Où aller en Intelligence Artificielle ?
Quelques réflexions sur le problème du bootstrap

Jean-Luc Dormoy
DER-EDF IMA-TIEM
1, avenue du Général de Gaulle
92141 Clamart Cedex
Jean-Luc.Dormoy@der.edf.fr

Résumé : *Ce papier expose quelques idées informelles sur le but, l'état et les voies de recherche en IA. Nous tentons en particulier de cerner les principes sous-jacents du phénomène de bootstrap. Nous évoquons divers domaines scientifiques, comme la logique, la théorie de l'évolution, et la biologie.*
Nous montrons comment la logique et son pendant moderne, le calcul, ne peuvent rendre compte des phénomènes de bootstrap. Nous discutons des différentes approches d'un "système qui se modifie lui-même" : l'apprentissage, le bootstrap, le méta-apprentissage, la découverte, et l'imitation d'un "enseignant". Nous cherchons dans la théorie de l'évolution et la biologie des pistes pour comprendre les mécanismes du bootstrap. Nous montrons en particulier comment ceux-ci peuvent être liés à la propriété d'autonomie d'un système. Nous discutons la notion, non formalisée à ce jour, d'autopoïèse. Nous montrons comment elle pourrait avantageusement se substituer à la sémantique du calcul, qui est l'arrêt, pour s'intégrer dans une certaine vision du bootstrap. Nous proposons une vision spéculative de la mémoire comme système autopoïétique en développement. Nous concluons par quelques remarques sur la vision largement répandue en IA selon laquelle l'esprit est gouverné par des mécanismes rationnels dans leur contenu, et nous tentons de cerner les "bons problèmes" sur lesquels nous pourrions tenter des expérimentations (par construction de systèmes) afin d'étudier ces mécanismes à découvrir.

1 Le but de l'IA et la méthode scientifique

L'Intelligence Artificielle est une discipline passionnante, et certainement une discipline de notre époque. On pourrait dire par un raccourci saisissant que depuis Galilée, nous ne sommes plus au centre du monde; que, depuis Darwin, nous ne sommes plus séparés du règne animal; en un sens, le but de l'Intelligence Artificielle est d'ôter à l'homme ce qui constitue la dernière raison par laquelle on pourrait le juger à part, hors de la nature : son esprit. Non pas que l'IA soit le point d'orgue d'une entreprise nihiliste, car lorsque l'IA aura été réalisée, il restera toujours à l'homme son humanité, dont sa liberté, entre autres celle de prendre son destin en main. Mais cela n'est pas la question que nous voulons aborder ici.

Il est donc communément admis que le but de l'IA est de créer une machine intelligente. Quoique des débats récents dans le bulletin de l'AFIA, et aussi les opinions émises lors de rencontres diverses divergent. Certains n'y voient qu'une nouvelle modélétique, d'autres ne jurent que par l'étude du cerveau humain, d'autres encore la font précéder d'une construction artificielle de la vie, jusques et y compris à invoquer des hypothèses que l'on peut qualifier de théologiques (l'hypothèse Gaïa).

Il ne s'agit pas ici de plonger dans la métaphysique, mais plutôt de fonder l'IA comme discipline scientifique, et d'affirmer certains principes de méthodologie scientifiques.

Il nous semble que l'IA repose sur la constatation qu'il existe dans la nature une sorte tout à fait particulière d'objet, que l'on nomme intelligence, cet objet étant digne d'étude scientifique. En fait, nous parlerons ici *d'esprit*, malgré les connotations de notre vocabulaire. En effet, non seulement les mots ne doivent pas nous faire peur, mais il nous semble que ce mot décrit encore mieux l'hypothèse sous-jacente à l'IA, et qui nous semble fondamentale. Cette hypothèse est que l'esprit est un objet en soi, régi par ses propres règles et lois, nécessaires à son existence et à son fonctionnement. Cela ne signifie pas que nous séparons l'esprit de la matière, auquel cas nous quitterions le terrain matérialiste sur lequel toute bonne science doit se fonder. Mais, même si l'esprit pour exister et fonctionner doit s'enraciner dans un substrat matériel, nous prétendons que ses lois et règles peuvent et même doivent être décrites indépendamment de celui-ci. Il s'agit ainsi pour l'IA de découvrir ces règles et lois, afin de recréer artificiellement un tel esprit via un substrat *ad hoc*.

Les liens entre esprit et matière sont évidemment dignes d'attention, ne serait-ce que parce que l'esprit doit être rendu possible modulo les lois du monde physique. Mais, pour prendre une analogie simple, nous pensons que la thermodynamique des gaz peut et doit dans une certaine mesure se faire dans l'ignorance des propriétés atomiques de la matière. Cela peut sembler paradoxal au vu des développements déjà anciens de la mécanique statistique. Aussi la mécanique statistique exprime-t-elle comment les lois de la thermodynamique s'enracinent et sont rendues possibles par les lois de la matière au niveau atomique, mais pas quelles sont ces lois. Une autre analogie est la situation comparée de la biologie et de la chimie. La biologie est envahie par la biochimie (biologie moléculaire), ce qui a permis des découvertes et des progrès fulgurants, mais elle ne se réduit pas à celle-ci. Soulignons malgré tout que cette vision des choses ne fait pas l'unanimité parmi les chercheurs en IA.

Maintenant, l'IA est plus que l'étude de l'esprit (tâche entamée sous un certain angle par la psychologie ou la psychanalyse). L'IA consiste à *construire* une machine intelligente. De science "explicative" des lois de la nature, elle paraît donc devenir une discipline d'ingénieur.

Nous entendons affirmer ici que ces deux points de vue ne sont pas contradictoires, en ce sens que l'étude des lois de la nature passe par leur *construction*. Cela, c'est l'histoire des sciences qui nous l'apprend.

Des milliers de pages ont été consacrés à la découverte scientifique comme *invention*. Les exemples fameux fourmillent. Ainsi, Kepler aurait voulu établir un

lien entre les 5 planètes du système solaire (on en connaisait 5 à son époque, plus la Terre) et les 5 polyèdres réguliers connus depuis l'antiquité. Son cheminement mental a donc été pour le moins détourné avant qu'il n'aboutisse à ses lois. Einstein et Infeld ont longuement insisté dans leurs écrits épistémologiques sur l'importance des *expériences de pensée* en physique. Cela ne signifie pas que l'expérience réelle ne joue pas de rôle en physique (ou dans les autres disciplines scientifiques), mais qu'elles sont précédées d'une réflexion théorique dont les expériences de pensée sont un outil majeur, nous y reviendrons. Ainsi, on peut souligner que l'œuvre magistrale de Galilée, les Discours, parlent beaucoup de billes roulant sur des plans inclinés. Ce sont des artifices de pensée que Galilée a utilisé pour étayer son argumentation scientifique en faveur de la loi de la chute des corps qu'il avait découverte, et il n'a à notre connaissance jamais fait d'expériences réelles de ce genre.

Un autre exemple est la découverte de l'oxygène par Lavoisier. Un autre chimiste avait peu avant lui isolé l'oxygène, mais il n'avait pas su se détacher de la vieille théorie du phlogistique. On dit communément que Lavoisier a *inventé* l'oxygène. Plus proche de nous, l'histoire de la structure en double hélice de l'ADN par Crick et Watson il y a 40 ans est presque devenue une légende. Ils ont expliqué *a posteriori* qu'ils cherchaient de l'or, alors que des laboratoires américains disposant de moyens bien plus importants étaient sur la piste. Alors ils se sont assis, et se sont demandés quelle pouvait bien être cette structure (modulo l'hypothèse que l'ADN renfermait le patrimoine héréditaire d'un individu, et qu'il devait donc être capable de se dupliquer sans faute). Après une série de propositions, l'hypothèse de la double hélice est venue.

On pourrait ajouter des exemples. Cela montre, malgré l'aspect schématique de cette présentation, que la découverte scientifique est le résultat de cheminements de pensée plus proches de l'invention que de la constatation après accumulation de données, vision positiviste que nous rejetons.

Les exemples précédents se référaient à des "sciences de la matière", dont les objets sont des objets matériels palpables et sensibles. Or, l'esprit et l'intelligence, s'ils sont des objets réels, n'ont certainement pas la même malléabilité matérielle. On peut les observer indirectement, on peut difficilement faire des expériences brutales avec eux, bref ils n'ont pas cette "malléabilité à l'expérience" d'une substance chimique ou d'une cellule vivante. Il nous paraît donc tout à fait justifié, si cette démarche a été couronnée de tant de succès - apparamment de manière paradoxale - dans les "sciences de la matière",de la reprendre à notre compte dans l'étude de l'esprit.

Autrement dit, l'étude de l'esprit doit comporter une part importante de "devinette". Mais il est clair que, ne jouant pas à construire des théories coupées de la réalité, nous devons à chaque étape monter des expériences pour tester nos hypothèses, c'est-à-dire les confirmer ou les réfuter. Ces expériences sont tout simplement les systèmes d'IA que nous construisons.

En résumé, notre objet d'étude est l'intelligence, ou l'esprit. Une démarche scientifique saine passe par la proposition d'hypothèses, et par la confrontation de ces hypothèses à la réalité par l'expérimentation, soit pour nous par la construction de systèmes d'IA.

La conséquence de tout cela est que, ce faisant, et comme dans toute discipline scientifique, *nous étudions des lois de la nature*. Un système intelligent, qu'il soit naturel ou artificiel, exhibe des lois naturelles, à savoir celles que nous recherchons, les lois de l'intelligence ou de l'esprit.

En corollaire, nous ne pensons bien évidemment pas que l'intelligence ou l'esprit humains sont les seules formes possibles d'intelligence; c'en sont des instances, et nous devons découvrir la classe.

2 Qu'est-ce qui caractérise une discipline scientifique ?

Nous avons déjà évoqué l'histoire des sciences, et nous allons continuer à le faire. Pour des épistémologues dont nous inclinons à partager le point de vue, l'histoire des sciences est faite d'une succession de périodes de "progrès calme", où est poussée, embellie, assimilée une théorie dominante - appelée aussi paradigme- , entrecoupées de brusques changements révolutionnaires, où la théorie dominante est bouleversée et finalement balayée au profit d'une nouvelle théorie qui devient alors le nouveau paradigme. Ces révolutions ne sont pas gratuites, elles surviennent lorsque la vieille théorie craque et ne peut plus résister sous les coups de contradictions ou de réfutations avec l'expérience. L'histoire de la physique fournit un exemple évident de ces processus.

Une discipline scientifique est donc nécessairement basée sur un paradigme dominant. On peut donc se poser la question : quel est le paradigme dominant de l'IA ?

Nous allons essayer de montrer que le paradigme dominant de l'IA actuelle est la *logique*. Cela ne plaît pas nécessairement, et nous devons avouer que c'est à notre corps défendant que nous avons dû l'admettre.

Pour cette démonstration, il faut remonter le cours du temps, et là aussi faire un peu d'histoire. Nous débuterons avec les débuts de l'ordinateur (nous ne remontons pas au Golem, ou à d'autres manifestations historiques de l'idée de l'IA, car nous désirons nous limiter à l'IA comme entreprise scientifique systématique).

Il nous semble en effet important de dater les débuts de l'IA de la naissance de l'ordinateur, pour deux raisons; tout d'abord, il ne saurait y avoir d'IA au sens moderne sans ordinateur; deuxièmement, dès que la possibilité de construire un ordinateur a été conçue, les acteurs scientifiques de l'époque ont immédiatement vu le profit que l'on pouvait en tirer. On connaît les papiers de Turing (1948); à notre connaissance, les premiers papiers d'IA datent de 1940, en Angleterre.

On le sait, l'ordinateur est la matérialisation d'une machine théorique, la machine de Turing (1937). L'histoire commence en 1931, lorsque, pour des problèmes de fondements des mathématiques qui n'ont encore rien à voir avec les ordinateurs, Gödel démontre ses fameux deux théorèmes. Leur démonstration contenait déjà une première définition de ce qu'est une *fonction récursive*, mais sa définition était trop limitée (elle correspondait à ce qu'on appelle aujourd'hui les fonctions récursives primitives). L'attention étant donc focalisée sur une définition de la *calculabilité*, Gödel propose en 1934 une définition générale de la récursivité dans les entiers

naturels, et Church peaufine son lambda-calcul entre 1931 et 1936. Commence à être clairement entrevue la possibilité d'une définition mécaniste de la calculabilité. C'est celle que Turing va proposer en 1937, avec sa fameuse machine. Mentionnons que Post donnera également une autre définition mécaniste, basée sur la métaphore du travailleur taylorien. Turing démontrant l'équivalence de ces trois formes de calculabilité, la porte était ouverte au calcul mécanique.

Cependant, la machine de Turing va plus loin que cela. Il existe, parmi toutes les machines de Turing possibles, une classe particulière constituée de ce que l'on appelle des machines universelles. Les machines de Turing ont en effet cette particularité qu'elles peuvent être codées pour être introduites comme données d'une autre machine de Turing. Une machine universelle est donc une machine qui, si on lui a fourni le code de n'importe quelle autre machine, agira exactement comme si elle était cette machine. Le calcul va donc plus loin que 2+2=4, il permet de réaliser l'essence même du calcul, dont les machines universelles sont des matérialisations.

Il est bien connu qu'une machine universelle, c'est un calculateur programmable, c'est-à-dire un ordinateur.

Dès la guerre, des projets (Royaume-Uni, Etats-Unis) voient le jour pour construire une machine universelle. Nous laissons de côté l'histoire de l'informatique à partir de cette date. Notons seulement pour la petite histoire que Turing lui-même eut bien du mal à se faire entendre, les "décideurs" de son époque prétendant plus profitable de construire une machine dévolue au calcul scientifique, et non une machine universelle.

A partir de là, les premiers travaux pionniers sur l'IA commencèrent. Ainsi, le célèbre papier de Turing *Mechanical Intelligence* suggérait-il de construire des programmes jouant aux échecs et faisant de la cryptarithmétique. Turing avait avant 1947 construit un programme d'échecs sur le papier, n'ayant pas encore de machine pour le faire tourner, programme combinatoire proche dans son esprit des programmes d'aujourd'hui. Il faut d'ailleurs lire ce papier pour voir comment Turing rajoute des epsilons à la fonction d'évaluation pour obtenir le résultat désiré ! Ce grand scientifique a décidément inventé beaucoup de choses, y compris la bidouille informatique.

Mais Turing ne limitait pas les possibilités de l'intelligence mécanique aux échecs ou à la cryptarithmétique; il évoquait aussi la possibilité de construire des robots, en écartant provisoirement leur réalisation concrète pour des questions matérielles, notamment de capteurs. Il insistait enfin sur la possibilité de construire des programmes qui *apprennent*.

A la suite de Turing, et trois ans après sa mort, l'Intelligence Artificielle est "née une seconde fois" en 1957 avec la célèbre conférence réunissant Simon, Newell, McCarthy et Minsky entre autres. L'entreprise scientifique devenait systématique.

Mais il faut bien souligner comment elle est née. Le fait qu'un même homme - Turing - ait inventé l'ordinateur et l'Intelligence Artificielle constitue une marque de naissance qui est restée indélébile.

Il faut aussi souligner qu'à la base est le calcul, dont les découvertes des années 30 montrèrent qu'il peut s'exprimer sous des formes diverses. Pour résumer de manière schématique calcul = fonctions récursives (générales) = machine de Turing. Bref, calcul = logique.

Ces faits sont bien connus, mais comment ne pas y voir le fil directeur qui nous fait aujourd'hui encore écrire des *règles d'inférence*, par exemple, même et y compris si l'on désire s'éloigner de l'inspiration logique ?

D'ailleurs, les multiples travaux de ces dernières années sur les logiques non classiques qui ont envahi les revues et conférences d'Intelligence Artificielle sont en un sens dans la lignée directe de l'IA des débuts : ils poussent le paradigme jusqu'au bout. On peut aussi mentionner la programmation logique. Ou la planification. Ou la plupart des formes d'apprentissage (y compris connectionnistes, nous ne pouvons argumenter sur ce point).

Aussi ne s'agit-il pas de démontrer que le paradigme dominant de l'IA d'aujourd'hui est la logique en faisant une sorte de "sondage" dans la "communauté IA", ou de "vote majoritaire" de ses publications - la minorité devant bientôt cesser, espérons-le, d'être silencieuse -, mais de comprendre les racines des tendances actuelles, y compris celles que l'on peut juger perverses.

On peut enfin opposer que la "représentation des connaissances", la notion même de connaissance, relèvent de conceptions distinctes. Sans doute les choses ne sont-elles pas chimiquement pures. Encore que, si l'on regarde les compte-rendus de la conférence internationale *Knowledge Representation*, on y verra surtout … de la logique. Mais il est clair que bien des chercheurs et des "écoles" cherchent, justement, à se séparer de la logique comme source d'inspiration. Cela est à notre sens la conséquence de la situation actuelle de l'IA, à savoir que *l'IA est en crise*.

3 Le péché originel : "Une erreur philosophique de Turing"

La situation de l'IA est un peu paradoxale. Si elle démontre bien les caractéristiques d'une discipline scientifique, à savoir l'existence d'un paradigme dominant, celui-ci est marqué d'un péché originel, justement à cause de la confusion des origines de l'IA et de l'informatique. On peut dire que la logique n'est qu'un *paradigme d'emprunt* pour l'IA.

Mais pour tenter de discerner quelles nouvelles directions de recherche pourraient être envisagées, nous allons considérer avec quelque détail ce dont nous disposons, à savoir la machine de Turing. Nous allons aussi voir comment Turing lui-même fournissait une justification humaine de l'architecture de sa machine. En fait, la machine de Turing est le premier modèle cognitif au sens où nous l'entendons lorsqu'on vient de l'IA, celui d'un homme accomplissant un calcul. On verra d'ailleurs que cette justification est issue d'une méthode d'investigation scientifique tout à fait similaire aux expériences de pensée des physiciens. Nous verrons ensuite comment un des acteurs majeurs de la logique et des débuts de l'informatique, Gödel lui-même, critiquait la justification de Turing. A notre sens, l'argumentation de Gödel remet en cause la thèse de Church, mais surtout envisage une perspective de recherche qui nous est familière.

Une machine de Turing est constituée de trois parties : un ruban, une tête de lecture-écriture, et une mémoire fixe. Le ruban est linéaire, infini dans au moins une direction, et constitué de cases pouvant contenir chacune un caractère d'un alphabet fini donné une fois pour toutes, soit A. La tête est toujours positionnée sur une case du ruban, et à chaque étape de calcul, elle lit le caractère c de la case qu'elle "pointe", écrit éventuellement à sa place un autre caractère c', et effectue un mouvement d=±1 d'une case vers la gauche ou vers la droite. Pour déterminer tous ces "éventuellement", un ensemble de transitions est contenu dans la mémoire, qui ont toutes la forme

q, c --> q', c', d

Les "q" sont les états de la machine, et forment un ensemble *fini* Q. La machine est dans un unique état à la fois. Ainsi, si la machine est dans l'état q, que la tête lit le caractère c, alors la machine passe à l'état q', la tête écrit le caractère c', et effectue le mouvement d. Puis la même série d'opération recommence, jusqu'à éventuellement ce qu'aucune transition ne soit plus applicable, auquel cas la machine s'arrête. Dans ce cas, on peut "lire" le résultat du calcul qu'elle a effectué sur le ruban.

Il peut y avoir plusieurs transitions commençant par un même couple (q,c). Dans ce cas, on dit que la machine est non-déterministe, car elle aura à choisir la transition à appliquer (selon des critères non spécifiés). Dans le cas contraire, la machine est dite déterministe. C'est un des résultats de Turing qu'une machine *non-déterministe* peut toujours voir son fonctionnement simulé par une machine *déterministe*. Ainsi, dans le calcul mécanique, le choix n'est qu'une fiction (sur un plan formel, en particulier tant que l'on ne parle pas d'efficacité en temps).

Les propriétés mathématiques de ces machines sont évidemment essentielles pour démontrer l'équivalence avec les autres formes de calcul. Mais ce qui nous intéresse plus est la justification de cette machine comme *modèle cognitif* de l'homme engagé dans une activité de calcul. Nous la reprenons ici de manière libre, mais l'idée y est.

Tout d'abord, on considère qu'un homme est constitué d'organes de perception et de fonctions cérébrales lui permettant de posséder des états mentaux, ainsi que de possibilités de mouvement et d'action sur le monde. Il est clair qu'à la fois ses organes de perception et sa capacité mentale sont limitées. On doit donc faire l'hypothèse que seul un nombre fini de "perceptions" ou "d'états mentaux" sont possibles. Quant aux mouvements qu'il peut faire, on se doit de considérer de la même manière qu'il ne pourront le mettre que dans un nombre de situations finies. On va donc modéliser le "monde" par le ruban et ses cases, les percepts possibles par l'alphabet A, une case donnée étant occupée par le "caractère" c si l'individu voit c dans la position de la case, la position de l'individu dans le monde par la position de la tête de lecture-écriture (et donc sa perception par la lecture via la tête, et son action par l'écriture), et enfin ses états mentaux par Q. Il est clair que A et Q peuvent être énormes, mais ce qui importe d'un point de vue mathématique est qu'ils soient finis. Maintenant, les actions de l'homme en fonction de ce qu'il perçoit doivent être définies par des règles, qui sont nos transitions. Si plusieurs décisions sont possibles pour un même état mental et une même perception, on mettra plusieurs règles, mais elles seront en nombre fini, là aussi , car la mémoire de l'homme doit avoir une capacité finie. Enfin, nous parlons de décisions raisonnables, et donc pas par exemple aléatoires, ce qui justifie l'existence de règles. S'il le faut, nous "découpons"

états et caractères en "granules" plus petits afin que de telles règles soient envisageables. L'essentiel, là aussi, est l'hypothèse de finitude du nombre de transitions.

On peut immédiatement soulever plusieurs objections. Ainsi, le ruban est mono-dimensionnel. Mais on démontre mathématiquement qu'une machine de Turing ayant un ruban multi-dimensionnel - avec un nombre de dimensions fini - a toujours une machine équivalente mono-dimensionnelle. On peut aussi réduire le nombre de caractères de A - jusqu'à n'en avoir que 2 - au prix d'un accroissement du nombre d'états et de transitions. On a affaire à une sorte de vases communicants entre les constituants de la machine - mais la somme ne doit pas dépasser les capacités humaines, et donc être de taille finie.

Bref cette métaphore s'applique à plus que "l'homme au calcul", c'est le fonctionnement tout entier de l'intellect de l'homme qui est ainsi modélisé. On aboutit donc à la question, qui a fait coulé beaucoup d'encre, notamment ces derniers temps : Peut-on considérer que la machine de Turing représente un modèle cognitif humain ? De manière plus concrète, le fonctionnement d'un esprit humain a-t-il un équivalent algorithmique, c'est-à-dire un programme, c'est-à-dire une machine de Turing particulière ?

Nous ne nous lancerons pas dans une critique des arguments avancés, notamment ceux basés sur les théorèmes de limitation adaptés de celui de Gödel. Nous citerons plutôt Gödel lui-même :

> "***A philosophical error in Turing's work.*** *Turing, in his 1937, page 250, gives an argument which is supposed to show that mental procedures cannot go beyond mechanical procedures. However, this argument is inconclusive. What Turing disregards completely is the fact that mind, in its use, is not static, but constantly developing, i.e. that we understand abstract terms more and more precisely as we go on using them, and that more and more abstract terms enter the sphere of our understanding. There may exist systematic methods of actualizing this development, which could form part of the procedure. Therefore, although at each stage the number and precision of the abstract terms at our disposal may be finite, both (and therefore, also Turing's number of distinguishable states of mind) may converge toward infinity in the course of the application of the procedure. Note that something like this indeed seems to happen in the process of forming stronger and stronger axioms of infinity in set theory. This process, however, today is far from being sufficiently understood to form a well-defined procedure. It must be admitted that the construction of a well-defined procedure which could actually be carried out (and would yield a non-recursive number-theoretic function) would require a substantial advance in our understanding of the basic concepts of mathematics. Another example illustrating the situation is the process of systematically constructing, by their distinguished sequences* a_n *--> a, all recursive ordinals of the second number-class."*

Outre le titre (il faut être Gödel pour se permettre d'écrire ça), ce que Gödel critique est que l'esprit, selon Turing, ne peut posséder qu'un nombre fini d'états distinguables. Après avoir envisagé (passage non cité) qu'après tout nous pourrions

en avoir un nombre infini, il admet que l'hypothèse de finitude semble assez réaliste. Aussi est-ce sur un autre point qu'il fait porter sa critique : même si, à un instant donné, notre esprit n'a qu'un nombre fini d'états mentaux (potentiels), de nouveaux états mentaux peuvent être "introduits" par "une procédure bien définie" mais que nous ne connaissons pas encore. Ainsi, une telle procédure serait un "mécanisme non mécaniste" (puisque, dans le vocabulaire des logiciens spécialistes de calculabilité, mécaniste est devenu synonyme de calculable, cf. la thèse de Church). En définitive, le nombre d'états mentaux potentiels pourrait tendre vers l'infini. Gödel souligne que c'est ce qui à son avis se produit lorsque l'on appréhende de nouveaux axiomes (par exemple la théorie des ensembles par opposition aux axiomes de l'arithmétique, ou l'ajout de l'axiome du choix). C'est pour lui une preuve "historico-mathématique" que le nombre d'états mentaux de ceux qui ont assimilé les nouveaux axiomes, ou de ceux-ci par rapport à leurs prédécesseurs, a augmenté. Pour Gödel, la connaissance accumulée par les générations précédentes et acquise par les nouvelles accroissent le nombre potentiel d'états mentaux de ces dernières par rapport à leurs ancêtres. Mis dans nos propres termes, le "bootstrap social" de l'accumulation de connaissances est une preuve du "bootstrap mental" de ses acteurs.

Il est clair que c'est à la fois d'apprentissage et de bootstrap dont il est question.

4 Bootstrap, apprentissage, découverte, imitation

Nous reviendrons sur les machines de Turing ensuite. Nous nous occupons ici de discuter différents types de systèmes "qui se modifient eux-mêmes".

En IA et dans d'autres disciplines, les notions de bootstrap, d'apprentissage et de découverte ont été utilisées. Sont-elles pour autant bien définies ? Sont-elles distinctes ? Si oui, en quoi ?

Les réponses à ces questions ne nous apparaissent pas clairement à ce jour. On pourrait dire que le bootstrap est un processus selon lequel un système se transforme pour devenir meilleur. Mais que signifie "meilleur" ? Cette définition peut être poussée pour devenir "récursive vers le futur". Nous entendons par là qu'un système subissant un bootstrap avec succès doit non seulement devenir "meilleur" en un sens fonctionnel, c'est-à-dire dans son rapport au monde, mais doit aussi contenir des possibilités de bootstrap supplémentaires. Donc, le succès de l'étape n de bootstrap, passant de l'état e_{n-1} du système à l'état e_n, est définie par son succès à l'étape n+1 :

$$\text{Succès}(n,e_n) = F(\text{Succès}(n+1,e_{n+1}))$$

L'apprentissage est un processus par lequel le système augmente ses capacités ou ses performances. Dans la plupart des travaux en IA, la partie du système qui assure l'apprentissage est donnée au système, et reste inchangée. D'ailleurs, le "coût" de conception de l'apprentissage dépasse en général largement le "gain" observé, c'est-à-dire qu'il aurait été de façon pragmatique beaucoup plus simple de fournir au système ce que l'apprentissage a construit que de fournir l'apprentissage lui-même. On peut espérer que cette situation s'inverse si l'apprentissage est "indépendant du domaine", et s'il est appliqué à beaucoup de domaines. Il n'empêche que, si les "connaissances" trouvées par apprentissage deviennent alors *en quantité* importantes, elles sont *conceptuellement* plus simples que celles qui assurent l'apprentissage.

On voit donc deux différences entre l'apprentissage et le bootstrap. Tout d'abord, "l'essence" du système, qui est d'apprendre, n'est pas modifiée. Il n'y a donc pas bootstrap de cette partie du système. Deuxièmement, les capacités d'amélioration du système sont épuisées dès la première étape : des choses sont apprises, mais une fois apprises, et à moins de changer de domaine, il n'y a plus aucune amélioration à espérer.

On parle aussi de méta-apprentissage. L'idée est simple est élégante : il s'agit d'appliquer un système d'apprentissage à lui-même pour qu'il s'apprenne. On a le sentiment d'avoir ici une piste pour réaliser un système de bootstrap. Au-delà de cette idée simple et exploitée systématiquement selon laquelle il est bon, si cela est possible, qu'un système s'applique à lui-même, la notion de méta-apprentissage n'est pas non plus claire. Tout d'abord, ce qui est "méta-appris" dépend de ce qui est appris. Si par exemple l'apprentissage consiste à accélérer l'utilisation du système sur lequel il agit, alors le résultat du méta-apprentissage va être d'accélérer le processus d'apprentissage. Mais le *contenu* de l'apprentissage n'est pas changé pour autant. De plus, si on applique cet apprentissage accéléré au niveau du méta-apprentissage, on aura un méta-apprentissage accéléré, mais en définitive on va simplement obtenir un système d'apprentissage accéléré, et dont l'utilisation au niveau méta va vite s'épuiser. On a donc ici un "bootstrap", mais sur une ou deux étapes, ce qui ne correspond pas au pattern ci-dessus, où après chaque étape le système a encore la capacité de subir un bootstrap.

Il est donc nécessaire, pour avoir un méta-apprentissage qui se rapproche d'un bootstrap, que l'apprentissage modifie qualitativement le système sur lequel il agit, qu'il lui fournisse des capacités qu'il n'avait pas auparavant, disons, dans notre langage habituel, qu'il construise de nouvelles connaissances. On peut alors en principe imaginer un phénomène de bootstrap réel.

Nous avons dit "construire de nouvelles connaissances". En général, cela est interprété comme "découvrir de nouvelles connaissances". Nous pensons là aussi que cette notion est assez mystérieuse. La vision communément admise de la découverte est que le système trouve *en son sein* les ressources pour acquérir des connaissances qui n'y étaient pas. Par ailleurs, les métaphores, ainsi que les essais tentés en IA, ont souvent à voir avec la découverte scientifique : découverte de théorèmes mathématiques, de lois physiques, de bonnes stratégies de jeu, etc. On a donc dans l'idée ce qui est vu comme l'activité la plus noble et la plus prestigieuse de l'homme : le génie inventif, la créativité. On peut aussi considérer que la découverte est l'essence d'un processus de bootstrap réussi.

Ces particularités attribuées à l'homme sont sans aucun doute très intéressantes, mais elles restent apparamment l'apanage d'un nombre restreint d'individus, et surtout se produisent peu fréquemment. On peut ainsi se demander combien de fois un être humain "normal" fait de découvertes dans sa vie, y compris s'il s'agit par exemple d'un scientifique ou d'un musicien. Mais on a plus le *sentiment* de faire ou d'assister à une découverte que la capacité de caractériser ce qu'est une découverte. Tout cela semble donc difficile à définir.

Par contre, nous subissons dans la vie courante un processus beaucoup plus systématique, qui est celui de l'enseignement. Cela se produit à l'école, mais aussi avant et après (le bébé qui apprend à marcher ou à parler, lire un livre). On a alors affaire à un acteur plus ou moins conscient de son rôle, “l’enseignant”, qui souligne les points intéressants à apprendre, fournit des méthodes et des exemples de résolution de problèmes, et des exercices. “L’élève” a donc un "modèle" à imiter. La différence de degré avec la "découverte" est donc énorme, et on pourrait ici parler en fait *d'imitation*. On peut par exemple considérer que le processus "d'apprentissage" consiste pour l’élève à déterminer et mettre au point les connaissances que peut bien utiliser l'acteur-enseignant. Evidemment, l'imitation repose sur une sorte de processus *culturel*. Elle ne rend pas compte, sinon par "coup de chance", du processus par lequel l'enseigné pourrait devenir "meilleur que le maître". Mais on peut considérer ce processus comme un bootstrap "borné", qui conduit progressivement l'élève de "rien ou presque", au niveau du maître. D'ailleurs, il y a dans tous les systèmes d'apprentissage en IA une partie d'imitation, puisqu'on leur donne les "bons cas" sur lesquels travailler. Mais la méthode d'apprentissage n'est pas imitative.

Là non plus, les choses ne nous paraissent pas très claires, mais il nous semble qu'il y a une grande différence de degré entre découverte et imitation. Au moins l'imitation au sens où nous l'entendons est-elle beaucoup plus fréquente.

5 Bootstrap ou bootstraps ?

Le problème du bootstrap a été posé par M. Pitrat dès 1984 [Pitrat 84], et une tentative de mise en application de ces idées a été réalisée dans Maciste.

Il nous semble que ces travaux reflètent un certain point de vue sur le bootstrap. Il est ici conçu essentiellement comme un *moyen de conception* d’un système d’Intelligence Artificielle. Entre chaque étape de bootstrap, c’est le concepteur qui intervient, l’avantage de la méthode étant qu’il dispose déjà d’un système pouvant l’aider à passer cette nouvelle étape. C’est “l’IA qui aide l’IA”.

Si cette idée nous paraît de toute évidence d’une grande richesse, il nous semble qu’elle n’a pas été “poussée jusqu’au bout”.

Une des conséquences intéressantes de la conception par bootstrap est que le concepteur doit très vite abandonner l’idée que le but du système qu’il construit est extérieur à ce système. Ainsi, le concepteur pourrait vouloir construire un système qui ait une certaine fonction, par exemple d’être le compilateur d’un certain langage informatique. Mais s’il veut y parvenir par bootstrap, le système devra s’appliquer à lui-même, ce qui contraint énormément son contenu par un double effet : sa “forme” ne doit pas être trop compliquée pour pouvoir être traitée par un système (lui-même) au contenu encore “faible”; son contenu doit préférentiellement traiter des problèmes dus à sa forme. En définitive, le concepteur est réduit à se désintéresser de son but initial pour ne plus avoir qu’une préoccupation “abstraite” : le système. *Dans le bootstrap, le système ne sert qu’à lui-même*.

En pratique, c’est non seulement le système, mais aussi les interactions entre le système et son concepteur qui doivent faire objet de développement. Ainsi, les

problèmes d'édition et de debugging ont-ils été prééminents dans Maciste. Cela donne au concepteur le sentiment d'une fuite en avant, d'un manque de maîtrise du système et de sa construction, et d'agir sans plan et au jour le jour.

De ce point de vue, un système issu de bootstrap a une fonction générique: il existe, tout simplement.

En définitive, l'utilisation du bootstrap comme méthode de conception change complètement la fonction du système : de particulière, elle devient générique. Dans le même temps, le système échappe à son concepteur, pour devenir "autonome", du moins dans son intention.

Cette propriété, vue à l'origine comme un effet secondaire de ces travaux sur le bootstrap, pourrait devenir définitoire. Dans le même temps, cela permet d'envisager d'autres types de bootstrap.

La nature fournit des exemples de deux types de bootstraps distincts : l'évolution et l'ontogénèse[1]. Ces types de bootstraps se distinguent entre eux par leur dynamique. L'évolution se déroule sur des échelles de temps géologiques, l'ontogenèse sur une échelle comparable à la durée de vie d'un organisme.

Mais ce n'est pas la seule différence. L'évolution est un processus de changements d'individus (de "systèmes") "stables", c'est-à-dire ne subissant pas de changement évolutif. De plus, l'évolution a pour facteur l'hérédité des caractères, dont nous savons aujourd'hui qu'elle est portée par les gènes. D'un autre côté, l'évolution agit sur le phénotype de l'individu, c'est-à-dire sur la manière dont les gènes ont été exprimés (il faudrait aussi éventuellement inclure des facteurs extra-génétiques). Enfin, l'évolution n'est pas "dirigée" par un concepteur.

L'ontogenèse semble par contre un type de bootstrap beaucoup plus "programmé à l'avance". Des théories du développement embryonnaire ont été formulées, notamment la topobiologie d'Edelman. Dans cette théorie, l'ensemble du programme est transmis aux cellules se divisant, mais une partie seulement s'exprime en fonction du lieu ou se trouve cette cellule par rapport aux autres. Ce sont des "messages" de ses voisines qui déclenchent ou inhibent tel ou tel processus. De plus, les cellules se déplacent, ces déplacements étant controllés par des substances elles-mêmes produites ou non-produites par des cellules voisines. Cependant, la "programmation" n'est pas complète, des facteurs épigénétiques rentrant en ligne de compte, comme l'a montré Edelman. Ceux-ci sont nécessaires au bon fonctionnement du "programme", mais ne sont qu'approximativement déterminés. On pourrait donc parler de "programmation statistiquement correcte".

Malgré leurs différences, ces trois phénomènes de bootstrap ont un point en commun : le système existe de manière relativement stable entre chaque étape de bootstrap.

Or, on peut penser à un type de bootstrap très différent, une sorte de *bootstrap permanent*. Ainsi, un individu pensant n'arrête pas d'interagir avec le monde, et ce faisant de "se modifier". Des processus "purement internes", et "d'intégration des

[1] C'est-à-dire la croissance de l'œuf fécondé à l'individu adulte.

éléments extérieurs" se recouvrent donc, ont lieu dans le même temps. Il est par exemple bien connu qu'un système d'apprentissage gagne à apprendre de la résolution d'un problème alors même qu'il est en train de le résoudre : il peut utiliser des connaissances apprises dans une première étape pour progresser dans une seconde.

Evidemment, les deux dynamiques (modification du système, activité propre au système sans modification) se recouvrant, les choses sont encore moins claires que lorsque les deux dynamiques sont séparées. Néanmoins, il nous semble évident que l'esprit réalise un tel phénomène de bootstrap permanent. Nous suggérons dans la suite une application de cette idée au fonctionnement de la mémoire.

6 Le monde extérieur, les ordinateurs et leur complexité

Un autre aspect dans l'analyse comparée du bootstrap, de l'apprentissage, etc, est la position du système vis-à-vis du monde extérieur. Dans l'apprentissage, les exemples susceptibles de traitement et fournis au système le sont en relation avec le monde extérieur. La découverte semble par contre reposer sur des ressources *internes* de créativité ou d'inventivité, son "génie" est en quelque sorte intrinsèque. Par ailleurs, le système a affaire à un monde extérieur "brut", en tout cas non organisé du point de vue de ce qu'il doit découvrir. A l'inverse, dans l'imitation, on a un monde extérieur, mais déjà "organisé" d'un point de vue "cognitif".

Il nous semble évident qu'il faut *toujours* un monde extérieur. Celui-ci peut être un monde de "bureaux et de pièces" comme pour beaucoup de robots actuels, ou un ensemble d'exercices de tactique aux échecs, peu importe[2]. Mais il nous semble impossible qu'un système évolue grâce à des ressources uniquement intrinsèques.

Maintenant, comment "concevoir" ce monde ? En général, on suppose que le système dispose de *capteurs* qui lui fournissent des *données* sur le monde : système = programme, monde = données. Cette vision nous semble incorrecte si l'on vise le bootstrap. En effet, il faut bien que le système se modifie dans ses parties les plus essentielles, et comment va-t-il le faire si cela doit être effectué par des "instructions" qu'il contient déjà ? Nous utilisons ici l'argument selon lequel un programme ne contient pas plus que ce qu'on y a mis.

Une version formelle de cette affirmation existe, c'est la notion de *complexité d'information algorithmique* de Chaïtin. Chaïtin a repris l'idée de la théorie de l'information de Shanon, où on associe une quantité d'information à une chaîne de caractères. Mais il l'a étendue en considérant qu'une chaîne de caractères est un programme (disons qu'elle code une certaine machine de Turing), et en mesurant la quantité d'information contenue dans la chaîne de caractères que *produit* ce programme. On peut dire pour simplifier que la théorie de Shanon mesure la quantité *statique* d'une chaîne de caractères, alors que la théorie de Chaïtin mesure sa quantité *dynamique*.

[2] En fait, cela importe sans doute. Nous discutons ce point ensuite.

Bien évidemment, on a des théorèmes de limitation exprimant dans ce contexte les résultats de Gödel. Le résultat principal peut s'énoncer (quelque peu informellement) :

> *Un programme de complexité k ne peut pas démontrer qu'un programme a une complexité k' avec* $k' > k.log_2 k$

Ce théorème concerne donc des programmes qui démontrent qu'un programme donné a une complexité au moins égale à une certaine quantité. Mais ces théorèmes ont une portée générale, et expriment bien "qu'un programme ne peut pas faire plus que ce qui y est contenu". Mentionnons que Chaïtin a donné plusieurs définitions de la complexité, et que des théorèmes du même genre tiennent pour chacune d'entre elles.

Il faut être en fait un peu plus précis. La chaîne de caractère dont on mesure la complexité en tant que programme code en fait le programme *plus* les données. Si on change ces données, la complexité du couple (programme + données) change donc. Cela va avoir un effet important si ces données sont elles-mêmes un codage d'une certaine machine de Turing, alors que notre "programme" est un interpréteur de cette machine (par exemple une machine de Turing universelle). On peut ainsi avoir une tour d'interpréteurs s'interprétant, tous étant codés sauf celui qui est "en haut de la hiérarchie". De toute manière, un tel programme fixe *doit* exister (même si les programmes inférieurs "se modifient eux-mêmes"). Mais nous pensons qu'alors un théorème similaire tient, qui exprime une limite sur la complexité du couple (programme "en haut" + données) en fonction d'une complexité intrinsèque de ce programme, de la complexité des programmes de la hiérarchie et de la "hauteur" de la hiérarchie.

Or, une définition possible "d'un programme qui s'améliore lui-même" serait une augmentation de sa complexité à la Chaïtin. Nous concevons combien cette définition, formelle, et qui ne s'intéresse pas vraiment "à ce que le programme fait" a d'insuffisant. Mais c'est là un critère qui, s'il est rempli, prouve qu'il y a eu accroissement de quelque chose. Donc, s'il doit y avoir bootstrap, c'est-à-dire augmentation du nombre d'états de la machine de Turing (Gödel), ou disons, "nouvelles instructions" non contenues, même indirectement, dans le programme initial, il faut que *ces nouvelles instructions viennent de l'extérieur*. Donc, ce qui vient de l'extérieur n'est pas constitué de données, *mais d'instructions*. En dernier ressort, c'est le monde extérieur qui "programme" le système. *Les capteurs envoient des instructions.*

Alors évidemment, on se trouve une nouvelle fois désarmé, puisque nous semblons avoir rejeté "l'essence" du bootstrap à l'extérieur du système. D'ailleurs, une situation banale où la complexité d'un programme est augmentée par ajout d'instructions de l'extérieur se produit lorsque chacun d'entre nous … programme (pourvu que nous nous considérions comme extérieurs au système). Mais les remarques précédentes ne se ramènent pas à ce genre de lapalissade. Dans la nature, il existe des systèmes qui bootstrappent (nous y reviendrons), et on doit rejeter l'hypothèse que ce monde extérieur est un "agent" conscient (qui ne pourrait d'ailleurs augmenter la complexité du système que jusqu'à atteindre sa propre complexité).

Evidemment, cette programmation par l'extérieur du système ne doit pas se faire "n'importe comment", c'est-à-dire que le système doit avoir une certaine capacité à trier, rejeter et organiser les "instructions" qui lui arrivent. Un tel mécanisme est fourni par la sélection naturelle, nous y reviendrons. En particulier, le système doit "survivre" de façon autonome dans le monde, ce qui signifie que la sélection doit s'opérer avant tout en fonction de cette survie. Or, un développement anarchique conduirait vite à un effondrement du système, et donc à sa disparition, ce qui évidemment ne permet pas d'aller bien loin dans le bootstrap… On retombe sur des problèmes de l'auto-organisation, en particulier celui de la viabilité. Nous développons ces questions ensuite.

Mais le système ne doit pas non plus fermer toute porte à des "essais de programmation" par un déterminisme rigide et strict, sous peine de retomber dans les limitations de complexité. Car les instructions qui peuvent augmenter sa complexité sont par nature hors de sa portée !

De plus, on peut se poser la question de savoir s'il est avantageux pour le système de disposer, outre d'un composant de "maintenance de sa viabilité", d'un composant de maintenance de sa capacité de bootstrap, par exemple d'un composant qui juge de la capacité heuristique d'une nouvelle instruction d'augmenter sa propre complexité (même si une démonstration de ce fait est impossible).

Ces questions, bien qu'abstraites et spéculatives, se posent néanmoins, et nous allons faire un détour par certains domaines de l'histoire naturelle pour leur donner corps.

7 L'évolution des espèces

La nature semble présenter des phénomènes de bootstrap. Citons : l'origine de la vie et l'évolution des espèces; le développement embryonnaire (l'ontogenèse); le développement de l'esprit, de l'enfant à l'adulte; (peut-être) le fonctionnement de l'esprit "adulte"; le bootstrap culturel d'accroissement des connaissances de l'humanité; la civilisation, aussi bien dans ses aspects matériels et sociaux.

Tous ces phénomènes se déroulent selon des dynamiques et des lois apparentes différentes, et d'ailleurs fort peu connues. Aussi devons-nous nous méfier des analogies trop directes. Mais il peut être utile pour quelqu'un s'intéressant au bootstrap en Intelligence Artificielle d'aller jeter un œil sur les connaissances scientifiques acquises dans ces domaines, même si l'on n'en devient certainement pas spécialiste.

Nous avons essayé de nous plonger un peu dans les développements en théorie de l'évolution. On sait que, depuis Darwin, l'évolution des espèces est régie par la sélection naturelle, qui peut être exposée en trois points :

1/ Les ressources disponibles à un être vivant sont limitées par l'environnement, incluant les autres êtres vivants de la même ou d'une autre espèce.

2/ Ceux qui présentent des dispositions favorables *héréditaires* à *survivre* dans cet environnement *et à se reproduire* seront favorisés.

3/ Le mécanisme de sélection fonctionne au niveau des individus, pas des groupes.

On mesure dans la version moderne la capacité à survivre et à se reproduire par le *fitness*.

Darwin ignorait les bases génétiques de l'hérédité. Depuis son temps sont nées des théories intégrant ces bases, dont le *néo-darwinisme*. Disons que la vision de l'évolution sur de grandes échelles de temps géologiques a changé[3], en ce qu'elle ne suppose plus une vitesse constante d'évolution, mais au contraire des périodes d'accélération suivies de périodes de ralentissement. Un point de vue un peu différent est celui de la *théorie des équilibres ponctués*, introduite en particulier par Stephen Jay Gould, qui clame que l'histoire évolutive est constituée d'une série de longues périodes de stabilité (équilibre) entrecoupées de très courtes périodes de changement accéléré (ponctuelles à l'échelle géologique, et en particulier indécelables dans les fossiles).

Par ailleurs, les spécialistes de l'évolution étudient la micro-évolution et la macro-évolution. Il s'agit du même phénomène, mais sur des périodes de temps incommensurables.

La micro-évolution étudie en particulier les phénomènes de spéciation, c'est-à-dire par quel mécanisme une espèce peut donner naissance à une nouvelle espèce sans autre intermédiaire. Plusieurs modèles de spéciation (spéciation allopatrique, parapatrique, endopatrique, effet fondateur, renforcement, orthogenèse) ont été proposés, et certains rejetés. Il est possible en micro-évolution de faire des expériences "de laboratoire", soit en examinant des épisodes évolutifs récents pour lesquels on dispose de bons fossiles (les perches du lac Victoria, les mouches de Hawaï), soit en appliquant en laboratoire une sélection sur des individus d'espèces se reproduisant très rapidement (notamment mouches drosophiles, pour lesquelles on connait bien par ailleurs le patrimoine génétique).

La macro-évolution étudie l'évolution sur des durées beaucoup plus longues (millions, voire centaines de millions d'années). Cette étude repose évidemment sur les fossiles. Signalons que l'on dispose de fossiles allant assez loin dans le passé (prokaryotes de 3,8 milliards d'années), et que l'on dispose donc d'un certain matériel pour étudier ce qui suit juste l'éclosion de la vie.

Ce ne sont pas ces quelques faits qui nous intéressent, mais ce que nous pouvons tirer de ce comment les évolutionnistes voient le "bootstrap" de l'évolution.

La plupart des modèles de *spéciation* (micro-évolution) dont nous avons pris connaissance sont basés sur des mécanismes de répartition, de division, et de combinaison d'un pool génétique existant dans une population à un moment donné. Par exemple, la spéciation allopatrique suppose que pour une raison quelconque (par exemple géologique de suppression d'un isthme ou de création d'une chaîne de

[3] Par rapport à l'interprétation initiale dominante du darwinisme, qui n'était pas, semble-t-il, celle de Darwin lui-même.

montagne, ou par voyage vers une île), une population est séparée en deux parties au moins. Il est clair que la pression sélective, si les deux milieux sont différents, ne va pas aller dans le même sens. Par exemple, cela peut conduire à des comportements sexuels différents. Auquel cas, au bout d'un moment, les espèces vont être sexuellement séparées, soit parce qu'elles ne se reproduisent plus entre elles (du fait de leurs comportements sexuels ou de contraintes sur l'accouplement), soit parce que les hybrides issus d'un croisement ne sont pas viables. Ce processus peut être renforcé par ce qu'on appelle *l'effet fondateur*, qui suppose qu'une petite population a été isolée, avec donc un appauvrissement de son potentiel génétique, et que des individus présentant des "combinaisons génétiques" qui seraient normalement éliminées vont pouvoir survivre (par exemple des homozygotes).

Ne détaillons pas plus. Répétons simplement que ces modèles décrivent des mécanismes permettant des combinaisons inédites d'un potentiel génétique existant. On peut donc dire que ces modèles de spéciation ne décrivent pas de "vraies nouveautés", c'est-à-dire des nouveautés génétiques, mais de nouvelles expressions de caractères à partir d'un pool génétique donné.

La macro-évolution doit, elle, attaquer le problème de la nouveauté génétique. Nous avons été un peu surpris et déçu de constater que le seul ressort supposé est la *mutation*, et que celle-ci est supposée se faire *au hasard*.

C'est sans doute notre naïveté qui a pu nous faire croire que nous allions trouver une piste en ce domaine. En effet, l'hypothèse de mutations au hasard comme ressort suffisant à l'évolution des espèces nous semble devoir être vérifiée, c'est-à-dire confirmée ou réfutée. Nous n'avons pas la moindre idée de ce comment cela pourrait être effectué.

Mais supposons un instant que cette hypothèse soit réfutée. Cela signifie alors que les mutations n'ont plus lieu "au hasard", mais qu'elles sont "dirigées", évidemment par le patrimoine génétique déjà présent. Lenat avait, dans un papier de 1984, émis une hypothèse un peu similaire : l'existence de méta-gènes, qui proposent de "bons coups" de mutations. Son argumentation était qu'une espèce disposant de tels méta-gènes serait très favorisée en tant qu'espèce par rapport à une autre qui n'en aurait pas. Il y a cependant trois problèmes dans cette vision. Tout d'abord, il faudrait expliquer comment les méta-gènes ont pu apparaître; ensuite, cette théorie se rapproche de la sélection de groupes, que rejette le darwinisme, basé sur la sélection des individus. Comme l'on ne rejette pas une théorie scientifique sans avoir de bonnes raisons, il s'agit là d'un problème très sérieux. Il faudrait en fait montrer comment une sélection par méta-gènes pourrait rentrer dans le cadre du darwinisme en l'étendant, mais sans le contredire. Enfin, il faudrait peut-être les trouver, ces méta-gènes…

Pour nous rapprocher de l'IA, mentionnons d'autres références que Lenat fournit dans son papier déjà cité, en particulier concernant des travaux de programmation automatique par variations aléatoires et *hill-climbing* effectués dans les années 60 à IBM, et qui ont été un échec. Signalons aussi le récent livre de Koza sur la synthèse de programmes par algorithmes génétiques, où il prétend avoir un certain succès pour des types de programmes délimités. Bref, soit ces tentatives ont échoué ou ont eu des succès limités parce que l'analogie avec l'évolution a été mal exploitée (ce qui

est en particulier visible dans les algorithmes génétiques, où la sélection se fait quasi-directement sur les gènes, et pas sur les phénotypes exprimés), soit il y a un problème avec la théorie de l'évolution telle qu'elle est vue actuellement.

Pour résumer, nous sommes donc dans une situation où le mécanisme de bootstrap proposé par les spécialistes de l'évolution se résume à deux sous-mécanismes :

1/ Un mécanisme de variation génétique aléatoire
2/ Le mécanisme de sélection naturelle, qui, lui, n'a rien d'aléatoire.

Et cela est très embêtant, car, outre nos doutes, nous n'aurons sûrement pas la patience d'attendre qu'un phénomène évolutif analogue se produise pour nos programmes…

8 L'autonomie, l'autopoïèse

Nous avons mentionné que le système subissant un bootstrap devait malgré tout être viable à chaque étape, c'est-à-dire survivre dans l'environnement qui lui est propre, c'est-à-dire être autonome. Cela est un problème très difficile quand on se mêle de programmes qui se transforment eux-mêmes. L'exemple d'Eurisko est flagrant : parmi toutes les heuristiques synthétisées, une seule était vraiment "meilleure" que celles qui lui ont donné naissance, la plupart étaient "égales", et 5 étaient fatales au système sans intervention de Lenat lui-même. Malgré les doutes que l'on peut émettre sur la précision de ce compte-rendu scientifique, le pattern semble raisonnable : un système qui se modifie lui-même a toutes les chances de s'effondrer.

Cela peut être dû à la trop grande intolérance du "milieu". Par exemple, un programme modifié à l'aveuglette a toutes les chances "d'être mauvais". Les langages de programmation ne sont donc peut-être pas, pris brutalement, un bon substrat de synthèse de nouveaux mécanismes. On peut aussi imaginer des mécanismes ad hoc permettant de "limiter la casse". Les "erreurs" sont inévitables, et de tels mécanismes sont sans doute indispensables. Un composant visant à améliorer la viabilité tend à accroître l'autonomie du système, comme l'a montré le travail de S. Kornman.

Mais cela ne répond pas à la vraie question : *qu'est-ce que survivre* ? Lorsque l'on écrit un programme, on pense à sa *fonction*. Par exemple, un programme calcule le PGCD de deux nombres, un autre est un moteur d'inférence, etc. Mais cette fonction est extérieure au système (c'est "l'utilité sociale" que nous lui attribuons), or il faudrait une définition qui soit intrinsèque au système. De plus, les exemples de programmes que nous avons cités sont des programmes qui *s'arrêtent*. L'habitude nous interdit en général de voir cette évidence, mais nous sommes bien dans le cadre du calcul défini par Turing. Il faut en effet distinguer machine de Turing particulière et *la* machine de Turing (générique)[4]. Une machine de Turing particulière a une fonction, une sémantique, qui lui sont aussi particulières, qui sont extérieures, et qui ne sont pas susceptibles de généralisation. Par contre, la machine de Turing

[4] Nous entendons par machine de Turing *générique* la classe de toutes les machines de Turing, ou la définition de ce qu'est une machine de Turing quelconque. Cela n'a rien à voir avec la machine de Turing universelle, qui est une machine particulière dont la fonction est d'interpréter une machine de Turing quelconque.

générique a une sémantique intrinsèque : l'arrêt. Une machine de Turing calcule, c'est-à-dire que nous pouvons constater à un moment donné qu'elle s'arrête, et lire alors le résultat du calcul sur sa bande.

Mais il est clair que la machine de Turing (générique), vue comme instrument de *calcul*, est à l'opposé de la notion d'autonomie et de survie. Car un système qui survit par définition *ne s'arrête pas* (jusqu'à sa mort, mais c'est une autre histoire).

Si nous voulons donc donner une définition de la survie, elle doit être générique. Par exemple, nous ne pouvons pas dire qu'un système survit en ce qu'il joue une partie d'échecs. Et cette définition générique doit être à l'opposé de l'arrêt.

Varela a proposé une notion qui nous semble intéressante : *l'autopoïèse*. Un système autopoïétique est un système qui se construit en permanence lui-même, et qui construit en particulier sa frontière avec le monde. L'exemple flagrant est la cellule. Très schématiquement, le noyau produit des substances qui voyagent vers la membrane, substances qui produisent alors la membrane elle-même. Réciproquement, la membrane produit des substances qui transitent vers le noyau pour y construire le noyau. Un système autopoïétique est donc constitué de composants, qui sont chacun des objets produits par d'autres composants, et qui chacun agissent comme des mécanismes produisant d'autres composants. Si l'ensemble forme une "boucle dynamique" bien réglée, alors le système est autopoïétique.

Varela définit le vivant par l'autopoïèse. L'autopoïèse est plus que l'autonomie, c'en est une forme particulière. Varela voit aussi par exemple le système immunitaire comme autopoïétique.

Il nous semble qu'on a là un début de définition potentielle d'une sémantique générique pour les systèmes qui survivent. Si l'on replace cela dans le contexte du bootstrap, on peut dire que l'autopoïèse se constate en observant le système sur une période de temps suffisamment courte. C'est en quelque sorte un "bootstrap arrêté" (c'est-à-dire sans modification globale du système).

La fonction de survie d'un système serait donc l'autopoïèse. C'est bien un critère générique, puisque l'on ne dit pas ce que le système "fait". Son problème n'est pas d'accomplir une "tâche" quelconque dans le monde qui lui serait imposée de l'extérieur, mais ("simplement" !) de survivre. Le système existe par lui-même et pour lui-même, et c'est tout.

Il est à noter que des tentatives, réussie comme Maciste, ou repoussée à un avenir proche comme Shal, présentent des similarités. Dans Maciste, tout programme initial a disparu, parce que l'ensemble des composants de Maciste était capable de compiler (plus généralement traiter) tous les composants de Maciste. Cependant, on a là une sorte d'autopoïèse "arrêtée", parce que le processus de compilation (jusqu'au prochain bootstrap) s'effectue une seule fois, alors qu'un système autopoïétique au sens strict est censé se construire lui-même en permanence. Une seconde différence est que les composants d'une cellule sont en général "consommés" lorsqu'ils agissent comme mécanisme (à l'exception de l'ADN). Par contre, une expertise de compilation de Maciste, lorsqu'elle agit, ne disparaît pas pour autant. Une troisième

différence, plus importante, est que Maciste produit une version compilée de lui-même, mais ne se produit pas lui-même (les expertises de Maciste n'ont pas été écrites par Maciste). Vouloir annuler cette différence est évidemment une autre affaire…

9 Et l'intelligence ? Spéculations sur la mémoire

Nous avons réussi jusqu'à présent à ne presque pas parler d'Intelligence Artificielle. Ainsi, nous avons cherché des métaphores dans la logique et la théorie du calcul, dans la théorie de l'évolution, et dans la vie. Mais en quoi cela est-il lié au problème de l'IA ?

Comme précisé au début, l'IA est difficile entre autres parce que les manifestations de son objet d'étude sont indirectes. Pour nous, le problème de l'IA est "d'ouvrir la boîte", et de proposer des mécanismes qui pourraient y être et rendre compte des phénomènes observés. Il nous semble donc normal de rechercher l'inspiration tous azimuts, pourvu que l'on revienne au bout d'un moment au problème initial.

Nous allons proposer une idée, purement spéculative, et qui n'a à ce jour fait l'objet d'aucune tentative d'exploration.

Une partie essentielle de l'esprit est la mémoire. Jusqu'à présent, dans tous les systèmes d'IA, la mémoire est assurée de la même manière que sur un ordinateur : on dispose d'un appareil qui peut retenir de façon permanente une liste de symbôles.

Or, où est le "disque dur" dans le cerveau ? Sans y connaître grand' chose, on sait que "tout" bouge dans le cerveau. Ou alors il faudrait voir la partie "stable" de la mémoire dans le réseau de connections : une connexion établie correspond à une mémorisation, une connexion détruite correspond à un oubli.

Evidemment, les gens ne sont pas aussi naïfs, et ils supposent bien que la mémoire est "répartie" entre les connections. C'est-à-dire qu'un "objet" mémorisé le serait par une grande quantité de connections - sûrement de façon redondante - et qu'en même temps une connexion participerait à la mémorisation de plusieurs objets. On aurait donc une relation multivaluée dans les deux sens entre symboles et objets. Mais il n'y a pas à ma connaissance de théorie précise basée sur cette idée[5].

D'autres émettent l'hypothèse que l'on peut trouver de la stabilité dans un processus dynamique. Par exemple, la Terre tourne autour du soleil, mais son orbite est un objet fixe. Il faudrait donc voir la mémorisation dans l'orbite, pas dans la Terre. On parle ainsi de "réseaux d'oscillateurs" que l'on étudie via des systèmes différentiels. Le chaos n'est pas loin.

Cependant, l'hypothèse même que la mémoire doive mémoriser des symboles, et que le cerveau est un appareil à manipuler ces symboles de façon similaire à un ordinateur, est violemment combattue par des neuro-biologistes comme Gerald Edelman.

---

[5] Cette affirmation peut être considérée comme une grosse bêtise par les connexionistes.

On peut effectivement émettre des doutes sur cette hypothèse. Ainsi, un résultat d'expérience en Sciences Cognitives nous a particulièrement frappé. Des cobayes ont été choisis et filmés dans une scène banale de la vie. Puis, 18 mois plus tard, on leur a fait subir un interrogatoire pour qu'ils décrivent cette scène. J'ignore s'ils avaient été prévenus de cet interrogatoire futur au moment du film, ou si, disons, quelque chose d'exceptionnel avait pu les marquer et aider à la mémorisation. Mais même dans ce cas, l'auteur ne croit pas pouvoir vous révéler ce qu'il a fait entre 18h30 et 19h le 19 février 1992 (parce qu'il ne s'en souvient plus).

Le but de cette expérience était en fait d'étudier le type d'interrogatoire auquel ont été soumis les cobayes. Celui-ci ne se focalisait pas sur la scène. On leur demandait par exemple de se remémorer des situations où ils s'étaient rappelé quelque chose, expérience qui les aurait marqués (quoique cela devait aussi être trop direct). Bref, il s'est avéré que les cobayes se remémoraient 90% de la scène.

Qu'est-ce que cela signifie ? Tout d'abord, même sous l'hypothèse que la scène était "directement" stockée quelque part, il a fallu exciter des voies très indirectes pour y avoir accès. La mémoire n'est pas une gigantesque base de données. Mais cela suggère aussi, pour aller plus loin, que la scène *n'était pas directement stockée*.

On peut en effet se demander si la mémoire humaine a une telle capacité qu'elle puisse *tout* ou presque mémoriser de la vie quotidienne. Mais encore une fois, même si ce "tout" était possible, "l'indexation" serait problématique.

On a donc certainement des connaissances de mémorisation, qui permettent de ne pas mémoriser un fait, mais "ce qu'il veut dire". Mais pour aller plus loin, il faut aussi supposer que ces connaissances ne sont pas mémorisées. En définitive, "rien" n'est mémorisé sur le long terme. Plus précisément, on peut voir la mémoire comme un système autopoïétique, dont les mécanismes produisent des "éléments de mémoire", ces mécanismes étant eux-mêmes produits par des "éléments de mémoire".

En particulier, les nouveaux "faits" provenant dans le système via les capteurs doivent être considérés comme des "instructions", qui produisent les éléments de mémoire qui permettront de retrouver ces faits.

On aurait donc en permanence production (*a priori* et inconsciente) de souvenirs, de telle sorte que lorsque une "demande de rappel" est effectuée, un des souvenirs produits va plus ou moins correspondre à cette demande, et provoquer la production d'autres éléments de mémoire. Le fonctionnement est à rapprocher de la façon moderne de voir le système immunitaire : selon lui, le système immunitaire produit en permanence des substances (les anticorps) *susceptibles* de s'attaquer à des antigènes, même si ces antigènes n'ont jamais été rencontrés auparavant. Par contre, lorsqu'un antigène est introduit dans l'organisme, la production de l'anticorps "sélectionné" est fortement accrue, et une "mémoire" de cela est conservée. Si l'on retourne à la métaphore de la mémoire, un souvenir dont on s'est "consciemment" souvenu va rester disponible beaucoup plus facilement.

Cette idée, abstraite et spéculative, peut aussi être rapprochée de la notion de *réintroduction* d'Edelman. La réintroduction est le mécanisme selon lequel, par exemple, le "résultat du traitement" par le néocortex de "données" issues du cortex

visuel est réintroduit dans celui-ci (une bonne image de ce que pourrait être une “image mentale”).

10 Une architecture et de grands points d’interrogation

On a subodoré dans les paragraphes précédents un modèle d’un système réflexif. Ce modèle est constitué d’objets, dont certains peuvent agir comme mécanismes sur d’autres objets. Plus formellement, on a un univers O d’objets, et une fonction m (comme mécanisme) de O dans F(O,O) (l’ensemble des fonctions de O dans O). Si o est un objet de O, alors o peut aussi être vu comme un mécanisme f=m(o) qui agit sur les objets de O. Ainsi, dans certaines circonstances à préciser, si o est mis en présence d’un autre objet x, alors o pourra agir sur x pour le transformer en f(x). Bien évidemment, o est lui-même susceptible d’être transformé par d’autres objets, ou le résultat par transformation d’autres objets.

Le séquencement de ces transformations doit être “réglé” par des lois immuables, qui tiennent le rôle des lois de la physique. Il est difficile de décrire de telles lois simulées, car elles agissent selon ce dont o est fait. En effet, nous avons introduit des objets abstraitement, sans dire de quoi ils sont constitués.

Nous avons été un peu surpris de constater que ce petit modèle amusant, largement complété, avait déjà été introduit et étudié par des logiciens, notamment Mostowski. Leur but est de faire une théorie générale, abstraite et axiomatique du calcul (entendons-nous, une théorie de la même veine que celles de Gödel, Turing et Church, mais pouvant s’appliquer à des objets plus généraux que les entiers).

On retombe donc apparamment sur le calcul. Cependant, on voit bien qu’on peut imposer une autre sémantique que l’arrêt, quelque chose qui ressemble à l’autopoïèse. Une définition théorique doit donc être possible.

Cependant, cette étude nous entraînerait sans doute trop loin, et nous couperait des réalités et de sources d’inspiration plus vivantes…

En tout état de cause, ce qui précède pose des questions, auxquelles nous n’avons pas de début de réponse à ce jour :

1/ Qu’est-ce que l’autopoïèse ?

2/ Comment la réaliser ?

3/ Quelles sont les conditions pour qu’un système autopoïétique le reste après modification ?

4/ Un système peut-il juger des “modifications proposées par le monde extérieur” ?

11 Quelles expériences tenter ?

Plusieurs expériences de bootstrap ont été tentées en Intelligence Artificielle : Maciste, Eurisko et le système de Bardinet, entre autres.

Il nous semble que, pour se lancer dans une nouvelle expérience, il serait souhaitable d'avoir les idées un peu plus claires sur les notions d'autopoïèse et de modification d'un système autopoïétique. Cependant, c'est aussi l'expérience qui permet de clarifier les choses.

Un problème se pose lorsque l'on désire commencer un travail : sur quel problème de base travailler ? En effet, la démarche (presque) toujours suivie en Intelligence Artificielle depuis Turing consiste à choisir une activité jugée "intelligente" et à tenter de la reproduire.

C'est là sans doute un problème crucial.

Tout d'abord, cette démarche peut conduire à une confusion extrêmement commune en IA : la confusion entre connaissance et mécanismes mentaux.

Par exemple, beaucoup de travaux ont été consacrés à la démonstration de théorèmes de mathématiques. Une large partie de ces travaux ont eu pour base le formalisme logique, considérant que les arguments logiques d'une preuve pouvaient être identifiés avec les mécanismes mentaux produisant cette preuve. Ces travaux ont donné peu de résultats probants, la raison étant à notre avis essentiellement cette confusion entre logique et moteur de la découverte de la preuve.

De là, un certain nombre de travaux ont voulu considérer que la preuve *et la logique sous-jacente* sont des *objets* issus de *mécanismes*, et que ces mécanismes relevaient de *connaissances*. Ces connaissances ne sont pas des connaissances mathématiques en soi, mais des connaissances sur la manière de trouver une preuve mathématique. Ces travaux, dont Muscadet est un exemple, ont eu un succès beaucoup plus important que les approches formalistes. Pourtant, ils présentent encore des problèmes importants. Ainsi, il est souvent nécessaire de rajouter des connaissances pour démontrer un nouveau théorème, et pas seulement des connaissances "de base" sur l'éventuel nouveau domaine mathématique abordé. De plus, il n'est pas du tout sûr que l'ajout de ces nouvelles connaissances ne va pas "troubler" le système si on lui demande de résoudre à nouveau un problème qui lui a été précédemment posé. En effet, il est possible que ces nouvelles connaissances offrent de nouvelles pistes de recherche que le système n'est pas capable d'écarter, alors que, dans l'ignorance où il était de ces possibilités, il n'avait pas antérieurement ces problèmes.

Il nous semble que ces problèmes ont deux origines. Tout d'abord, ce genre de système repose sur l'hypothèse que les connaissances constituent l'essentiel des mécanismes mentaux. Or, on peut se demander s'il ne s'agit pas encore d'une confusion, si les connaissances ne sont pas elles-mêmes le *produit* des mécanismes mentaux, et pas les mécanismes mentaux eux-mêmes. Deuxièmement, un domaine comme les mathématiques est justement un domaine qui exige beaucoup de compétences prélables. Ce qui conduit aux connaissances.

Le choix d'un domaine d'étude pour baser des recherches sur des mécanismes d'apprentissage génériques, par exemple, nous semble donc difficile et crucial. Un domaine trop "cognitif" risque fort de nous pousser à favoriser la mise en œuvre de connaissances, au détriment de mécanismes généraux. Un exemple caricatural : faire

un système qui “apprenne” la multiplication à partir d’exemples et de la relation “successeur” sur les entiers naturels.

Il n’est donc certainement pas bon de prendre pour “problème de base” d’expérimentations des problèmes qui réclament pour leur résolution des connaissances *trop précises* ou *correctes*. Cela nous semble d’autant plus vrai pour un système d’apprentissage ou qui doit bootstrapper, pour une raison simple : un tel système *ne peut pas* produire de connaissances précises (sinon à être très sophistiqué, et sans doute hors de portée d’un être humain seul) .

En effet, un système qui doit se bootstrapper n’est sans doute pas un système “rationnel”. De manière ultime (théorie de la complexité de Chaïtin), s’il doit s’améliorer, *il ne peut pas l’être*. En conséquence, un système peut et *doit* contenir des connaissances “fausses”. Un exemple de méthode d’apprentissage extrémiste dans l’autre sens (c’est-à-dire rationnelle dans son contenu et ne générant que des connaissances réputées exactes) est l’EBL (Explanation-based Learning). Sans détailler ce qu’est l’EBL, disons que son principe est de générer des connaissances qui expliquent *exactement* pourquoi une certaine situation s’est produite. Il s’avère que les connaissances générées sont extrêmement précises, et même trop précises, pour pouvoir s’adapter à des cas inconnus. Nous ne prétendons pas que pouvoir analyser les raisons d’une situation n’est pas intéressant, mais que cela est insuffisant. De plus, l’EBL est une méthode assez sophistiquée, dont on ne voit pas très bien comment elle pourrait se constituer à partir de quelque chose d’assez simple.

En résumé, le choix du problème de base est délicat. Il doit être *réel* (par exemple nous ne considérons pas comme réel un problème de planification à la STRIPS, il s’agit plutôt d’un problème formel de combinatoire d’opérateurs formels). Il ne doit pas réclamer des connaissances trop poussées (comme les mathématiques). L’objectif du système ne doit pas être trop élevé, ou du moins, la mesure de son succès ne doit pas l’être (un bon programme d’échecs). Bref, la résolution d’un problème de base ne doit réclamer que des actions “élémentaires” et “grossièrement correctes”. Du moins si l’on veut se focaliser sur le bootstrap.

Il est d’ailleurs à noter que la génération de programmes n’est certainement pas un bon problème de base, car c’est à la fois un problème très complexe et réclamant des connaissances très précises. Ce qui est un problème pour des systèmes contenant un compilateur…

Nous avons mentionné que le problème de base devait être soluble à partir d’actions élémentaires. Ce problème se pose aussi pour les systèmes d’apprentissage. Ainsi, une notion communément utilisée par ces systèmes est celle *d’échec* ou de *succès* (ou des versions similaires comme le renforcement). Nous suggérons que ces notions ne doivent pas être considérées comme primitives, mais sont au contraire le résultat de processus très élaborés.

Tout d’abord, le succès ou l’échec dans un système d’apprentissage est l’estimation d’une situation visant à une génération ou à une sélection de mécanismes de divers ordres. Si cette estimation doit être précise, on retombe dans le travers des connaissances analytiques. Sinon, elle est grossière, et cette notion imprécise est

elle-même susceptible d'apprentissage. De plus, on pourrait dire que lorsque l'on ne sait pas, on ne sait pas non plus si l'on a échec ou succès. Il est donc difficile de porter des jugements *a posteriori* sur les actions entreprises. Cela suggère qu'il faut porter en quelque sorte ces jugements *a priori*. D'ailleurs, il faudrait encore que le système ait la capacité d'arriver à une situation où un échec ou un succès soit mesurable. L'approche prise en général en IA est de considérer que l'on a à la base un résolveur de problème "bête" qu'il s'agit d'améliorer. Mais c'est oublier que ce "résolveur bête" est déjà très sophistiqué. Par exemple, un système capable de trouver un plan d'utilisation d'opérateurs de planification par simple recherche combinatoire est plus sophistiqué que les "connaissances de contrôle" apprises.

Un exemple de la vie courante. Une petite fille ne savait pas faire de vélo. Un matin, alors qu'elle n'avait pas touché à son vélo depuis des semaines, elle se réveille et déclare "ça y est, je sais faire du vélo". Elle monte sur son vélo, et pédale avec succès. Il y a certainement eu apprentissage dans cet exemple, mais il est difficile de l'expliquer de manière simple par essai-erreur et succès et échec.

12 En conclusion

Nous tentons de chercher quelque chose de très différent de l'IA actuelle. Pour Mac Carthy et sa "tendance scientifique", l'IA consiste à modéliser le monde de "sens commun", à poser des problèmes sur ce monde, et à imaginer des systèmes résolvant ces problèmes. A l'opposé de ce "néo-behaviorisme", nous croyons qu'il faut "ouvrir la boîte de l'esprit" et chercher ses mécanismes. Ceux qui sont impliqués dans les approches multi-agents, les connexionistes ou les tenants de la vie artificielle voient l'esprit comme un processus émergent de processus élémentaires. Nous pensons que l'esprit est plus que cela : si l'on met des molécules ensemble, nous avons un gaz; mais si nous mettons les constituants élémentaires de l'intelligence ensemble, nous n'avons pas l'intelligence (de la même manière que si nous mettons les constituants de la cellule ensemble, nous n'avons pas une cellule vivante).

1/ Etudier le bootstrap, du genre évolutif, et "permanent".
2/ Partir de problèmes élémentaires, au sens où ils peuvent nous permettre d'explorer les "éléments" des mécanismes à découvrir.
3/ Mener une réflexion générale, "tous azimuts".
4/ Ne pas se perdre dans les méandres de la "métaphysique".
5/ Si l'IA doit fonctionner, ce sera en définitive à coup de "poulies et de courroies".

Jeu : Cherchez l'intrus

[Chaïtin, 19]

[Dormoy, 19]

[Edelman, 19]

[Edelman, 19]

[Einstein & Infeld, 19]

[Gödel, 1972]

[Kornman, 1993]

[Kuhn, 19]

[Lenat, 1984]

[Pitrat, 1984]

[Pitrat, 199]

[Minton, 19]

[Rosenfield, 19]

[Skelton, 1993]

[Turing, 194]

[Varela, 19]