



HAL
open science

Qu'est-ce qu'écrire une loi ? L'écriture informatique en science et en droit

Jean Lassègue, Giuseppe Longo

► To cite this version:

Jean Lassègue, Giuseppe Longo. Qu'est-ce qu'écrire une loi ? L'écriture informatique en science et en droit. Samantha Besson et Samuel Jubé (curateurs). *Concertar les civilisations - Mélanges en l'honneur d'Alain Supiot*, Seuil, 2020, 9782021440577. hal-02904222

HAL Id: hal-02904222

<https://hal.science/hal-02904222v1>

Submitted on 21 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Qu'est-ce qu'écrire une loi ? L'écriture informatique en science et en droit

Jean Lassègue

(DR CNRS - Centre Georg Simmel, Recherches franco-allemandes en sciences sociales, CNRS-EHESS, Paris)

Giuseppe Longo

(DRE CNRS - Centre Cavailles, CNRS-ENS, Paris & School of Medicine, Tufts University, Boston)

Jean Lassègue :

Il y a un impensé dans le rôle que joue l'informatique aujourd'hui : on oublie de remarquer qu'elle est exclusivement une manipulation de signes. Cet impensé a pour corrélat une évaluation subjectiviste de l'informatique : nous sommes fascinés par les prouesses de l'informatique qui sont censées nous « dépasser » alors que l'évaluation de l'informatique devrait être essentiellement sémiotique et porter sur la nature précise de la manipulation de ces signes. Cette manipulation est de nature alphabétique : il s'agit d'une combinatoire de marques graphiques faisant partie d'un répertoire qui a une très longue histoire derrière elle. Commencer par reconnaître la nature alphabétique de l'informatique, y voir la dernière étape en date dans la longue histoire de l'écriture, comme l'a souligné C. Herrenschmidt, voilà ce qui permet de sortir du subjectivisme et de l'imaginaire du dépassement de l'espèce humaine qu'il véhicule.

Giuseppe Longo :

Quant à l'usage de l'alphabet et des signes mathématiques conçus dans une perspective purement formaliste, il faut partir de Hilbert qui, au début du XX^e siècle, propose de fonder les mathématiques sur la certitude de la manipulation formelle de suites de signes, les axiomes et les règles de déduction, indépendamment de leur interprétation, géométrie en particulier. Face à la crise affectant la correspondance séculaire entre la géométrie euclidienne et les espaces cartésiens de la physique newtonnienne, il fallait sauver les mathématiques, la géométrie en particulier, et travailler indépendamment des espaces dans lesquels l'écriture axiomatique acquiert un sens et devient un facteur d'intelligibilité.

JL : Comment travailler sans le sens en mathématiques ?

GL : Au XIX^e siècle des grands géomètres avaient montré l'indépendance du 5^{ème} axiome d'Euclide sur les parallèles, en inventant des géométries possibles, non-euclidiennes, à courbure non-nulle, qui, ultérieurement, seront au cœur d'une révolution en physique, la relativité d'Einstein. Mais quel sens donner à ces géométries, de quels espaces physiques parlent-elles ? Hilbert propose une réponse audacieuse : oublions la signification spatiale de ces géométries, il suffit qu'elle ne soient pas contradictoires. Dans ce but, il encode, en 1899, les différentes géométries dans l'Arithmétique, la Théorie Formelle des Nombres. Il pose donc, lors d'un grand colloque en 1900 à Paris, le problème de fournir la preuve de la cohérence (non-contradiction) de l'Arithmétique par des méthodes finitaires, donc internes à l'Arithmétique. Cette démonstration aurait alors garanti la cohérence des différentes géométries en dehors de leur sens comme organisation mathématique de l'espace physique. En 1904, Hilbert croit même avoir obtenu cette preuve, par induction sur la longueur de la preuve formelle – ce à quoi Poincaré répondra avec un grand éclat de rire : « M. Hilbert pense avoir démontré la cohérence de l'Arithmétique par induction, quand l'induction est l'axiome fondamental de l'Arithmétique » - voilà une belle circularité pour ceux qui ne pensent qu'à la cohérence interne des systèmes !

JL : Le droit a suivi une évolution parallèle : la théorie formelle du droit, et tout particulièrement le positivisme juridique de Kelsen, a tenté d'autonomiser le droit par rapport à toutes les pratiques sociales et de fonder sur sa seule norme interne la possibilité d'exercer le jugement et ce, dans les mêmes années que Hilbert et dans la même aire culturelle. De façon générale, le caractère formel du raisonnement juridique a été mis en exergue au détriment de toutes les ritualisations qui font pourtant le cœur de la légalité, ce qui fait sens pour les communautés humaines. Il y avait là un attachement à la norme juridique pensée comme un système de nature essentiellement textuel. L'attachement à la lettre de la loi finit aujourd'hui par transformer le droit lui-même quand le code n'est plus seulement pensé comme un recueil de lois (le « code civil » par exemple) mais qu'il devient encodé informatiquement : la légalité devient non plus formelle mais formaliste, seulement accessible par le biais des ordinateurs et non par le biais des textes de loi et de leur sens.

GL : C'est au cours des années 1930, que Herbrand, Gödel, Turing et bien d'autres spécifient avec rigueur ce que « formalisme axiomatique-déductif » veut dire. Ils définissent des systèmes logico-déductif totalement formalisés dans lesquels le sens est consciemment éliminé. Ligne après ligne, des règles d'écriture et de *réécriture* permettent de déduire à partir des axiomes, en tant que suites de signes alphabétiques, toutes les propriétés de la théorie formalisée : ce faisant, elles exécutent un

calcul. L'idée de procédure mécanique-scripturale se met en place avec parfaite rigueur, d'abord par la définition de la notion de fonction calculable, puis par la spécification d'une machine mathématique, celle de Turing. Elle deviendra le paradigme de l'ordinateur moderne, avec son départage très original logiciel/matériel, son compilateur et son système d'exploitation.

JL : Ainsi, à partir des années 1930, le nombre devient-il l'instrument fondamental du codage : il s'agit de réduire à des nombres et à des calculs sur des nombres toutes les propositions contenues dans ce qui jusqu'à présent étaient des textes (qu'ils soient géométriques ou, bien plus tard au cours du siècle, juridiques). Mais ce codage touche la notion de nombre elle-même : le nombre est lui-même réinterprété comme simple marque graphique choisie dans un répertoire de marques graphiques combinables, un alphabet. Ce principe alphabétique est potentiellement mécanisable parce qu'il est exécutable sans intervention humaine : il s'agit seulement de combiner des caractères graphiques selon des règles, des règles de *réécriture*, dit-on en informatique. L'arithmétisation utilise donc en son fond le principe d'une combinatoire alphabétique de nature entièrement formelle dans laquelle on est parvenu à distinguer radicalement le sens des caractères employés de leur simple forme sur la surface concrète qu'ils occupent.

GL : Cette histoire du codage arithmétique a été en effet fondamentale. C'est Gödel qui en a fait l'outil central de sa construction, en 1931. Pour répondre à la question de la cohérence posée par Hilbert, il encode les axiomes et les règles de déduction de n'importe quelle théorie formelle dans l'Arithmétique. Par ce biais, n'importe quel ensemble d'axiomes devient une (suite de) nombres, transformée par les règles de déduction dans d'autres nombres. La déduction logique est alors réduite à un calcul arithmétique. Puisque cela s'applique aussi à l'Arithmétique, un résultat inattendu devient possible : le codage, en Arithmétique, de la phrase « cette phrase n'est pas démontrable en Arithmétique », qui, si l'Arithmétique est cohérente, n'est pas démontrable. Voilà, une circularité splendide, référence explicite au paradoxe du menteur, qui donne une phrase indémontrable, c'est-à-dire dont on ne peut pas écrire la démonstration avec les ressources du formalisme alphabétique pertinent. De plus, et ce n'est pas évident, la cohérence de l'Arithmétique implique cette phrase, en Arithmétique, et devient alors aussi non démontrable par des méthodes alphabético-numérique : Poincaré est vengé ... par voie formelle ! Bien évidemment, pour atteindre cette prouesse, Gödel est obligé de spécifier ce que veut dire « fonction calculable » en Arithmétique. Grand succès de ces années, les définitions de fonction calculable ou algorithmique, fort différentes (Gödel, Church, Turing ...) sont toutes équivalentes. Voilà donc un invariant mathématique fondamental, la notion de fonction calculable sur les nombres entiers, donc

d'algorithme. Il ne dépend pas du système formel, c'est-à-dire du graphisme alphabétique-numérique utilisé, le software (logiciel) spécifique dirions-nous aujourd'hui, ni des dimensions de l'espace de calcul, ni du hardware (matériel) qui implémente ce logiciel. Cela donne une pleine généralité aux résultats négatifs donnés par ces trois auteurs - car, il ne faut pas l'oublier, ils inventent la calculabilité pour en démontrer les limites : des propositions indémonstrables, des fonctions non-calculables ! Church et Turing proposent alors une thèse selon laquelle n'importe quel système formel qui permet de définir des fonctions sur les entiers, n'importe quel logiciel, définit au plus ces fonctions calculables, voilà leur invariance maximale. Ils sont explicites : ils ne parlent que de formalismes logico-scripturaux, de type alpha-numérique – donc de toute marque graphique dans un répertoire codable par des nombres entiers. Mais au delà de cette thèse sur les formalismes arithmétiques, le code alpha-numérique devient porteur d'une ontologie mythique qui prend ce très bel invariant mathématique pour un *absolu*, de la nature, de la loi. Les algorithmes et les programmes informatiques deviendront intrinsèques à l'ADN, aux interactions physiques et aux processus sociaux. « La surprise, c'est que la spécificité génétique soit écrite, non avec des idéogrammes comme en chinois, mais avec un alphabet comme en français se félicite F. Jacob en 1965 ; grâce à cela, en biologie, « le concept de programme informatique donne maintenant un statut légal... à la téléonomie ... » (1970). Plus généralement, Wolfram imagine en 2013 que l'univers « operates like some behaviour of a Turing machine ». Et des algorithmes interactifs, les « écorithmes », constituent pour Valiant, en 2013, la dynamique par laquelle l'évolution, l'apprentissage animal (et humain) fonctionnent et ... fonctionnaient, avant l'invention des algorithmes à la Church-Turing : la nature ne fait qu'approximer, dit-il, notre invention maximale, les fonctions calculables par nos machines formelles, limite absolue de toute activité naturelle et humaine concevable.

JL : Commençons par cette prise de position métaphysique quant à la nature de la manipulation graphique des signes : elle est considérée comme ontologiquement fondée et il ne s'agit donc que de la retrouver. Vieux projet leibnizien : Louis Couturat, commentant dans sa *La logique de Leibniz*, le projet général de Leibniz en venait à le décrire de cette phrase devenue aujourd'hui prémonitoire : « La nature est le produit d'une logique divine, de ce calcul immense qui est la création ; elle est pour nous une admirable machine à calculer, car elle nous fournit, tout faits, les résultats de calculs qui dépassent la portée de notre entendement » (Paris, Alcan, 1901, p. 256). La thèse de Church-Turing n'est donc plus seulement une thèse sur la définition et l'extension du calcul formel, mais son périmètre s'accroît sans principe régulateur : elle devient, de façon totalement abusive, une thèse globale sur la nature physique et biologique et sur l'intelligibilité des rapports sociaux. C'est

particulièrement le cas dans le domaine du droit où la norme juridique contenue dans des recueils de textes de loi se trouve désormais mise en compétition avec une norme encodée arithmétiquement dans des logiciels visant une mise en ordre alphabétique. Or le droit est un facteur de production de normes sociales qui construit une histoire et non pas seulement l'application de règles d'écriture et de réécriture hors de tout contexte social. Il a cette capacité d'instituer des modalités dans l'intelligibilité des interactions sociales en distinguant le possible, le souhaitable, l'obligatoire et l'interdit, modalités qui demandent des interprétations. Autant dire que le divorce formaliste entre la combinatoire des marques alphabétiques et le sens de ces marques ne peut pas y avoir cours. La cybernétique avait à l'origine, dans les années 1940, essayé de penser la gouvernance au moyen d'une théorie du contrôle continu qui n'était pas seulement régie par des moyens calculables : du non-linéaire était présent dans les rétroactions assurant le contrôle et ce non-linéaire pouvait ne pas être approximable par du discret numérique.

GL : L'omniprésence des ordinateurs a finalement eu raison du projet cybernétique originel : le contrôle féroce, mais « lisse » (smooth) de la société, est devenu tout simplement le contrôle par le calculable, par la règle d'écriture formelle. L'évolution darwinienne et les sociétés humaines en viennent à être interprétées comme gouvernées par des algorithmes interactifs, appelés par Valiant « écorithmes », des règles d'écriture et de réécriture qui se modifient par friction réciproque, avec un but local, sans finalité globale. Ce but local a une caractéristique analogue à la maximisation du profit d'entreprise. La maximisation par voie algorithmique du « corporate profit » est régie par le même type de règles alphabétiques que l'adaptation évolutive, basée sur les quatre lettres de l'ADN : par un jeu de réécriture, les écorithmes optimisent localement la « fitness », explique Valiant en 2013 en formalisant algorithmiquement les thèses du néo-darwinisme.

JL : C'est donc le même principe d'intelligibilité qui s'applique à la nature et à la société, sans rupture : le calculable assure le lien entre les deux domaines. En particulier, les dynamiques attribuées à la nature valent aussi pour la société qui en vient elle aussi à fonctionner *via* des règles de réécriture. Ainsi la contrainte sociale est-elle dorénavant fondée en nature. Quels effets sur le droit ? La légalité juridique relevait jusqu'à présent d'une distinction principielle entre le fait et le droit. Le niveau du droit consistait en une idéalité qui ne devait en aucun cas être confondue avec le niveau des faits sous peine de mettre en danger la possibilité même du jugement : pour juger d'un fait, il était en effet indispensable de se doter d'un point de vue où ce fait aurait pu ne pas se produire et c'est ce qui permettait d'attribuer une liberté et une responsabilité aux sujets de droit (ils auraient pu agir autrement qu'ils ne l'ont fait). L'encodage informatique rompt avec l'idée d'une

légalité fondée sur la distinction entre le fait et le droit : c'est à même les faits, dans la nature calculable, que la légalité prend sa source car elle est seulement la réponse adaptative la meilleure à des dynamiques dont l'évolution définit le périmètre des contraintes. Cette idée rompt avec l'idéalité de la distinction droit / fait et avec la façon dont l'idée de loi est constituée dans le cadre de la science et instituée dans le cadre du droit. Elle a des conséquences importantes sur la façon de qualifier les faits puisque c'est en s'en tenant exclusivement à eux que l'idée de norme doit cependant faire surface. La norme juridique devient alors la régularité statistique de faits lestés d'un poids ontologique : la norme n'est plus modifiable sous l'effet de la délibération collective puisqu'elle est censée émaner de la nature des faits eux-mêmes. Cette norme s'exprime alors algorithmiquement puisque les algorithmes eux-mêmes ont été au préalable attribués à la nature.

GL : Ainsi, on en vient à ontologiser les algorithmes ou les écorithmes évolutifs, d'apprentissage etc. Turing, qui se pose déjà en 1948 le problème de l'apprentissage pour une machine, propose très modestement, en 1950, un jeu de *l'imitation* : par une interaction écrite, un humain doit reconnaître qui est une machine et qui est un humain. En 2000, dit-il, dans un jeu qui ne dure pas plus de 5 minutes, la machine aura 30% de chances de se faire passer pour l'humain. Aujourd'hui, après les progrès remarquables de la nouvelle intelligence artificielle, l'IA du Deep Learning, certains, Valiant entre autres, nous expliquent comment fonctionne l'apprentissage chez un enfant : il imite la machine ! Expliquons nous. La vieille IA est morte : le mythe de la machine logico-déductive qui ferait comme un humain n'existe plus. Des méthodes remarquables, qui ont leur origine en physique-mathématique, *catégorisent* images, sons, mots ... par filtrage linéaire et convolution non-linéaire sur plusieurs couches de réseaux de « neurones » (Deep). Sur la base de catégories d'image déjà faites (des millions d'images de chats, de voitures ... étiquetées par des prolétaires indiens ou chinois), voire par stabilisation d'invariants (des saillances) des images, la machine « reconnaît » un chat, une voiture ... elle leur met des étiquettes. Voilà, c'est ainsi que les animaux, nous compris, apprennent : ils mettent des étiquettes sur les choses et, dans notre cas, nos étiquettes sont des mots alphabétiques. Ainsi, d'excellents informaticiens, en ignorant des siècles de réflexions sur ce qu'est un animal, voire un enfant, nous disent comment ils apprennent (ils catégorisent comme les machines) et ce que c'est qu'un langage (des listes d'étiquettes). Je ne peux que rappeler les travaux récents de P. Violi sur le rôle de médiation sémiotique de la mère et la constitution d'un espace intersubjectif autant vécu affectivement et corporellement qu'intellectuellement. Cette dématérialisation et dé-historisation de l'homme, se base sur une vision d'un cerveau conçu comme hardware statique, machine *input-output*, avec une dynamique de type software (les réseaux sont « mathématiques »). Or, un cerveau animal n'est pas une machine *input-output* : il est super-actif dès

l'embryogenèse, chaotique si on le laisse sans les contraintes d'une interaction écosystémique. Cette dernière canalise l'activité cérébrale, avec la médiation d'un corps, des émotions partagées, sélectionne les connexions actives : le sens résulte d'une friction entre notre matérialité corporelle plastique et active, sans logiciel, et ce monde auquel on donne graduellement du sens. Même les déformations du cortex visuel primaire paraissent influencées par les émotions. Avec une certaine crainte le bébé s'approche d'un chat, le caresse pour la première fois et apprendra pour toujours ce que c'est un chat, non pas du tout une liste de traits mais une certaine qualité de douceur, d'indépendance, d'agilité transposable à des situations qui peuvent ne rien avoir de commun avec ce petit mammifère (« mon petit chat » pourra-t-on appeler l'enfant), ce qui est si typique du langage comme l'ont remarqué P. Cadiot et Y.-M. Visetti.

JL : Limiter le rôle du langage à étiqueter un réel déjà constitué par ailleurs passe aussi complètement à côté du caractère immédiatement collectif des langues : les signes linguistiques ne sont pas séparables en marques combinables sans signification d'une part et en sens susceptible de s'incarner dans n'importe quel support de l'autre. On doit plutôt concevoir les langues comme des constructions polysémiques de prégnances à tous les niveaux de constitution du sens. Il peut être méthodologiquement commode de distinguer différents niveaux de constitution du sens, de la prosodie à la syntaxe, mais y voir des modules préétablis comme dans un ordinateur participe de la projection du calcul sur la nature et occulte les processus à l'œuvre dans la construction du sens. C'est la souplesse des langues qui rend compte à la fois de leur résilience et de leur capacité à évoluer. De ce point de vue, les langues restent la meilleure façon de comprendre ce qu'est une interaction sociale changeante et l'écriture doit en être le témoin, pas le fossoyeur. C'est vrai également dans le droit où le niveau global du texte rend possible le niveau local de la décision qui n'est pas une simple déduction déterministe à partir de règles.

GL : J'ai beaucoup écrit sur la façon dont le déterminisme laplacien a été réintroduit en cognition et biologie par le biais du calculable - Turing et Schrödinger, qui fut le premier à faire usage de la notion de « code génétique », sont les premiers à le reconnaître, je ne peux que renvoyer à ma page web (quelle merveille d'en avoir une, quelle technologie formidable ! de plus, son accès est bien déterminé, exact... laplacien, sauf pour ce qui est du bruit). Je rappelle, à ce propos, la confusion entre aléatoire et bruit. Le premier est l'objet d'une analyse très fine en physique, classique et quantique, le deuxième est propre aux théories de l'élaboration (Turing) et de la transmission (Shannon) de l'information ; il est omniprésent en biologie moléculaire depuis Monod (« l'évolution est due à du bruit ») et suggère que la diversité du vivant, « mécanisme cartésien »,

dérive de qui est normalement considéré comme une nuisance. L'analyse de l'aléatoire introduit l'imprédictible en physique, même dans les théories déterministes classiques, Poincaré le montre dès la fin du XIX^e siècle. Boltzmann sur ce thème fait référence à Darwin qui l'avait considéré comme une composante essentielle de l'évolution, en tant qu'« extreme sensitivity » des organismes aux conditions de leur environnement. Je voudrais faire un parallèle entre cette forme de sortie de la détermination prédictible propre au monde de Descartes et Laplace et l'incommensurable, par lequel les Grecs nous ont appris à sortir du logos arithmétique. Toth et Chiurazzi récemment, et bien d'autres, ont mis en évidence le rôle dans la pensée grecque de la construction du premier nombre « irrationnel », cet « a-logos » comme dit Platon dans *La République*, ce nombre qu'on ne finit pas de calculer, rendu possible, comme pi, par une construction limite au cœur de la géométrie d'Euclide, fondée sur l'idée de ligne sans épaisseur. Voilà l'acte de naissance de la philosophie et des mathématiques, voire des sciences occidentales, consigné par Platon dans le *Théétète*, au moyen de constructions qui sortent du logos alphabétique, discret, celui du calculable arithmétique réifié dans les choses. Plus tard, l'invention de la perspective italienne avec son point projectif, limite infinie dans le tableau, forme symbolique du divin, ouvre la voie, comme le dit van Fraassen, au changement de « perspective » opéré par Copernic et aux espaces infinis de Descartes et Desargues. Tout cela est gommé par ces cadres du calculable arithmétique et du programme informatique intrinsèques à la nature, auquel on vient d'ajouter, inspiré par les réseaux informatiques, un peu d'interaction. On revient à un néo-pythagorisme, une ontologie du nombre entier et de l'alphabet déjà inscrit dans le monde, qui précède l'invention de notre science, sans même conduire à la splendeur de ce résultat négatif, que l'école de Pythagore elle-même avait su reconnaître comme limite de son propre savoir, lointain ancêtre des théorèmes négatifs de Gödel, Church et Turing.

JL : Il y a ici un paradoxe de l'écriture qu'il faut tenter d'explicitier du point de vue des pratiques collectives. Historiquement, la question de la diffusion de l'écriture est liée à celle de l'égalité devant la loi, l'isonomie grecque. Celle-ci était au cœur de la citoyenneté athénienne et relevait, comme les mathématiques, de l'idéalité : l'écriture et la lecture de la loi visait à trouver le moyen de produire un équilibre, une mesure entre les citoyens dont on pouvait espérer qu'il soit mathématique. Cet équilibre était atteint par la pratique de la délibération collective orale au sein d'institutions politiques qui introduisaient constamment de la diversité et de la variabilité dans l'égalité entre citoyens : les charges publiques étaient renouvelées périodiquement pour éviter autant que faire se peut la constitution d'oligarchies politiques ou financières. La délibération collective était au cœur de ce processus et reposait sur une élaboration partagée de la loi, la possibilité de son écriture et de sa lecture en commun. À l'époque moderne et contemporaine, les

délibérations démocratiques issues de l'égalité devant la loi ont progressivement imposé une alphabétisation généralisée : pour que nul ne soit censé ignorer la loi, encore faut-il qu'il soit capable de la lire. L'alphabétisation est donc une condition *sine qua non* de la citoyenneté démocratique. Or c'est cette alphabétisation rendant possible l'élaboration de la loi qui est remise en cause par la nouvelle forme d'écriture informatique et qui retourne en son contraire l'élaboration collective de la loi. Dans le cas du droit, le slogan de Lawrence Lessig « Code is Law » sous-entend qu'aujourd'hui la loi n'est plus lisible, ni par les citoyens ni même par les professionnels du droit. Accessible par mot-clé, selon des catégories pré-établies, elle ne fait plus partie des narrations développées à partir des textes légaux sans être contenues en eux : la loi informatisée donne une image fragmentée des conflits sociaux qu'il devient très difficile de se représenter dans leur globalité. Jusqu'à présent, il s'agissait d'en passer par le texte pour s'en affranchir et produire collectivement un jugement, par exemple lors de la délibération particulière qu'est le procès ; aujourd'hui, l'encodage informatique de la loi rend paradoxalement celle-ci illisible en érodant les conditions collectives de son intelligibilité. Ce faisant, on prend le risque, en déléguant à des logiciels les conditions de délibération de la loi et en s'en remettant à ces scribes modernes que sont les informaticiens, de renoncer aussi à l'autorité symbolique qui accorde à chacun l'exercice de la responsabilité.

GL : Nul ne doute de la puissance et de l'utilité de nos nouveaux instruments de calcul et de l'immense intérêt de la modélisation informatique en science, mais leur *projection ontologique* sur le monde a des conséquences sociales extrêmement graves. Venons-en donc aux raisons de ce que j'appellerais l'opportunisme dans cette vision computationnelle, en particulier concernant le vivant. On comprend les raisons de l'enthousiasme des pionniers qui, dans les années 1950 et 1960, ont inventé ces nouvelles sciences de l'information. Une vision totalitaire est toutefois déjà explicite chez de grands techniciens comme von Neumann et Wiener. Des jeux d'automates sur des cases écrites (les « automates cellulaires ») ou le contrôle automatique (par « feed-back/forward ») des dynamiques sociales et du vivant étaient censés rendre intelligible, voire gouverner, l'évolution et les organismes, la paix et la guerre, l'économie et les conflits. Simultanément, la nouvelle approche, moléculaire-alphabétique, de la biologie a immédiatement fait preuve d'une arrogance et un autoritarisme académique qui exclut toute pensée différente : Waddington, McClintock, Prusiner (par la découverte des prions, dans les années 1980, il avait déjà contredit le Dogme Central de la biologie moléculaire de 1958)... seront longtemps marginalisés, voire souffriront d'attaques personnelles. Plus récemment, la marginalisation de Kupiec, Sonigo, Montévil ... donnent des exemples des difficultés, voire du tabou portant sur une pensée différente de la matérialité en

biologie. Les plus anciens sont désormais réhabilités, pas les autres. Cela est possible car une nouvelle alliance entre ces visions computationnelles du monde, le sens commun et ... la finance s'est établie, très difficile à critiquer du fait du pouvoir sans partage qu'elle exerce. D'une part, tout le monde comprend ce que c'est qu'un programme informatique ou croit le comprendre ; c'est d'ailleurs facile : l'ADN en est un ! De l'autre, la finance guide désormais l'économie entière par des mathématiques largement implémentées sur des ordinateurs, voire par des programmes de gestion automatique des opérations boursières. Une nouvelle caste de scribes gère les affaires du monde au moyen d'outils inaccessibles même aux économistes qui ignorent tout de l'intégrale d'Ito pour les processus stochastiques et des programmes pour le « fast trading ». Alors, quand Ladyman et d'autres écrivent, en 2008, que « what cannot be computed cannot be thought », on passe de la vision misérable de la cognition humaine mentionnée plus haut à une *normativité* effrayante du calculable : il est interdit de penser ce qui ne peut pas être calculé par les machines construites ou programmées par les GAFAM. La pensée alpha-numérique du monde, le fait de projeter sur celui-ci le littéral de calculs sur les nombres entiers, justifie ultérieurement ces interdits : l'inerte, le vivant, la pensée et les rapports sociaux eux-mêmes seraient algorithmiques !

JL : Nous vivons une époque de *littéralisme de la loi*. Ce formalisme littéraliste a deux conséquences épistémologiques dont l'une ne va pas sans l'autre. D'une part, l'attention littéraliste qui se porte exclusivement sur la forme reconnaissable des lettres - « pattern-matching » susceptible d'être déléguée à des machines - tend à concevoir la perception sous la forme d'un apprentissage par induction à partir de cas *circonscripts comme des lettres*, cas pour toujours particuliers quels que soient leur nombre. D'autre part, méthodologiquement, cette attention littéraliste tend à ignorer la portée d'idéalité propre aux résultats négatifs qui permettent d'aborder les questions méta-théoriques du sens mathématique (complétude, consistance, décidabilité) d'un point de vue non-empirique. Ce littéralisme a transformé si profondément ce que nous entendons par réalité qu'on en oublierait presque qu'il exige de corroborer constamment son socle anthropologique, généralement non questionné : le divorce entre le niveau des marques alphabétiques et celui du sens. Turing l'avait bien vu, n'en déplaise à ceux qui l'ont lu trop vite, lui qui, dès 1950, avait montré dans le « jeu de l'imitation » que l'identification humain / machine était en elle-même indécidable puisque les conditions de sa réalisation empêchaient à tout jamais cette réalisation elle-même (il faudrait en effet, pour atteindre le résultat selon lequel humain et machine sont indistinguables, à la fois s'identifier et ne pas s'identifier à une machine). Et pourtant, vivre avec cette ambivalence est ce que l'on exige aujourd'hui massivement dans tous les métiers : on demande en effet de se comporter comme une machine (consistant à suivre des « protocoles » en médecine et des «

prédictions statistiques » en droit) en prenant de ce fait le risque d'être remplacé par elle. Il est facile de lire aujourd'hui les débats déjà anciens sur la possibilité de l'Intelligence Artificielle comme une tentative de hiérarchisation sociale dans laquelle plus les humains sont remplaçables par des machines, plus ils sont situés en bas de l'échelle sociale. Ou plus exactement, du fait que la maintenance physique des machines a un coût, celles-ci en viennent à occuper une position sociale intermédiaire entre le bas de l'échelle sociale où des humains sont astreints à se comporter comme des machines pour un prix inférieur à celui de leur maintenance et le haut de l'échelle sociale où d'autres humains déploient des activités non-mécaniques hors de prix. La réponse au stress collectif imposé par le divorce formaliste entre marques alphabétiques et sens exige de ce fait un débat de grande ampleur dans lequel le statut de la loi, qu'elle soit scientifique ou juridique, se révèle être une pièce stratégique.