérer un syntagme nominal non plus comme une entité individuelle (une constante) mais comme un prédicat du second ordre (ou en termes ensemblistes : un ensemble d'ensembles). « Un promeneur », c'est la propriété, pour la propriété « d'être un promeneur », d'être vraie d'au moins un individu. La représentation sémantique sera donc : $\lambda P.\ \exists x\ (\mathrm{promeneur}(x)\ \&\ P(x))$. Mais alors l'application ne se fera plus du SV vers le SN, mais du SN vers le SV, en effet $(\lambda P.\ \exists x\ (promeneur(x)\ \&\ P(x)), \lambda x.\ \textit{écoute\_le\_chant\_des\_étoiles}(x))$ donne par conversion :

- d'abord : $\exists x\ (promeneur(x)\ \&\ (\lambda x.\ \textit{écoute\_le\_chant\_des\_étoiles}(x), x))$
- puis : $\exists x\ (promeneur(x)\ \&\ \textit{écoute\_le\_chant\_des\_étoiles}(x))$

Là encore, on pourra prétendre que c'est bien peu saisir du sens des phrases que n'en obtenir que des représentations sous forme de formules de logique en fin de compte assez triviales. Néanmoins, si nous avons maintenant « tout promeneur écoute le chant d'une étoile », va apparaître une question touchant à l'ambiguïté d'une telle phrase : s'agit-il d'une seule étoile, ou bien l'étoile dépend-elle du promeneur ? L'intérêt de la traduction de phrase en formules réside dans le fait que le langage formel cible permet de discriminer nettement les deux lectures, qui correspondent respectivement à :

- $\forall x\ (promeneur(x) \Rightarrow \exists y\ \textit{étoile}(y)\ \&\ \textit{écoute}(x, \_le\_chant\_de(y)))$ et
- $\exists y\ \textit{étoile}(y)\ \&\ \forall x\ (promeneur(x) \Rightarrow \textit{écoute}(x, \_le\_chant\_de(y)))$

Une telle théorie garde cependant un caractère *ad hoc* dans la mesure où l'attribution d'instructions d'assemblage des significations à des règles génératives paraît arbitraire. Cette arbitrarité sera supprimée et les règles d'assemblage encapsulées dans le mécanisme générateur si on passe à une autre conception de la grammaire, beaucoup plus « lexicalisée » et où les règles, réduites à un petit nombre, auront un caractère de généricité permettant de *les mettre en facteur par rapport à toutes les grammaires possibles*. Les règles en question, à cause de leur très grand niveau de généralité, mériteront alors d'être vues comme constituant *une logique*, une logique du signifiant en quelque sorte. C'est Jim Lambek qui a le premier développé cette conception dès la fin des années cinquante (Lambek, 1958). Son système est une préfiguration de la logique linéaire de J. Y. Girard dans la mesure où, pour la première fois, il développe l'idée de *sensibilité aux ressources*. On peut décrire à un niveau très général le fonctionnement d'une phrase comme un processus de consommation et d'échange. Dans

« Pierre dort », une unité (« dort ») qui exprime un état, attend une autre unité (« Pierre »), exprimant, celle-ci, un individu, pour finir par donner une unité autonome (« saturée »). Autrement dit, le verbe consomme son argument. Il le consomme littéralement dans la mesure où une fois que le processus s'est accompli une fois, il ne va pas s'accomplir une deuxième : je peux ajouter d'autres noms renvoyant à des individus devant « Pierre », ils ne seront pas « consommés ». De la même manière, un modifieur consomme une ressource pour en produire une autre, de même type, ainsi « bleue » peut consommer « étoile » pour produire à la place : « étoile bleue ». Les règles du calcul de Lambek[10] sont les suivantes, exprimées dans le formalisme du calcul des séquents :

$$\text{(Id)} \quad A \vdash A$$

$$\text{(Cut)} \quad \frac{\Gamma \vdash A \quad \Gamma', A, \Delta \vdash B}{\Gamma', \Gamma, \Delta \vdash B}$$

$$[/\ D] : \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash B / A} \qquad [\backslash\ G] : \quad \frac{\Theta \vdash A \qquad \Gamma, B, \Gamma' \vdash C}{\Gamma, \Theta, A \backslash B, \Gamma' \vdash C}$$

$$[\backslash\ D] : \quad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \backslash B} \qquad [/\ G] : \quad \frac{\Theta \vdash A \qquad \Gamma, B, \Gamma' \vdash C}{\Gamma, B / A, \Theta, \Gamma' \vdash C}$$

Ce qui est particulièrement intéressant est qu'on peut « décorer » ces règles au moyen de l'homomorphisme de Curry-Howard. Ainsi, à chaque étape de dérivation, un pas de plus sera accompli dans la construction d'une représentation logico-sémantique de la phrase analysée.

$$\text{(Id)} \quad x : A \vdash x : A$$

$$\text{(Cut)} \quad \frac{\Gamma \vdash u : A \quad \Gamma', x : A, \Delta \vdash \gamma : B}{\Gamma', \Gamma, \Delta \vdash \gamma[u / x] : B}$$

$$[/\ D] : \quad \frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : B / A} \qquad [\backslash\ G] : \quad \frac{\Theta \vdash u : A \qquad \Gamma, f(u) : B, \Gamma' \vdash C}{\Gamma, \Theta, f : A \backslash B, \Gamma' \vdash C}$$

$$[\backslash\ D] : \quad \frac{x : A, \Gamma \vdash u : B}{\Gamma \vdash \lambda x.u : A \backslash B} \qquad [/\ G] : \quad \frac{\Theta \vdash u : A \qquad \Gamma, f(u) : B, \Gamma' \vdash C}{\Gamma, f : B / A, \Theta, \Gamma' \vdash C}$$

On n'a plus besoin d'une stipulation des règles et de leurs contreparties sémantiques : désormais tout réside dans ces quelques règles logiques et dans la définition du lexique. Par

[10] sans produit, car on peut ajouter un autre constructeur, correspondant au produit (conjonction multiplicative) de la logique linéaire, mais sans toutefois apporter plus de capacité générative.

exemple, pour obtenir la même représentation que précédemment pour « un promeneur écoute le chant des étoiles », nous aurons pour le déterminant indéfini l'entrée suivante :

(3) un :: (s/(sn\s))/n : $\lambda P\ \lambda Q\ \exists x\ (P(x)\ \&\ Q(x))$

Le théorème d'élimination des coupures est valide dans le calcul de Lambek : cela signifie tout simplement que l'on peut toujours « brancher » ensemble des significations obtenues séparément et que l'élimination de la coupure traduisant ce branchement va composer entre elles les significations de manière à obtenir une signification globale. Intuitivement, cela correspond au rétablissement d'un sens « canonique » à partir du sens d'expressions données d'une manière quasi-« parataxique » : à un certain niveau d'abstraction, « Pierre, le soleil, il l'a en plein dans les yeux » a le même sens que « Pierre a le soleil en plein dans les yeux »[11]. Autrement dit, vous pouvez donner des morceaux d'expressions linguistiques dans un relatif désordre : la dynamique de conversion des preuves s'en arrangera et vous pourrez obtenir finalement une signification canonique stable.
Il serait toutefois faux de croire que les ambiguïtés de lecture liées à une pluralité de quantificateurs, chacun avec sa propre portée, puissent être résolues si facilement dans le calcul de Lambek : il faut au moins supposer une pluralité de types pour les déterminants comme « un » et « tous ».
Une autre solution est basée sur *les continuations* (en analogie avec les transformations CPS (*Continuation Passing Style*) connues des informaticiens). En quelques mots, la programmation par continuation repose sur l'idée qu'on évalue toujours des flux de calculs : l'évaluation courante doit se tenir prête à accepter (à être éventuellement modifiée par) une suite du calcul a priori imprévue (par exemple une interruption due à une erreur). Elle prend donc les contextes en arguments. T. Griffin a démontré que cette conception permettait d'étendre l'isomorphisme de Curry-Howard à la logique classique (Griffin, 1990), avec cette réserve cependant : en logique classique, les conversions ne sont pas confluentes, autrement dit des résultats différents peuvent provenir d'ordres d'évaluation différents entre les arguments. C'est justement ce qui se passe avec une phrase comme (4) où l'application du prédicat (« ont enterré ») à ses arguments possède deux ordres possibles d'évaluation, menant aux deux lectures distinctes.

(4) Tous les villageois ont enterré un poète

On voit donc qu'en linguistique également, *l'idée de programme* n'est pas qu'une métaphore : le sens des phrases se calcule comme on évalue le texte d'un programme.
On notera cependant que le développement de la linguistique (dite « computationnelle ») est limité par une impasse tant que nous nous limitons à la production de formules pour représenter le sens car *le sens d'une expression n'est pas une formule* : si c'était le cas, il faudrait donner le sens de cette formule… au moyen d'une autre formule et ainsi à l'infini.
Il faut donc franchir un cap supplémentaire dans la réflexion sur le langage. Pour cela, nous serons peut-être aidés à nouveau par les évolutions récentes de la logique et par la façon dont l'idée de conversion elle-même peut conduire à des métamorphoses quant à la manière de concevoir les processus.
Si le « sens » n'est certainement pas figeable en des formules parfois absconses, nous sommes de plus en plus tentés de le considérer « en action ». Le sens de mon énoncé est dans la

[11] Une telle phrase n'a évidemment pas d'interprétation « dénotationnelle », on serait bien en peine de trouver un modèle où le soleil est littéralement dans les yeux de Pierre, mais à ce niveau d'analyse, cela importe peu, l'essentiel étant de montrer qu'on peut toujours combiner des significations obtenues antérieurement. La phrase est un montage, on peut laisser provisoirement dans l'ombre ce dont sont faits les morceaux…

manière dont il est poursuivi par moi ou par mon interlocuteur dans une interaction dialogique qui ne s'arrête en réalité jamais. Mon discours embraye sur celui d'autrui et sert d'embrayeur à son tour pour qui prendra ma suite ou me répondra. Les énoncés émergent non pas d'un « sujet idéal » à la source du sens, ni d'un point abstrait d'un espace, à partir duquel se génèreraient tous les énoncés possibles dans le cadre d'une grammaire générative globale et définitive, mais d'un « inter - discours » au sein duquel se constitue une mémoire discursive à l'œuvre pour chacun. La signification devient alors manière de se déplacer dans cet espace, d'y repérer des formes stables ou des bifurcations « catastrophiques ». On peut imaginer que la topologie de cet inter – discours soit celle d'arborescences infinies qui se construisent en interaction constante les unes avec les autres. La notion de *jeu*, au sens logique du terme, peut être utile à penser ces processus car elle embrasse la notion de dialogue et ne présuppose pas un « commencement » absolu. J. Y. Girard a interprété la logique des preuves comme une « géométrie de l'interaction » : nous n'avons pas la place ici de développer ce point, disons simplement qu'il existe une représentation simple des preuves dans certains systèmes (en particulier en logique linéaire) en tant que réseaux, on les appelle « réseaux de preuve ». Une preuve se construit dans de tels systèmes par identifications deux à deux de nœuds appartenant à des sous-réseaux associés à différentes formules. Cette conception a été complétée par une vision plus « théorie des jeux » (que justement Girard qualifie de « ludique ») où la construction d'une preuve se fait au cours d'un dialogue entre un proposant et un opposant, l'opposant essayant à chaque pas de construire une « contre – preuve » à l'argument que propose le proposant. Celui des deux qui perd est celui qui utilise son « daimon » : sorte de moyen de clore la discussion quand on s'avoue vaincu, mais il est possible d'envisager des jeux qui ne s'arrêtent jamais… comme c'est le cas sans doute du « jeu du langage ».

## *5 Conclusion*

Cette contribution au thème de « la conversion » nous a permis de passer en revue des exemples où la conversion apparaît comme fonctionnement fondamental, à la base de mécanismes de transformation vitaux. Il s'agit dans chaque cas de faire interagir des instances d'objets déjà là, qu'il s'agisse de cellules pour produire d'autres cellules au moyen d'un processus de transcription d'un code (celui de l'ADN) en actions (production de protéines) ou qu'il s'agisse de phrases à partir d'autres phrases au moyen d'un processus de combinaison des significations. Le modèle commun à ces mécanismes est le calcul logique, en tant qu'il inclut une dynamique le rendant apte à modéliser les processus. C'est la marque que la logique qui nous fut héritée d'Aristote, toute entière centrée sur les normes de la discursivité, a considérablement changé son objet au fur et à mesure qu'elle était confrontée à la question des fondements des calculs effectués par une machine. Aujourd'hui, la logique s'oriente vers une science générale des processus de calcul tels qu'ils apparaissent dans toutes les disciplines. De plus en plus, elle met au cœur de nos démarches la nécessité de refuser les idées simples comme celle selon laquelle « le complexe viendrait de la combinaison de l'élémentaire » ou bien celle selon laquelle il serait toujours possible de ramener les phénomènes à une origine. Le concept de « conversion » apparaît alors comme particulièrement bien choisi pour représenter ce mouvement : la « conversion » part toujours d'une réalité qui pré-existe, à laquelle simplement on impose une transformation pour en produire une autre, la première n'étant ni plus ni moins complexe que la seconde, mais étant simplement « plus abstraite » au sens de « plus éloignée des réalités concrètes ». Le phénomène de la cognition lui-même nous paraît devoir entrer dans cette perspective : comme le rappelle Sylvain Auroux[12] (Auroux, 1998), l'idée du cognitivisme classique selon laquelle chaque « sujet » est un centre de fonctionnement à penser selon un modèle « computo-représentationnel » semble aujourd'hui conduire à une impasse. De plus en plus, on est conduit à penser l'intelligence (et la cognition) comme des entités « distribuées ». La conversion est alors le processus de transformation actif qui génère du sens à partir « des hordes de suites finies de signes sans signification [qui] hantent notre univers mental » (pour paraphraser une expression qu'emploie G. Longo[13]) et qui, ajouterais-je, pour la plupart, nous pré-existent.

[12] Dans « La raison, le langage et les normes », S. Auroux prétend que « la grande question métaphysique aujourd'hui, en matière de langage, est de savoir si un modèle computationnel suffit à expliquer le comportement linguistique humain. Est-ce que la faculté de langage de chacun d'entre nous et, plus spécialement, sa compétence linguistique s'explique parce que nous avons des algorithmes implémentés dans notre tête ? ». Il conclut dans le sens d'un nécessaire *externalisme*, dont les « hypothèses principales reviennent à soutenir que l'intelligence est originairement un artifice et que ses manifestations sont dépendantes d'instruments externes » (Auroux, p. 7). C'est remarquablement dire que nos pensées et nos actions, loin d'être les produits mystérieux d'une conscience autonome et individuelle, résulteraient de la conversion de systèmes qui nous sont extérieurs et dont les rencontres construisent nos singularités.

[13] voir note 1 ci-dessus

**Annexe**

*Le λμ-calcul*

Il est curieux de constater que le fait de plonger un ensemble de phénomènes d'interprétation touchant à des phrases d'une langue naturelle dans une logique plutôt qu'une autre a pour effet de faire surgir une ambiguïté de lecture à laquelle on ne s'attendait pas forcément. Il est tout aussi curieux de constater que le Schrödinger de 1944 faisait d'une telle pluralité de lectures une propriété importante de la vie lorsqu'il disait : « on a même découvert ce qu'on a appelé des « allélomorphes multiples », c'est-à-dire, deux ou plusieurs « versions » ou « lectures » différentes – en plus de la normale, non mutationnée, correspondant à la même place du code chromosomique »[14]. Cette annexe concerne le premier problème : la logique choisie fait figure d'une sorte d'espace dont les régularités permettent des interprétations plus ou moins riches. Nous prendrons l'exemple de l'opposition entre logique intuitionniste et logique classique.

Quand on présente les logiques intuitionniste et classique sous la forme de *calculs des séquents*, ce qui les différencie au premier coup d'œil, c'est que les séquents intuitionnistes ont une seule formule en partie droite alors que les séquents classiques ont un nombre arbitraire de formules à droite. Les séquents intuitionnistes s'interprètent ainsi facilement en termes fonctionnels : les éléments de la partie gauche sont les *inputs* et l'élément de la partie droite est l'*output*, ce qui est cohérent avec la définition même d'une fonction (une seule valeur est associée à un choix donné de valeurs pour les arguments), d'où le *déterminisme* du calcul intuitionniste (le fait qu'il soit *confluent*). En logique classique, comme il y a plusieurs formules en partie droite, il faut à chaque pas d'un calcul pouvoir sélectionner celle sur laquelle on travaille et qui devient de ce fait la formule courante à droite. Pour cela, il faut d'une part pouvoir nommer les formules et d'autre part pouvoir sauter directement à une formule nommée en utilisant simplement son nom. Cette manière de faire a alors son corollaire du point de vue d'une extension du λ-calcul. Des noms peuvent être attribués à des sous-termes et au cours de l'évaluation (donc au cours de l'exécution du programme) on doit pouvoir sauter[15] directement à certains sous-termes uniquement à partir de leurs noms (on dit aussi « étiquettes »). Les termes du λμ-calcul contiennent donc des termes nommés et des termes non nommés, qui sont définis récursivement de la manière suivante (Parigot, 1992) :

- x est un terme non nommé, si x est une λ-variable,
- λx.u est un terme non nommé, si x est une λ-variable et u un terme non nommé,
- (t u) est un terme non nommé, si t et u sont des termes non nommés,
- μα.e est un terme non nommé, si e est un terme nommé et α une μ-variable,
- [α]t est un terme nommé, si t est un terme non nommé et α une μ-variable.

Les règles de réduction de base sont :

- **Réduction logique** : (λx.u v) > u[v/x]
- **Réduction structurelle** : (μβ.u v) > μβ.u[[β](w v)/[β]w]
- **Renommage** : [α]μβ.u > u[α/β]

où u[[β](w v)/[β]w] est obtenu à partir de u en remplaçant inductivement chaque sous-terme de la forme [β]w par [β](w v).

*commentaire* : la règle de réduction logique est standard, c'est la β-réduction du λ-calcul,

[14] E. Schrödinger, op. cit.

[15] que l'on pense à l'instruction `goto`, très traditionnelle en programmation.

la règle de réduction structurelle dit qu'appliquer un terme $\mu\beta.u$ à v, c'est remplacer dans u tous les sous-termes nommés $\beta$, de la forme donc : $[\beta]w$, par $[\beta](w\ \ v)$, autrement dit aller chercher chaque sous-terme distingué et lui donner v comme argument supplémentaire.
Le système logique servant au typage des $\lambda\mu$-termes contient les règles classiques pour les parties lambda, étendues au second ordre (le système $AF_2$ de Krivine), mais rajoute des règles spécifiques pour le nommage. Comme on est en logique classique, il y a nécessairement des règles pour la négation. La règle d'introduction à droite de la négation permet de changer la place de la formule A et de la faire passer à droite, au prix de sa négation, donc le nommage correspond à la négation, alors que la $\mu$-abstraction correspond à l'élimination de la double-négation. De Groote utilise cet aspect pour formaliser la sémantique des langues naturelles, en faisant de **t** (le seul type « observable ») l'équivalent de $\perp$. Dans la formulation qu'il donne du problème de l'ambiguïté de portée des quantificateurs, le $\lambda$-terme associé à *toute personne* : $\lambda P.\lambda Q.\forall x$ **personne**(x) $\Rightarrow$ Q(x) est remplacé par le $\lambda\mu$-terme : $\mu\alpha.\lambda P.\forall x$ **personne**(x) $\Rightarrow \alpha(x)$.

## Références


Auroux, S., 1998 : *La raison, le langage et les normes*, ed. P.U.F., Paris,

Bailly & Longo, 2004 : 'Incertitude et incomplétude en Physique et en Mathématiques', in Paolo Parrini & Luca Scarantino eds. *Il pensiero filosofico di Giulio Preti*, Guerini & associati, Milano

Barker, C., 2002: 'Continuations and the nature of quantification', *Natural Language Semantics*, 10(3): 211-242,

Bever, T. et Townsend, D., 2002: 'Quelques phrases sur notre conscience perceptive des phrases' pp 147-160 in Dupoux, E. (ed) *Les Langages du cerveau*, ed. Odile Jacob, Paris,

Danchin, A., 2004 : http://www.pasteur.fr/recherche/unites/REG/lectures/Stockholm1_04.pdf

Dujon, B., 2005 : *Comment évoluent nos gènes ?* co-ed. Le Pommier & la Cité des Sciences et des Techniques, Paris,

Gazdar, Klein, Pullum, Sag, 1985 : *Generalized Phrase Structure Grammars*, ed. Blackwell,

Girard, Lafont, Taylor, 1989, *Proofs and Types,* Cambridge Tracts in Theoretical Computer Science 7, Cambridge University Press, Cambridge,

Girard, J-Y., 1987, 'Towards a Geometry of Interaction', *Categories in Computer Science and Logic*, Contemporary Mathematics, AMS 1989, pp. 69-108

de Groote, P., 2001: 'Type raising, continuations, and classical logic', in *Proceedings of the 13th Amsterdam Colloquium*, ed. R. van Rooy et M. Stokhof, 97-101, Institute for Logic, Language and Computation, Amsterdam,

Griffin, T., 1990 : 'A formulae-as-types notion of control' in Conference record of the seventeenth annual ACM symposium on Principles of Programming Languages, pp 47-58, Hindley, Lercher et Seldin, 1972 : *Introduction to Combinatory Logic*, London Mathematical Society, Lecture Note Series 7, Cambridge University Press, Cambridge,

Hofstadter, D., 1979 : *Gödel, Escher, Bach : an Eternal Golden Braid*, Basic Books, New York, trad. franc. 1985 : *Gödel, Escher, Bach : les Brins d'une Guirlande Eternelle*, InterEditions, Paris,

Joinet, J-B., 2005 : *Le logique et le biologique*, introduction à la Journée du même nom, Sorbonne, Paris, et intervention au séminaire sur "la mathématisation" organisé par Sophie Roux en 2006 (dept de philosophie, Université Pierre Mendès-France, Grenoble),

Lafont, Y., 1995, 'From Proof-Nets to Interaction Nets', in (Girard, Lafont, Regnier eds.) *Advances in Linear Logic*, Cambridge University Press,

Lambek, J., 1958 : 'The mathematics of sentence structure', *American Mathematical Monthly*, 65,

Montague, R., 1974 : *Formal Philosophy: Selected Papers of Richard Montague*, edité par Richmond Thomason, Yale University Press, New Haven,

Parigot, M., 1992 : 'λμ-calculus : an algorithmic interpretation of classical natural deduction' in A. Voronkov (ed) *Proceedings of the International Conference on Logic programming and Automated Reasoning*, vol. 624, LNAI, Springer-Verlag,

Schrödinger, E., 1967 : *What is Life?* Cambridge University Press, trad. franç. *Qu'est-ce que la vie?* 1986, ed. Christian Bourgois, Paris