

6 - La conscience est-elle un processus algorithmique ?

par Hervé Zwirn

Le problème de la conscience est sans doute l'un des plus difficiles sinon le plus difficile auquel on puisse s'attaquer. J'en veux pour preuve non seulement la très abondante littérature qu'il a suscitée mais surtout la part de celle-ci consacrée non à sa résolution ou à l'exposé de propositions de solutions mais simplement à montrer que le problème est mal posé, à tenter de le formuler clairement, voire à essayer de montrer qu'il n'a pas de solution ou même que le problème n'existe pas.

Mon objectif dans cet exposé sera extrêmement modeste et limité. Je me contenterai d'aborder la question selon un angle bien précis et donc forcément restreint en commençant par formuler de manière non ambiguë une des questions que l'on peut se poser au sujet de la conscience, à savoir "la conscience est elle un processus algorithmique ?", question qui peut se décliner en deux versions : une version faible, "peut-on *simuler* la conscience à l'aide d'un algorithme ?" et une version forte, "la conscience émerge-t-elle *réellement* lors du déroulement de certains processus algorithmiques?". Ces questions peuvent être exprimées d'une manière imagée sous la forme: "un ordinateur peut-il simuler la conscience ?" (version faible) ou "un ordinateur peut-il réellement être conscient ?" (version forte).

Je vais commencer par donner un certain nombre d'éléments pour que cette question ait un sens précis. Je présenterai ensuite deux arguments qu'on a pu opposer à une réponse positive à cette question et les réponses éventuelles à ces objections. Je n'ai donc aucune prétention à couvrir une proportion notable du sujet et même dans le champ restreint de la question à laquelle je vais m'intéresser, je serai loin d'être exhaustif.

Précisons tout d'abord le sens du terme "processus algorithmique" car la simple phrase "un ordinateur peut-il penser ?" a quelquefois tendance à induire en erreur ceux qui, ne connaissant pas la définition précise de ce qu'est un processus algorithmique, utilisent des arguments non recevables pour donner une réponse négative.

Calculabilité

On a une notion intuitive de ce qu'est une fonction calculable $f(n)$ dont l'argument est entier. C'est une fonction dont on saura toujours calculer la valeur à partir de la donnée de son argument. En général, on pensera qu'une fonction est calculable si on connaît (ou si on est susceptible de déterminer) la liste des opérations qu'il convient d'effectuer à partir de n pour trouver $f(n)$. La difficulté de cette définition est que la liste des différentes opérations qui peuvent intervenir est a priori infinie. Est-ce qu'il est possible de la déterminer précisément et exhaustivement ? C'est une question à laquelle il a été répondu aux alentours de 1936 par Turing, Church, Post, Gödel et d'autres.

La thèse de Church-Turing (que je vais expliciter ci-dessous) fixe la classe des fonctions calculables et détermine en conséquence la liste des opérations qui doivent permettre de calculer toute fonction de ce type. Ceci signifie d'une part, que toute fonction calculable l'est en utilisant uniquement ces opérations et d'autre part, que rajouter d'autres opérations (qui sont intuitivement

faisables) ne permet pas de calculer des fonctions autres qu'on ne pourrait pas calculer à l'aide des premières.

La thèse de Church-Turing dit qu'une fonction de N dans N (ou ce qui revient au même d'un ensemble dénombrable dans un ensemble dénombrable) est calculable si et seulement si :

- Il existe une machine de Turing T qui la calcule, ou si (ce qui est équivalent),
- C'est une fonction récursive partielle ou si (il existe de nombreuses autres formulations équivalentes).

La force de cette thèse réside dans le fait qu'on a pu démontrer qu'un grand nombre de définitions de la calculabilité, partant d'intuitions différentes, sont en fait équivalentes, ce qui semble prouver qu'elles capturent bien toutes le concept intuitif de calculabilité en le prenant par des bouts différents mais qui finalement reviennent au même. De plus, aucune fonction dont on pourrait penser qu'elle est intuitivement calculable mais qui ne soit pas calculable par une machine de Turing (ou qui ne soit pas récursive) n'a pu être produite. Cette thèse est donc maintenant communément acceptée par tous les mathématiciens et informaticiens.

Machine de Turing

Il existe plusieurs manières équivalentes plus ou moins sophistiquées de décrire une machine de Turing. Nous adopterons la définition simple suivante : une machine de Turing est un dispositif idéal composé d'un ruban infini découpé en cases, le long duquel se déplace une tête de lecture-écriture. Une case peut contenir le symbole 1 ou être vide. De plus, la machine peut se trouver dans un certain nombre fini d'états. Au départ, la machine se trouve dans l'état 1 et sa tête de lecture se trouve à gauche de la partie du ruban sur laquelle des 1 sont écrits. A gauche de la tête, le ruban est donc vierge. Les 1 présents représentent la donnée d'entrée de la machine. La machine exécute alors un programme qui est une suite d'instructions du type : lire le caractère inscrit sous la tête de lecture puis a) si le caractère est un blanc écrire un 1 (ou laisser un blanc) se déplacer ensuite vers la droite (ou vers la gauche) et passer dans l'état $N^{\circ} k$ ou s'arrêter, b) si le caractère est un 1 alors laisser un 1 (ou écrire un blanc) puis se déplacer vers la droite (ou vers la gauche) et passer dans l'état $N^{\circ} k'$ ou s'arrêter.

Turing a montré qu'avec un dispositif de ce type, pour toute fonction calculable, il est possible de trouver une machine programmée comme il convient, qui partant d'une donnée n présente au départ sur le ruban, s'arrête en ayant écrit $f(n)$ sur le ruban. Chaque fonction calculable est donc associable à une machine de Turing particulière (celle dont le programme lui correspond) qui, quand on lui donne le nombre n en entrée, écrit le nombre $f(n)$ sur le ruban et s'arrête.

Mais Turing a démontré de plus, qu'il existe une machine de Turing dite universelle, qui est capable de simuler le calcul de n'importe quelle autre machine de Turing. Je ne vais pas rentrer dans les détails de la manière dont on dit à cette machine universelle quelle autre machine elle doit simuler, ce qu'il faut retenir c'est qu'une machine de Turing universelle est capable de calculer n'importe quelle fonction calculable. Si on met de côté le fait que la mémoire de nos ordinateurs actuels est limitée (contrairement au ruban d'une machine de Turing qui doit être infini) un ordinateur n'est rien d'autre qu'une machine de Turing universelle. Il sait faire exactement la même chose, ni plus ni moins.

Le fait de dire que les fonctions calculables sont les fonctions qu'une machine de Turing peut calculer signifie que l'on peut prendre comme opérations élémentaires servant à calculer toute fonction calculable, les opérations de base d'une machine de Turing (à savoir: changer d'état, déplacer une tête de lecture à droite ou à gauche d'une case sur un ruban, lire un caractère, écrire ou effacer un caractère, s'arrêter).

Fonctions récursives

La deuxième manière, équivalente je le rappelle, de présenter la thèse de Church-Turing fait référence aux fonctions récursives. Une fonction récursive est définie à partir des fonctions de base que sont la fonction zéro, la fonction successeur et les fonctions projection, et de la construction de nouvelles fonctions par composition, récursion primitive¹ (on obtient alors les primitives récursives totales c'est à dire partout définies) et minimisation² (on obtient alors les récursives partielles). Le fait de dire que les fonctions calculables sont les fonctions récursives partielles signifie que l'on peut prendre comme opérations élémentaires servant à calculer toute fonction calculable, les opérations de base que nous venons de mentionner. Les deux définitions (par machine de Turing et par récursivité) sont équivalentes et elles sont aussi équivalentes à d'autres formulations comme le λ -calcul proposé par Church³.

Une fois qu'on a une définition précise de ce qu'est une fonction calculable, on peut donner une définition précise de ce qu'est un algorithme.

Algorithme

Le programme, c'est-à-dire la liste des opérations, qu'exécute la machine de Turing qui calcule une fonction (ou la liste des opérations de composition, récursion primitive et minimisation qu'il faut effectuer à partir des fonctions de base pour obtenir la fonction) est un algorithme. C'est la description de la succession des opérations élémentaires à effectuer pour passer de l'argument de la fonction à sa valeur. C'est donc d'une certaine manière la description d'un calcul. Un ordinateur qui fonctionne ne fait rien d'autre qu'exécuter un algorithme.

Système complexe et émergence

Je voudrais maintenant donner quelques précisions sur la notion d'émergence. On a déjà eu l'occasion d'évoquer dans ces conférences le concept de système complexe, composé d'un grand nombre de constituants en interaction non linéaire. Les systèmes complexes font souvent apparaître un comportement ou des propriétés qu'on qualifie d'émergents. Une propriété émergente est une caractéristique qui apparaît lorsqu'on observe le système au niveau global et qui n'est pas prévisible (ou au moins pas visible) au niveau local. La propriété pour l'eau à la température et à la pression ambiante d'être liquide est par exemple une propriété émergente qu'il serait extrêmement difficile de prédire directement à partir de la donnée de la structure de la molécule d'eau et des lois de la physique microscopique. Il n'est pas possible ici de rentrer plus dans le détail de ce qu'est l'émergence dans les systèmes complexes mais il nous suffit de savoir que la dynamique de tels systèmes engendre quelquefois une propriété globale de l'ensemble du système qui résulte des interactions de ses composants. Dire que la conscience est une propriété qui pourrait émerger lorsqu'un algorithme se déroule signifie que lors de l'exécution de l'algorithme par un dispositif

¹ On appelle récursion primitive le processus consistant à construire la fonction $h(x,y)$ par l'itération des formules suivantes : $h(x,0)=f(x)$, $h(x, s(y))=g(x,y,h(x, y))$.

² On appelle minimisation le procédé permettant de définir : $h(x)=$ le plus petit y tel que $f(x, y)=0$ ou indéfini s'il n'y en a pas.

³ Le symbole λ est défini par la formule $\lambda x.[f(x)] = f$ (par exemple $\lambda x.[\sin(x)]$ représente la fonction sinus).

donné, le système constitué par ce dispositif en train d'exécuter l'algorithme pourrait ressentir une sensation identique à celle que nous éprouvons lorsque nous disons que nous sommes conscients. Il est évident qu'une telle hypothèse paraît choquante a priori et c'est justement le but de la discussion qui va suivre d'en examiner la plausibilité.

Nous sommes maintenant armés pour exposer la position des tenants de la thèse de l'Intelligence artificielle forte et pour examiner les objections qui lui ont été opposées.

Les différents niveaux de points de vue

Je reprendrai les 4 niveaux que Penrose utilise dans son dernier livre à ce sujet⁴.

A. Toute pensée se réduit à un calcul; en particulier, le sentiment de connaissance immédiate consciente naît simplement de l'exécution de calculs appropriés.

B. La connaissance immédiate est un produit de l'activité physique du cerveau; mais bien que toute action physique puisse être simulée par un calcul, une telle simulation ne peut par elle-même susciter la connaissance immédiate.

C. La connaissance immédiate est suscitée par une action physique du cerveau, mais aucun calcul ne peut simuler cette action physique.

D. On ne peut expliquer la connaissance immédiate à l'aide du langage de la physique, de l'informatique, ni de quelque autre discipline scientifique que ce soit.

Le point de vue D nie résolument la possibilité de progresser dans la compréhension de la conscience. Ses partisans ont un point de vue que Penrose qualifie de mystique. De toute façon, cette position clôt la discussion. Laissons là donc de côté.

Le point de vue A est celui qu'on appelle souvent la thèse de l'IA forte. Il est soutenu par exemple par des gens comme Dennett et Hofstadter⁵. Il consiste à considérer qu'à partir d'un certain niveau de complexité de l'algorithme considéré, la conscience (au sens réel du terme, c'est à dire celui de l'expérience privée ressentie) émerge spontanément lors de l'exécution de l'algorithme.

Le point de vue B, qualifié quelquefois de thèse de l'IA faible, prétend qu'on peut simuler par programme un comportement conscient de telle sorte que de l'extérieur, on ait l'impression que le programme est conscient, sans pour autant que celui-ci le soit réellement.

La différence entre le point de vue A et le point de vue B est matérialisée par l'acceptation ou non de ce qu'on appelle le "test de Turing". Pour savoir si un ordinateur peut penser Turing a proposé la démarche suivante : un homme interroge à l'aide d'un clavier, un autre homme et un ordinateur placés tous deux dans une autre pièce. Si après une série assez longue de questions, il lui est impossible de déterminer qui est l'homme et qui est l'ordinateur à partir des réponses qu'il a obtenu, alors dit Turing, il faudra attribuer à l'ordinateur la faculté de penser. Si on met de côté les objections immédiates qui viennent à l'esprit contre ce test (par exemple le fait qu'il est facile de voir qui est l'ordinateur en posant une question portant sur un calcul difficile auquel l'homme ne saura pas répondre, objection qui s'élimine facilement), alors accepter le résultat du test de Turing revient à accepter le fait que si la simulation de la pensée est parfaite alors il y a réellement pensée. C'est ce point que les tenants de la thèse B refusent. Searle par exemple le rejette totalement. Il ne refuse pas le fait qu'il soit théoriquement possible de simuler parfaitement la pensée (ou la conscience, ou la connaissance) mais il prétend qu'une telle simulation ne fait pas naître dans l'ordinateur qui exécute le programme de simulation, une véritable sensation personnelle de pensée,

⁴ Penrose R. *Les ombres de l'esprit*, Interéditions, 1995.

⁵ Voir par exemple, Hofstadter D., Dennett D. *Vues de l'esprit*, Interéditions, 1987.

de conscience ou de connaissance. Nous allons voir l'argument qu'il utilise pour défendre ce point de vue.

Le point de vue C est celui que défend Penrose. Il consiste à considérer que non seulement un ordinateur ne peut pas être réellement conscient mais qu'il lui est même impossible de simuler la conscience. Selon Penrose, la conscience fait en effet appel à des processus non algorithmiques. Je présenterai ensuite une des raisons principales qui lui font adopter cette position. Mais je vais commencer par donner un des arguments les plus célèbres, du à Searle, contre le fait qu'un ordinateur puisse réellement être conscient ou même simplement comprendre ce qu'il fait. Cet argument est connu sous le nom de "la chambre chinoise"⁶.

La chambre chinoise de Searle

Pour bien comprendre cet argument, il faut préciser la question à laquelle il répond. Le point souvent considéré comme le plus difficile peut s'énoncer comme suit : Il est évident que les organismes conscients ont des facultés de réactions aux stimuli, de classification, de catégorisation, de réaction et de contrôle de leur comportement. Considérons un organisme qui manifeste un tel comportement qu'on supposera être indiscernable de celui d'un homme. Nous aurons dans ce cas, tendance à prêter une conscience à cet organisme. C'est le fameux test de Turing que nous avons présenté plus haut. Cependant, pourquoi tout ce traitement d'informations ne pourrait il pas se dérouler en l'absence de toute sensation, de toute conscience ? Dit autrement, même si on arrive un jour à décrire par le menu toute la neuro-physiologie et la physico-chimie en œuvre dans le cerveau et qu'on soit ainsi capable de décrire pourquoi et comment un homme réagit de telle et telle manière à tel ou tel stimulus, on n'aura pas expliqué pourquoi et comment l'ensemble des processus neuro-physiologiques ou physico-chimique en question aboutit à faire expérimenter à cet homme une sensation intérieure de conscience. Comprendre la totalité du comportement en terme de physico-chimie et de réaction à des stimuli ne nous fera pas avancer d'un pas en direction de la compréhension de l'apparition de la sensation privée que nous éprouvons en tant qu'être conscient. Comme le dit Thomas Nagel⁷, "il paraît impossible de donner un critère permettant de savoir si un organisme est conscient car on pourrait très bien imaginer qu'il se comporte exactement de la même manière (en tant qu'ordinateur ou en tant que zombie) sans pour autant éprouver la même sensation subjective". En d'autres termes nous ne saurons toujours pas pourquoi cela nous fait cet effet que d'être conscient. Nagel poursuit en remarquant que "si les processus mentaux sont de faits des processus physiques, alors cela produit un certain effet, intrinsèquement d'avoir certains processus physiques. Qu'une telle chose ait lieu demeure un mystère".

C'est pour aller dans ce sens que Searle a proposé son argument de la chambre chinoise.

Imaginez, propose-t-il, que vous soyez enfermé dans une pièce où se trouve un panier plein de symboles chinois. Imaginez vous ne compreniez pas un mot de chinois mais que vous disposiez d'un livre en français disant comment manipuler ces symboles de manière purement syntaxique et que ce livre donne aussi les règles à suivre pour assembler certains de ces symboles lorsque d'autres symboles vous sont présentés. Il vous suffirait alors de suivre les règles données par le livre pour être capable de produire en sortie une suite de symboles constituant une réponse à toute suite présentée en entrée en tant que question. Si les règles sont suffisamment précises et efficaces, il sera impossible à quelqu'un d'extérieur de distinguer une réponse produite de cette manière automatique d'une réponse donnée par un vrai Chinois. Et pourtant, cette manipulation automatique de symboles à laquelle vous vous livrez ne vous donnera en aucun cas une compréhension du chinois. Vous

⁶ Searle J., *Minds, Brains and Programs*, The Behavioral and Brain Sciences, Vol 3, Cambridge University Press, 1980.

⁷ Nagel T., "What is it like to be a bat?", in Nagel T., *Mortal Questions*, Cambridge University Press, 1979.

aurez donc simulé la manière dont un Chinois répondrait à ces questions mais vous ne comprendrez pas le chinois pour autant. Searle en tire la conclusion que la simulation en question ne produit pas une compréhension véritable et que, par extension, la simulation de la conscience ne produirait de la même manière aucune conscience réelle.

La réponse donnée entre autres par des gens comme Dennett ou Hofstadter à cet argument est la suivante : Searle fait une erreur de niveau quand il veut nous prouver qu'il est possible de faire simuler le comportement de quelqu'un qui comprend le Chinois par quelqu'un qui ne le parle pas et ne le comprend toujours pas en se contentant d'exécuter l'algorithme. Tout d'abord, il sous estime totalement la complexité du problème en supposant qu'un opérateur serait capable d'exécuter ces instructions. Etant donné le nombre d'instructions nécessaires et leur complexité, un homme seul ne saurait en aucun cas exécuter le programme en un temps compatible avec sa durée de vie, il faudrait sans doute pour cela toute une armée de personnes travaillant en parallèle. Dans ce cas, ce serait le système global (c'est à dire l'ensemble de ceux qui exécutent les instructions) qui comprendrait et non pas les opérateurs en tant qu'individus. La compréhension est le fait de l'ensemble du système. La même erreur serait faite par quelqu'un qui prétendrait qu'un homme ne comprend pas ce qu'il dit parce qu'aucun de ses neurones ne comprend le Français. Lorsque Searle tente de réfuter l'objection de ses adversaires selon laquelle la compréhension n'est pas le fait des individus qui exécutent l'algorithme mais de l'ensemble de ces individus, il répond que cet ensemble est réduit à lui-même s'il apprend lui-même et lui seul la totalité de l'algorithme et des règles nécessaires pour l'exécuter. Mais comme nous venons de le dire cette hypothèse est totalement irréaliste. L'argument de Searle selon les tenants de l'IA forte n'est donc pas vraiment convaincant même s'il est séduisant à cause d'une sorte d'illusion, celle qui consiste à gommer les différences de niveaux et à sous-estimer la complexité nécessaire. Ceci étant, même si on n'est pas convaincu par l'argument de Searle, dire qu'à partir d'un certain niveau de complexité, le système en tant que tout voit émerger une réelle compréhension de ce qu'il fait reste extrêmement mystérieux et aucune explication du *comment* cette compréhension émerge n'est donnée par les tenants de cette position.

La question de savoir *comment et pourquoi* un processus donné fait apparaître une conscience est semble-t-il la plus difficile qui soit. Si on admet que rien d'autre n'existe que la matière (et l'énergie) alors la conscience doit trouver une explication en termes de processus ne faisant intervenir que la matière et l'énergie, donc en termes de physique. La question est alors de savoir si ces processus qui engendrent la conscience, peuvent être purement algorithmiques. En d'autres termes, la question sera de savoir si, étant supposé que la conscience est un phénomène qui émerge spontanément lors du déroulement de certains processus physiques, il est possible que ces processus soient purement algorithmiques. Si la réponse est oui et Searle admet accepter cette réponse, il semble qu'un ordinateur programmé avec un tel algorithme pourrait expérimenter une sensation interne de conscience. Il semble même que n'importe quel substrat exécutant un tel algorithme pourra expérimenter une conscience. C'est là où la position de Searle semble la plus faible puisque curieusement, il considère bien que le cerveau est une sorte d'ordinateur mais il refuse pourtant de croire qu'un ordinateur puisse être conscient. Poussé à expliquer pourquoi, il adopte une position qui paraît très peu solide selon laquelle, le cerveau engendre la conscience en raison de ses qualités biologiques. C'est donc le substrat biologique qui serait responsable selon lui de l'apparition de la conscience. Une telle position n'est pas très satisfaisante et n'explique en fin de compte pas grand chose.

Si en revanche, comme le suppose Penrose, la réponse est non, alors un ordinateur ne peut ni expérimenter ni même simuler la conscience.

Je me contenterai donc à ce stade de dire que la thèse de l'IA forte n'est pas réfutée mais qu'elle n'est pas prouvée non plus.

La position de Penrose

Je vais maintenant passer à l'argument utilisé par Penrose pour défendre le point de vue selon lequel la conscience nécessite des processus non algorithmiques. Un mot avant sur la philosophie générale de Penrose dans ce domaine. Penrose est intimement convaincu que la physique actuelle est incomplète (ce en quoi il a raison comme le montre le fait qu'on a toujours pas réussi à unifier les quatre interactions fondamentales) mais il pense que cette unification se fera à travers une physique nouvelle faisant apparaître des processus non calculables à l'opposé de quasiment tous les processus que la physique actuelle traite. Je ne vais pas rentrer dans le détail des raisons que donne Penrose et je me contenterai de dire qu'il lie ce problème avec celui de la mesure en mécanique quantique. Nous avons eu l'occasion d'évoquer ici ce fameux problème et de voir comment les théories de la décohérence sont une tentative de solution. Penrose ne s'en satisfait pas et voit dans la gravité quantique non calculable (une théorie qui n'existe pas à l'heure actuelle) une solution possible de ce problème. Sa position est très contestée par les physiciens et personnellement je ne suis pas du tout séduit par le fait de faire intervenir la gravité quantique pour résoudre le problème de la mesure. Ceci étant, la critique majeure qu'on peut lui faire à ce sujet est de faire intervenir une théorie à venir et pour le moins mystérieuse dont lui-même n'a aujourd'hui aucune idée précise, comme solution d'un problème qui est déjà en soi assez préoccupant! De plus, sa position est encore plus fragilisée par le fait qu'un des arguments majeurs sur lequel il s'appuie pour montrer qu'il existe des processus non algorithmiques à l'œuvre en physique est lui-même fortement contestable. Cet argument, qui a été initialement donné sous une forme légèrement différente par Lucas, s'appuie sur le théorème de Gödel. Je vais maintenant être obligé de rentrer dans une partie nécessairement plus technique mais il est difficile d'en faire l'économie si on veut réellement comprendre la position de Penrose. Je vais cependant essayer de présenter les choses de la manière la plus simple possible. Nous aurons l'occasion d'en discuter ensuite et je remercie Jean-Paul Delahaye pour les discussions que nous avons eues à ce sujet.

Le théorème de Gödel

Le théorème de Gödel stipule que dans tout système formel consistant (c'est à dire non contradictoire), il existe une proposition vraie qui n'est ni démontrable ni réfutable en utilisant les démonstrations autorisées à l'intérieur du système lui-même. La proposition de Gödel est constructible mécaniquement à l'intérieur du système. On peut donc l'énoncer sous la forme : "si le système S est consistant, alors il est possible de construire à l'intérieur du système une formule G qui bien que vraie, ne sera ni démontrable ni réfutable". La deuxième partie du théorème dit que la consistance du système n'est pas démontrable à l'intérieur du système.

L'argument de Lucas, repris par Penrose, est alors le suivant: Supposons que nous soyons décrit par un algorithme ou ce qui est équivalent que nous soyons décrit par un système formel. Dans ce cas, nous possédons une proposition de Gödel que nous pouvons construire mais que notre algorithme personnel ne peut pas démontrer. Comme par définition, tous nos raisonnements sont formalisables à l'intérieur de notre système formel donc sont comme une démonstration à l'intérieur du système, nous ne devrions pas être capables de raisonner pour voir que notre proposition de Gödel est vraie. Cependant, par construction de cette proposition nous le savons. Nous savons donc quelque chose en tant qu'homme que notre algorithme personnel ne peut pas nous fournir. Il en résulte que nous ne nous réduisons pas à un système formel quel qu'il soit.

En fait ce raisonnement est erroné dans la mesure où il oublie un point fondamental du théorème de Gödel : le fait que pour conclure à la vérité de la proposition de Gödel d'un système S, il est nécessaire de supposer que ce système est consistant. La manière correcte de l'exprimer est : "si le système S est consistant alors la proposition de Gödel de ce système est vraie bien qu'elle ne soit pas démontrable à l'intérieur du système". Il en résulte que pour conclure à la vérité de la proposition de Gödel, il faut accepter la consistance du système. Or, la deuxième partie du théorème de Gödel dit justement que la consistance ne peut être démontrée à l'intérieur du système, ce qui signifie que nous n'avons aucun moyen d'être sûr de la consistance du système formel qui nous représente (si nous sommes représentés par un système formel) et que par conséquent nous n'avons aucun moyen de conclure à la vérité de la proposition de Gödel de ce système, pas plus que le système lui-même. La supériorité que Penrose veut mettre en avant pour montrer que nous sommes capables de "sortir" de tout système formel sensé nous représenter, s'effondre et avec elle, la preuve qu'il avance.

On peut présenter les choses de manière imagée à travers le dialogue suivant: Pierre veut prouver à Paul qu'il est plus (qu'il va au delà) tout système formel que Paul prétendra que Pierre est. Selon Lucas et Penrose, pour tout système formel S que Paul prétendra représenter Pierre, Pierre exhibera la proposition de Gödel G du système S et dira: "voilà la proposition de Gödel du système que tu prétends que je suis. Ce système ne peut en aucun cas prouver la vérité de cette proposition, or moi, en tant que Pierre, je sais qu'elle est vraie donc je sais quelque chose que le système que tu prétends que je suis ne sait pas. Donc je ne suis pas correctement représenté par ce système formel".

Le problème de cette argumentation est qu'elle passe sous silence le fait que pour savoir que la proposition de Gödel en question est vraie, Pierre suppose implicitement que le système S est consistant. Or, Pierre n'a aucun moyen de savoir si cela est vrai puisque si S le représente, comme la deuxième partie du théorème dit justement que la consistance de S n'est pas démontrable dans S, Pierre ne peut savoir que S est consistant. Il n'est donc pas justifié à dire que contrairement au système S, lui sait que la proposition de Gödel de S est vraie.

Le raisonnement correct qu'il pourrait faire est le suivant: "Si je suppose que S est consistant alors je saurai que la proposition de Gödel de S est vraie". Mais dans ce cas, Pierre n'a plus aucune supériorité sur S puisque on peut prouver que la formule $\text{Cons}(S) \Rightarrow G$ ⁸ est parfaitement démontrable dans S, ce qui signifie que le système sait aussi démontrer que s'il est consistant, sa proposition de Gödel est vraie. L'argument de Lucas et Penrose tombe.

Pour rendre totalement justice à Penrose, il faut mentionner qu'il tente de justifier le fait qu'un mathématicien est capable de reconnaître la consistance d'un système formel par des moyens qui sont d'une certaine manière liés à son intuition. Mais cet argument repose sur beaucoup d'ambiguïté et sur une conception réaliste platonicienne des mathématiques qu'on n'est pas forcé d'accepter. Nous ne pouvons rentrer ici dans l'analyse des critiques qui peuvent lui être faites mais elles ont été formulées par de nombreux logiciens qui ne trouvent pas satisfaisant le raisonnement de Penrose⁹. A ce stade, nous sommes conduits à penser que l'argument de Penrose n'est pas concluant.

Conclusion

Alors, que peut on conclure de tout ça ?

⁸ Cette formule signifie que la consistance du système S implique la proposition G.

⁹ Voir par exemple la critique très claire de Feferman S., *Penrose's Gödelian Argument*, Psyche, 1995.

L'argument de Searle contre le fait qu'un ordinateur puisse réellement comprendre ce qu'il fait ou puisse être conscient n'est pas convaincant et on peut considérer qu'il a été réfuté par ses adversaires. Il en résulte qu'il n'est pas prouvé qu'à partir d'un certain niveau de complexité, un ordinateur qui exécute un algorithme simulant un comportement humain (j'utilise ce terme flou pour englober à la fois un comportement total d'être humain auquel cas on se pose le problème de la conscience mais aussi un comportement partiel comme celui de comprendre une langue et dans ce cas on se pose le problème restreint de la compréhension etc.) ne ressente pas réellement une compréhension de ce qu'il fait ou ne soit pas conscient. D'un autre côté, rien ne prouve non plus que tel est bien le cas.

Par ailleurs, l'argument de Lucas et Penrose contre le fait qu'il soit possible de simuler un comportement humain par algorithme n'est pas concluant non plus. Il en résulte que rien n'interdit de penser qu'il est possible de simuler un tel comportement par un ordinateur. Mais rien ne prouve non plus que tel est bien le cas.

La conclusion prudente mais il faut bien l'avouer frustrante à ce stade est donc que aucune des thèses A, B et C n'est ni démontrée, ni réfutée.