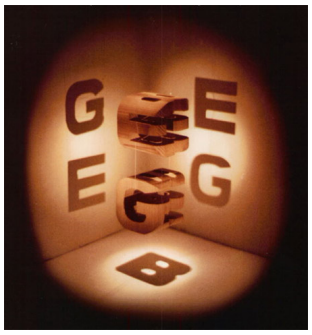
Décembre 2008 **Numéro 42**

Cahiers de l'IMA



Hofstadter Douglas - Gödel Escher Bach

les brins d'une guirlande éternelle

Lecture critique de Thomas Mueller

Institut de Mathématiques Appliquées
Faculté des S.S.P.
Université de Lausanne
Anthropole
1015 Lausanne

Table des matières

Dans ce livre volumineux, de presque 900 pages, Douglas Hofstadter se confronte à toute sorte de difficultés. Le style est sobre et convivial, un compromis entre la rigueur anglo-saxonne et la nécessité de communication d'un ouvrage qui a gagné un prix Pulitzer, et qui pour cela ne se restreint pas thématiquement. Imaginez un conte de Lewis Carroll qui est aussi un ouvrage de neurosciences, qui est aussi une discussion informée d'intelligence artificielle, qui cache des élucubrations savantes sur les langages formels en mathématiques, qui est hantée par l'esprit d'Achille et de Zénon, d'une tortue, d'un crabe, de quelque maître Zen, de métagénies, de tourne-disques indestructibles, de jeu de mots et de musique.

Il y a tout cela (et bien plus) dans ce livre ; il y a des analogies, des jeux, des aventures. Tout ceci ne peut se faire qu'au prix d'une certaine difficulté : ce sont 900 pages qui m'ont demandé une trentaine d'heures de lecture. Bien que très pointu à certains égards (notamment sur les problèmes mathématiques qui sont au cœur de la première partie) le livre devient parfois flou, et ne satisfait pas toujours les désirs cognitifs des lecteurs. La partie conclusive sur la conscience et les boucles étranges est décidément trop courte et peu développée.

Je ne saurais dire si cela compte comme un défaut ou comme une qualité : cela donne certainement envie de lire plus, et Hofstadter a développé ces idées dans d'autres ouvrages.

Finalement, c'est un livre qui suscite des réflexions ; il est daté maintenant de presque 30 ans. Il y a des problématiques qui restent aujourd'hui aussi sombres qu'alors. D'autres nous font sourire : nous savons ce qui s'est réellement passé !

Le lecteur ne peut pas rester neutre. J'ai essayé de clairement séparer mes commentaires des opinions de l'auteur, mais ce n'est pas toujours simple, et j'imagine que certaines des idées que vous allez attribuer à DH sont en fait des ajouts personnels qui se sont glissés entre les lignes. Pire encore : il se peut que vous preniez parfois pour miennes des idées de DH. J'espère qu'il ne m'en voudra pas, et qu'il conviendra que le critique a parfois une vue d'ensemble qui jouit de quelques symboles de plus haut degré surtout lorsqu'il est hors du système à la fois dans l'espace et dans le temps, mais qu'il n'est pas en mesure, lui non plus, d'échapper aux terribles conséquences du théorème de Gödel.

1 Introduction

Le livre commence par l'introduction, plutôt informelle, des trois personnages mentionnés dans le titre :
Johann Sebastian Bach (1685 - 1750) est un compositeur claveciniste et orga-

niste allemand ; il est connu pour ses fugues et canons, qui seront d'ailleurs au coeur de la narration.

Maurits Cornelis Escher (1898 - 1972) est un peintre hollandais, connu pour ses recherches sur la perspective, fortement influencé par les découvertes en mathématiques et physique de son époque.

Kurt Gödel (1906 - 1978) est un mathématicien et logicien austro-américain qui a notamment contribué au développement des deux grands théorèmes de mathématique qui portent son nom.

Le théorème d'incomplétude de Gödel, le plus important des deux théorèmes mentionnés, affirme notamment qu'il est impossible de démontrer la complétude et la consistance (non contradiction) d'un système d'axiomes donné ; cela revient à dire que n'importe quel langage formel peut construire un bien plus grand nombre d'affirmations vraies que de preuves, ce qui permet de conclure qu'il y a toujours des affirmations dont on ne peut pas démontrer la vérité.

Hofstadter considère ces trois auteurs comme les projections sous des angles différents d'un même objet. La reconstruction de cet objet, c'est le livre en question, et le symbole de cette projection triple est l'illustration à côté du titre.

Par la suite, nous nous trouvons plongés dans une discussion fictive entre deux personnages : il s'agit d'Achille et de la tortue discutant entre eux le paradoxe de Zénon. Au cours du développement du livre, nous retrouverons à plusieurs reprises ces personnages et bien d'autres. Ils y jouent un rôle métaphorique évident (ils sont les Gedankenexperimenten de l'auteur), de plus ils vont eux mêmes mettre en place des expériences : on peut dire qu'ils sont une méta-métaphore. Escher leur fournira des arguments de discussion ; les titres des chapitres qui les concernent seront toujours des musiques inspirées de Bach, analogues en rythme et en structure aux fugues et au canons du compositeur allemand.

2 Les langages formels

2.1 Le langage MU et le langage pg

Le premier chapitre est dédié aux langages formels : un langage formel est un ensemble de symboles (des chaînes de caractères), munis d'un ensemble de règles de formation et d'inférence (ce qu'on a le droit de faire avec ces chaînes, leur grammaire). Le premier exemple de Hofstadter est le système MIU. Dans

ce monde, nous avons trois lettres (M, I et U [1]), qui sont les symboles admis dans ce langage. Nous avons ensuite les règles du système MIU qui sont :

- Si vous avez la lettre M suivie d'une chaîne de lettres x [2], vous pouvez la transformer en M suivie de deux fois x. Par exemple, MIU peut devenir $MIUIU$.
- Si la chaîne se termine avec I, vous pouvez ajouter la lettre U à la fin (MUI devient $MUIU$).
- Si trois I apparaissent à la suite vous pouvez les remplacer par U (par exemple $MUIII$ devient MUU).

Il y a des chaînes qui peuvent être obtenues par manipulation d'autres chaînes (par exemple, $MIUIU$ peut s'obtenir en manipulant MIU) et il y a aussi des chaînes impossibles à obtenir (par exemple on ne peut pas obtenir MU en partant de MI).

Si une chaîne peut être obtenue en appliquant les règles du système formel, alors c'est un *théorème*. Il faut bien distinguer ce qui se passe à l'intérieur et à l'extérieur du système : dans le système, l'application stricte des règles est tout ce qui peut avoir lieu. Une machine tout à fait stupide pourrait engendrer des théorèmes par tentatives et erreurs en appliquant les règles aveuglement. Mais un humain a tendance à sortir du système et à le regarder du haut : il peut alors en comprendre certains mécanismes, et il peut parfois l'interpréter en lui donnant une signification. Le langage ne possède pas une signification interne, mais il en possède parfois une externe (et il se pourrait qu'il en possède plusieurs).

Un autre exemple de langage formel présenté dans le livre est le langage pg ; en voici les règles et le fonctionnement :

vous avez trois lettres p, g, $-$, (le tiret fait office de lettre). Toute chaîne de la forme $xp-gx-$ est un axiome du système. Bien sur, x n'est pas une lettre admise dans le système : elle remplace une certaine quantité de tirets.

Par exemple

1. $--p-g---$ est un axiome ;
2. $---------p-g----------$ est aussi un axiome ;

et ainsi de suite. Or, il saut aux yeux que le système pg peut être interprété comme représentant l'addition. En effet, nous pourrions lire p comme *plus*, g comme *égale* (give en anglais), et les tirets comme des chiffres. Mais il ne faut pas se laisser transporter : l'interprétation externe est utile pour que nous puissions voir un sens dans le langage, mais ce sont les règles internes qui

1. Nous distinguons les lettres et expressions du métalangage, sans contenu sémantique en les écrivant en italique M, plutôt que M

2. x n'est pas une lettre du langage à plein droit ; c'est un symbôle que nous utilisons afin de remplacer n'importe quelle chaîne de lettres du langage.

dominent. Par exemple $--p--p--p--g----------$ n'est pas un théorème car il ne respecte pas les règles du langage, bien que $2+2+2+2=8$ soit bien sûr vrai ! Et notre interprétation n'est pas unique : par exemple, on pourrait lire p comme *égale* et g comme *ôté de* et nous aurions une signification différente, bien que le système soit tout à fait équivalent.

Le langage formel peut aussi ne pas avoir d'interprétation du tout. Il peut avoir une interprétation très bizarre ; et il se pourrait qu'il exprime des vérités encore à découvrir. Ce qui est intéressant est bien sûr la possibilité de passer de l'intérieur à l'extérieur du système.

Munis des langages formels, nous sommes maintenant en mesure d'introduire le fameux théorème de Gödel, et quelques métaphores amusantes que Hofstadter nous propose dans son livre.

2.2 Consistance complétude et phonographes

Nous commencons par une rencontre entre nos métaphores vivantes : voici ce qui se passe dans la dernière discussion d'Achille et de la tortue.
La tortue raconte que son ami le crabe possède un phonographe haute fidélité, capable de reproduire n'importe quel disque. Elle prétend que c'est impossible, et elle raconte comment elle a réussi à détruire les phonographes du crabe, en démontrant la justesse de son opinion.
C'est très simple : elle a construit des disques capables d'émettre des sons d'une fréquence très particulière, qui - lorsqu'ils sont lus - cassent le phonographe. Le fait même que le phonographe soit fidèle dans la reproduction rend son destin sans issue !

La tortue a ainsi détruit le phonographe du crabe ; celui-ci ayant été battu, il décida d'acheter un phonographe de meilleure qualité, mais la tortue a une fois de plus été capable de trouver le son destructeur. La seule solution pour le crabe aurait été d'utiliser un phonographe basse-fidélité ; mais dans ce cas, la reproduction de musique serait mauvaise, ce qui bien sûr va à l'encontre même du but du phonographe.

La métaphore est expliquée en détail dans le chapitre suivant :

- Le tourne-disque représente le système axiomatique. Un système haute-fidélité est un système fort, capable de dire un peu tout, alors qu'un système basse-fidélité n'en est pas capable.
- Un schéma de fabrication, utilisé par la tortue afin de trouver le point faible du tourne-disque, représente les axiomes et les règles du système formel, alors que le disque symbolise les chaînes construites avec les ca-

ractères du langage.

- Un disque qui ne peut pas passer sur un tourne-disque est une chaîne qui n'est pas un théorème du langage formel, alors qu'un disque qui tourne est un théorème.
- Un son est une assertion vraie de la théorie : lorsqu'il est reproductible, il peut être interprété par le système, alors qu'un son non reproductible est une assertion vraie qui n'est pas un théorème.

La métaphore nous indique qu'il y a forcément des affirmations d'un langage formel qui ne peuvent être prouvées par ce langage même. Gödel a prouvé qu'il n'y a pas de "phonographe parfait", c'est-à-dire qu'il n'y a aucun langage formel capable d'engendrer toutes les affirmations vraies sous forme de théorème.

Le nom du théorème (théorème d'incomplétude) vient de deux notions qu'on n'a pas encore explicitées.

La consistance signifie que le système formel ne se contredit pas. Dans le cas du tourne-disque, c'est la capacité de faire tourner n'importe quel disque, et de reproduire n'importe quel son qui y serait écrit. La non contradiction concerne l'extérieur du système : il y a contradiction lorsqu'un système a une interprétation naturelle et que cette interprétation se contredit. Bien sûr, puisque l'interprétation n'est pas unique, il est possible de changer d'interprétation, quoi que cela puisse parfois devenir très complexe. La consistance affirme que tout théorème, interprété, est vrai.
Si la non contradiction est une condition *sine qua non* (minimale), nous pouvons imaginer une condition maximale, la complétude.

La complétude affirme que tout ce qui est vrai (et peut être écrit dans un langage formel) est un théorème [3]. La complétude est interne au système ; elle nous dit que ce langage formel est assez stable pour prouver tout ce qu'il arrive à dire. Il contient assez de règles pour tout prouver !

Le problème qui fait coincer le mécanisme (et qui casse le phonographe), consiste en ce qu'il y a des disques qui peuvent avoir comme "trace musicale" le chant "je ne peux pas être lu par ce phonographe". C'est une forme de récursivité (qu'on va discuter en détail par la suite) qui est permise par des langages assez puissants, des phonographes haute-fidelité. On engendre ainsi des contradictions.

Si l'on rajoute de la puissance à notre langage formel (dans la métaphore, il s'agit de construire un phonographe de très haute-fidélité), les choses ne se passent pas mieux. Un tourne-disque sera d'autant plus facilement victime d'un disque à casser des phonographes qu'il est de plus haute fidélité ; mais un tourne-disque qui se protège contre cette possibilité, n'étant pas "haute-

3. Autrement dit : on peut le dériver via les règles du langage.

fidélité", ne pourra pas exécuter des programmes intéressants : dans notre métaphore, cela signifie qu'un langage formel ne pourra pas dire tout ce qu'on aimerait.
Pour y voir plus clair, on analysera maintenant les processus récursifs.

3 Processus récursif et métalangage

Un processus est récursif lorsqu'il se répète identiquement à plusieurs reprises et cela de manière emboîtée. Par exemple, un appel téléphonique A qui est mis en attente afin de répondre à un appel B, qui est mis en attente afin de répondre à un appel C, qui... et ainsi de suite. La récursivité est très importante dans la programmation informatique : elle demande notamment de pouvoir construire des piles, ou des niveaux de profondeur. Dans notre exemple, l'appel C se trouve plus en profondeur que B, et B plus en profondeur que A.

Le langage aussi est hautement récursif ; l'exemple de récursivité le plus frappant est néanmoins celui de certaines successions de nombres en mathématiques. Considérons par exemple la suite de Fibonacci :

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...$$

où les deux premiers chiffres sont choisis au hasard, puis on somme les deux dernières de la liste (ainsi 1+1=2, 2+1=3, 3+2=5 ...). Cette suite nous paraît complètement dépourvue de sens au premier regard. Pourtant, elle est obtenue grâce à un procédé simple et hautement répétitif. D'autres règles peuvent nous donner des suites numériques bien plus chaotiques, alors que derrière elles se cachent des règles très simples.
Les objets récursifs les plus fameux des mathématiques sont probablement les fractales (figure 1), des dessins qui sont composés de fragments, chaque fragment étant une copie du dessin, et chaque fragment du fragment aussi, et de même pour les fragments des fragments de fragment. La figure 2 pourrait être un bon exemple de "toile récursive".

3.1 Signification et connexion

Où réside la signification ? Pour discuter de cette idée, Hofstader nous repropose l'histoire d'Achille et de la tortue, cette fois en train d'écouter un disque sur un tourne-disque. Le disque contient bien évidemment une seule trace musicale, des sillons qui codent pour une seule musique. Mais le pari de la tortue, c'est de prouver que tout cela peut représenter plus qu'une chanson. Afin de le prouver, elle utilise l'engin suivant : lorsqu'on veut écouter une certaine chanson, un tourne-disque TD1 avale le disque et nous fait entendre sa

FIGURE 1 – Exemple de fractale

FIGURE 2 – Poissons et écailles de M.C. Escher

musique. Si l'on veut écouter une autre musique, le tourne-disque TD2 avale le même disque. Le deuxième tourne-disque a pour effet de multiplier les intervalles entre les notes (avec des astuces telles que tourner plus vite). Par exemple la suite RE MI FA, qui est une suite de ton - demi-ton, devient une suite deux tons - ton, dans notre cas RE FA♯ SOL♯. La mélodie obtenue vient de changer, alors que la trace écrite sur le disque est toujours la même ! On pourrait voir dans cela un analogue de l'approche interne/externe des langages formels. Le disque est le langage, alors que le tourne-disque est l'interprétation du langage. Mais on peut aussi se demander où se trouve la musique, c'est-à-dire la signification des sillons du disque. Elle n'est vraisemblablement pas dans le disque (puisqu'un même disque code pour plusieurs mélodies) ; elle n'est pas non plus dans le tourne-disque, qui à lui seul ne peut rien créer. Il est alors sensé de dire que la signification est dans la *relation* entre les deux, une approche que l'on peut qualifier de relationnelle (par opposition au substantialisme), ou, en empruntant un mot à la philosophie de l'esprit, connexionniste.

Hofstadter propose d'autres métaphores qui sont du même esprit : la plus intrigante est certainement la question du rapport langage/contexte dans une chaîne d'ADN.
L'ADN code les informations nécessaires au développement des êtres vivants. C'est l'équivalent du disque dans notre métaphore. Mais ce qui est assez intéressant, c'est qu'il code aussi son propre système de lecture. L'ADN a besoin d'un contexte chimique pour être lu (et ce contexte est bien évidemment la cellule et l'organisme dans lequel il se trouve), mais c'est l'ADN lui-même qui code pour son propre contexte. On a donc une forme d'auto-référence. Hofstadter conjecture que si un brin d'ADN était envoyé dans l'espace et retrouvé par une civilisation extra-terrestre intelligente, il y aurait des chances qu'elle aboutisse à sa compréhension au bout de quelques tentatives (quelque millions peut être !). Par contre l'écriture abstraite, telle que ATTAGCCGCTA ne donnerait pas autant d'information, et serait, hors de son contexte, totalement indéchiffrable. L'expression du phénotype par rapport au contexte est un problème qui est loin d'être une conjecture abstraite des mathématiques : savoir quand le message codé est "exprimé" est le coeur du débat autour du thème de l'avortement.

Finalement, j'oserais proposer une petite idée alternative que Hofstadter ne pouvait pas connaître à l'époque de la rédaction de GEB : on pourrait penser que une partie au moins du "contexte" de l'expression de l'ADN est codée dans la cellule, mais pas dans l'ADN lui même. Il l'est dans l'expression épigénétique, qui est influencée par l'environnement, alors que le code ADN ne change jamais au cours de la vie d'un individu. C'est de cette manière, je crois, que l'environnement à un rôle à jouer comme "contexte", le "tourne-disque" par rapport à l'expression phénotypique du code ADN, l'ADN étant les sillons

du disque.

3.2 Langage naturel

Le langage par excellence est le langage naturel, et le langage naturel est hautement récursif. Prenons par exemple en considération des phrases telles que "*Cette phrase se compose de sept mots*".
La question qu'on aimerait analyser est "De quelle façon la vérité ou la fausseté d'une affirmation dépend de l'auto-référence ?". Par exemple nous pensons en général que des phrases vraies, si elles sont énoncées l'une après l'autre, forment une phrase vraie. Ainsi "Les poissons vivent dans l'eau" est une phrase vraie et "Mussolini est mort il y a longtemps" en est une aussi. Il s'ensuit que "Mussolini est mort il y a longtemps et les poissons vivent dans l'eau" est vraie, bien que peu sensée.
Prenons maintenant les deux phrases suivantes : "Schopenhauer était philosophe", et "Cette phrase a cinq mots", qui sont les deux vraies. Or, "Schopenhauer était un philosophe et cette phrase a cinq mots" est fausse (puisque la phrase a dix mots) ! On voit bien que l'auto-référence est dangereuse dans la construction d'un langage.

Ces problèmes peuvent heureusement être contournés si l'on emploie un langage formel, un langage qui tente d'échapper à l'auto-référence. Un langage formel ainsi fait s'appelle "calcul de propositions" et c'est un langage dans le genre qu'on a déjà décrit, tel le langage *pg*.
Il y a des variables qui représentent des choses qui peuvent être vraies ou fausses, et des manipulations du type "si x est vraie et y est vraie, alors x *et* y est aussi vraie". Une fois qu'une phrase qui respecte les règles est formée, les manipulations possibles sont nombreuses : nous pouvons mettre des "et" des "ou" des "pour chaque" des "existe" et des "non", et on peut composer toutes sortes de combinaisons de ces symboles.

La manipulation du calcul des propositions permet de décider de la cohérence d'une affirmation donnée, si on est assez habile pour construire une suite de manipulations menant d'une affirmation très embrouillée telle que "Si Dieu est tout-puissant, alors peut-il créer une pierre aussi lourde qu'il est incapable de la soulever ?" à "Dieu peut" ou bien "Dieu ne peut pas". Mais y a-t-il une manière de décider de la question de savoir si une affirmation produite par nous ne sera jamais auto-contraditoire ?

Nous allons voir par la suite que bien que le langage formel permette d'échapper à l'auto-réference directe, il est victime de l'auto-réference indirecte, la référence à des systèmes autres que lui-même, mais qui ont la même structure que lui.

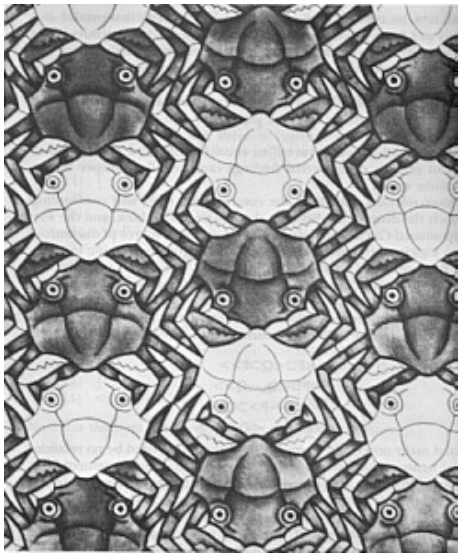


FIGURE 3 – Le code et le contexte dans "canon cancrizan" de M.C. Escher.

3.3 La théorie des nombres de Peano

Dans la suite, nous allons nous intéresser aux axiomes de Peano et à la théorie qui en découle, la théorie des nombres naturels.

Dans la théorie de Peano, il y a les symboles de la logique formelle (*et*, *ou*, *il existe*, *pour tout*, ...). De plus, nous avons le symbole 0 et le symbole S, l'addition usuelle ($+$) et la multiplication usuelle ($.$). Les axiomes sont les suivants :

1. Pour tout $a \sim (Sa = 0)$
2. Pour tout a $(a + 0 = a)$
3. Pour tout a et tout b $(a + Sb) = S(a + b)$
4. Pour tout a $(a.0) = 0$
5. Pour tout a et tout b $(a.Sb) = ((a.b) + a)$

L'interprétation de S est "successeur" ; cela signifie que $S0$ est le successeur de 0, c'est-à-dire 1. On peut vite comprendre que $SSSS0$ est 4. Finalement il suffit de compter le nombre de S pour retrouver le chiffre interprété. Les axiomes de Peano se liraient alors comme ceci :

1. Pour chaque a, il est faux que le successeur de a est 0 (le petit $\sim$ signifie "il est faux que"). Si vous préférez, cet axiome signale qu'aucun nombre n'existe avant 0.
2. Pour tout a, $a + 0$ reste égale à a.

3. Pour tout couple de nombres a et b, $a + (b + 1) = (a + b) + 1$.
4. Pour tout nombre a, $a.0 = 0$.
5. Pour tout couple de nombres a et b $a.(b + 1) = (a.b) + a$.

On fixe ainsi les règles de base de calcul sur les nombres naturels. Bien évidemment, dans le monde de Peano, il n'y a pas de nombres négatifs, ni imaginaires, factoriels ou irrationnels. Néanmoins, il y a déjà un grand nombre d'affirmations possibles ; il serait tentant alors de dire la chose suivante :
imaginons que nous avons une affirmation quelconque, et que cette affirmation puisse être traduite dans le langage de la Théorie des nombres naturels. On pourrait penser que cette affirmation est soit vraie, soit fausse, et que si elle est fausse, sa négation, l'affirmation écrite dans notre langage avec un signe de négation qui la précède, serait vraie.

Nous pourrions alors construire une "machine à théorèmes" qui engendre tous les théorèmes possibles de la Théorie des nombres par application aveugle des règles du langage. A un moment donné, on tomberait sur notre affirmation, ou sur sa négation, et on saurait alors séparer ce qui est vrai de ce qui ne l'est pas. Ce programme est très puissant, mais pour qu'il soit réalisable, il faudrait montrer que toute phrase correctement composée de la Théorie des nombres, peut être obtenue par manipulation formelle des cinq axiomes de base. De plus, il faudrait trouver un moyen de "transporter" un langage quelconque dans la théorie des nombres.

Le deuxième problème peut se résoudre de cette façon : prenons par exemple le système *pg*. Nous écrivons la lettre p comme le chiffre 11, la lettre g comme le chiffre 22 et les traits comme le chiffre 9. Un axiome du système *pg* aura alors l'air suivant :

9119229

999999119229999999

et nous pourrions chercher (bon courage !) une interprétation et des règles dans cette nouvelle traduction. L'atout qu'on vient de gagner, c'est que nous pouvons traduire ces chiffres dans le langage de la théorie des nombres : nous pouvons écrire $SSS...SSSS0$ pour un chiffre quelconque, il suffit d'imprimer assez de S. Et bien évidemment, cela permet de traduire n'importe quel langage formel dans le langage de la théorie des nombres.

D'une certaine manière cela revient à numéroter toutes les affirmations possibles avec un chiffre personnel. Ce nombre est parfois immense, mais il existe ; on l'appellera "nombre de Gödel".
Or les lecteurs attentifs, n'auront pas manqué de remarquer que la Théorie des nombres peut aussi être traduite en chiffres (par exemple en écrivant "S" comme"111", "pour tout" comme "222", "il existe" comme "232", ...). Le

système de la Théorie des nombres peut non seulement associer un nombre de Gödel à n'importe quelle affirmation d'un langage formel, mais il peut aussi numéroter ses propres affirmations. Cela revient à dire que la Théorie des nombres est aussi son propre métalangage, et que certaines affirmations de la théorie des nombres sont aussi le codage d'autres affirmations de la théorie des nombres. En quelque sorte, il se peut qu'une affirmation de la Théorie des nombres en contient une autre, qui est incluse dans elle de façon passive.

Maintenant c'est le moment de jouer un petit tour de "magie logique". Nous construisons dans le langage de la Théorie des nombres l'affirmation G, qui est auto-réferentielle. G est une affirmation de la Théorie des nombres, dont une des significations passives dit "G n'est pas un théorème de la Théorie des nombres". La question sera de savoir si elle est bien un théorème de la Théorie des nombres, c'est-à-dire s'il existe une preuve formelle qui permet de la ranger parmi les affirmations qui sont vraies ou fausses. Malheureusement si l'affirmation est vraie, elle est une affirmation de la Théorie des nombres, qui prétend ne pas l'être. C'est une contradiction. Si elle est fausse, elle n'est pas une affirmation de la Théorie des nombres, qui prétend l'être (puisqu'elle dit ne pas l'être et elle est fausse), ce qui est aussi contradictoire. C'est une trappe sans sortie, qui nous mène à la conclusion suivante :

il y a des phrases bien formées de la théorie des nombres pour lesquelles il n'est pas possible d'exhiber une preuve, c'est-à-dire qu'il est impossible de décider si elles sont vraies ou si elles sont fausses.

On peut donc conclure que la théorie des nombres est incomplète.

4 Niveaux de description

On aborde maintenant un nouvel argument : on s'intéressera aux niveaux de description possibles d'un problème donné. On peut citer des exemples nombreux, mais celui qu'on va suivre, en ligne avec Hofstadter, est la stratification du langage des ordinateurs. A la base, l'ordinateur contient des suites de 1 et de 0 ; ce sont des bits. Une chaîne de bits constitue un mot. Un mot peut représenter un peu n'importe quoi. Ça pourrait être un chiffre, un pixel sur l'écran, une lettre dans un texte. Il peut aussi être une instruction : un exemple d'instruction pourrait être ADDITIONNER(x,y), suivie de l'endroit où chercher deux mots, qui seront donc interprétés comme des chiffres et additionnés. Mais on pourrait aussi avoir des instructions telles que IMPRIMER(x), qui interprète donc x comme une chaîne de lettres. En général, le niveau des instructions est supérieur à celui des bits, et construit grâce à lui. Mais la signification des

0 et des 1 n'existe pas. Ce n'est qu'au niveau supérieur que les instructions ont un sens. C'est ce qu'on entend en général lorsqu'on dit qu'il y a réduction verticale (le niveau supérieur est construit grâce à celui d'en-dessous), mais indépendance horizontale (il n'est pas possible de comprendre les instructions d'un niveau en regardant ses composantes).

A un niveau plus élevé, il pourrait y avoir un niveau de programmation convivial. Celui-ci devrait pouvoir écrire des instructions plus complexes, qui fonctionnent grâce aux instructions de base. Il s'agit donc d'un langage qui comprend le langage d'en-dessous.

Un programme en langage machine pourrait être quelque chose comme "prend l'adresse X (qui contient le chiffre un) et additionne avec X", ce qui revient à faire 1+1=2. Mais si nous voulons un niveau abstrait d'addition, on aimerait quelque chose de plus complexe, tel que z+y=(répète z fois l'instruction additionne(x,x) puis repète y fois additionne(x,x), puis additionne(z,y)). Nous n'avons pas envie de voir tout cela : on peut alors construire des procédures plus complexes, qui "cachent" les structures simples et répétitives, et qui sautent à un niveau d'abstraction plus élevé. C'est comme cela qu'on gagne un niveau. Nous pouvons facilement imaginer qu'une fois qu'on sait additionner deux nombres, il est très simple d'apprendre à les multiplier (il suffit d'appeler récursivement l'addition), pour ensuite passer aux calculs plus compliqués. Nous n'avons évidemment aucune envie de refaire continuellement des opérations de base très simples et très ennuyeuses. Nous construisons donc un coquillage (imaginaire) autour d'un groupe d'instructions qui font quelque chose de structuré. Ensuite nous appelons ce coquillage par un nom, telle que "binôme de Newton", et nous nous contentons d'appeler "binôme de Newton", à chaque fois qu'il nous est utile. C'est certainement plus simple (et plus compréhensible) que de demander "effectue l'opération 11010010100111010011101011010101011" encore que "effectue l'opération" est aussi une suite de bits.

C'est un peu pour cela que les informaticiens rigolent à la fameuse blague de l'erreur fatale

> WinErr 16547 :LPT1 introuvable. Effectuer backup (Papier et crayon SYS)

Un backup "papier crayon" à base de 1 et de 0 n'a simplement aucune signification.

En effet, le problème du niveau de langage ne se pose pas lorsque tout se passe de façon correcte. Nous avons un aperçu des difficultés seulement en cas d'erreur : une des tâches difficiles de la programmation est de rendre compréhensibles les problèmes et les erreurs conséquentes, engendrés par une procédure incorrecte. Une annonce telle que "exécution du programme interrompue lors de la lecture du registre 110100100111101" n'est d'aucune utilité

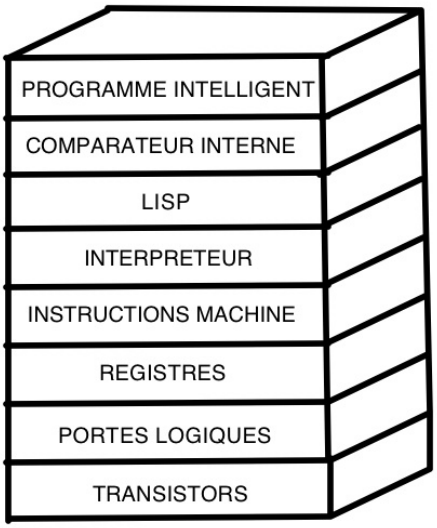


FIGURE 4 – Les niveaux d'un ordinateur "intelligent"

au programmeur.

Un ordinateur est généralement rigide vis-à-vis de sa programmation : si vous tapez "IMPRAMER" au lieu de "IMPRIMER", l'ordinateur va engendrer des erreurs et arrêter ses procédures. L'une des difficultés dans la conception d'une intelligence artificielle sera certainement de dépasser cette rigidité par la construction d'un niveau assez haut, capable d'accepter une certaine souplesse. Bien évidemment la souplesse a un prix : en mémoire et en précision. Un ordinateur qui interprète, aussi bien qu'un humain qui interprète, peut mal comprendre et partir dans une direction totalement erronée. D'un autre côté un niveau interprétatif (il y a de nos jours des programmes qui interprètent : par exemple *Copycat*, dont Hofstadter est l'un des concepteurs), est interprétatif en tant que tel seulement au niveau supérieur. Les niveaux inférieurs restent de la programmation rigide et câblée. Hostadter propose la métaphore suivante : ce dont un ordinateur est capable (et "conscient") de faire à un niveau donné, n'est pas forcément vrai à un autre niveau : par exemple le programme PARRY, qui simule un paranoïaque en vous répondant par des phrases répertoriées, n'est absolument pas conscient de ce qui se passe au niveau inférieur. Ainsi s'il ralentit à un moment, à cause de la charge en utilisateurs, vous pouvez regarder facilement le nombre de personnes qui l'utilisent, et savoir ce qui l'empêche d'avancer rapidement. PARRY n'en sait pourtant rien du tout. C'est un peu comme si on vous demandait : pourquoi construisez-vous autant de globules rouges ? Vous n'en savez rien : et pourtant c'est bien vous qui accomplissez cette tâche.

4.1 Comportement émergent et symboles

On va maintenant s'occuper plus en détail de la représentation souple dans un ordinateur ainsi que de la création et de la reconnaissance des symboles. La métaphore proposée par l'auteur, Achille et la tortue étant de nouveau appelés à l'action, est celle des fourmis. Une fourmi est "stupide", mais un groupe organisé de fourmis ne l'est pas. Achille et la tortue écoutent l'explication d'un expert du sujet (un fourmiller, dont la travail consiste à soigner les troubles psychiques des colonies de fourmis par ablation chirurgicale). Le fourmiller raconte qu'il est en très bon rapport avec Madame de Montfourmi, la "conscience" de la colonie, bien qu'il soit craint par chacune des fourmis.

Dans un nid, la structure de base est constituée par les fourmis, chacune d'entre elles communiquant avec ses voisins de façon juste assez forte pour ne pas se disperser au hasard.
Le mouvement d'une fourmi seule n'a pas grand-chose d'organisé, elle bouge grosso modo au hasard, mais elle communique assez avec ses voisines pour que des fois des comportements "en avalanche" commencent. Si par pur hasard, à un moment donné, certaines fourmis ont la tendance à se comporter de manière homogène, alors ces fourmis engendreront un comportement chez les proches voisins, en émettant un signal collectif plus fort. Imaginons qu'une tâche particulière doive être accomplie : un groupe de fourmis se crée de façon spontanée, par fluctuation statistique, et se met au travail. Des groupes de fourmis se forment sans cesse ; seulement, quand elles ont une tâche à accomplir, elles ne se défont pas. Cela marche, car si une fourmi trouve "quelque chose à faire" (par exemple récolte de la nourriture), elle envoie un signal proportionnel à son enthousiasme. Si la quantité est assez grande, il se forme un gros groupe, et l'effet avalanche se déclenche.

Le groupe traverse alors la colonie : c'est un signal qui vient de se créer et qui se déplace. Il ne faut surtout pas penser au groupe fourmi par fourmi. En effet, des fourmis sont tout le temps en train d'entrer dans le groupe et de le quitter. Le signal se conserve, mais les fourmis qui le composent pourraient très bien à la fin n'avoir aucun élément en commun avec les fourmis qui le composaient au début.

Le signal se propage : tant que l'extérieur transmet d'autres signaux (par exemple soin des petits, nettoyage) le signal continue d'avancer. Au moment où ce qu'un signal peut offrir correspond au besoin de son environnement, il se défait. Il faut remarquer qu'aucune fourmi n'est consciente du signal qu'elle compose. Elle n'en sait rien, et elle n'en a pas besoin ! Le signal semble être

intentionnel au niveau des autres signaux, mais pas à celui de la fourmi.

Tout cela marche car il y a un environnement (des castes de fourmis) qui fait du sens : la signification a été forgée par des millénaires d'évolution. C'est ainsi que le fourmiller peut "parler" avec les signaux de Mme de Montfourmi, alors qu'il est l'ennemi juré de chacune d'elles.

La métaphore est celle du cerveau humain dont les fourmis répresentent les neurones. Bien évidemment, le cerveau n'est pas une colonie de fourmis : le niveau d'implementation, le niveau de base, est très différent. Ce qui compte c'est que des signaux organisés (au moins en l'apparence) peuvent circuler.

Les "fourmis dans le cerveau", les neurones, fonctionnent de manière analogue aux fourmis : leur possibilité est de s'exciter ou pas, et cela jusqu'à 1000 fois par seconde. La décision de s'exciter est prise en fonction d'une valeur de seuil : si les entrées dépassent ce seuil, le neurone s'active.

Le cerveau possède des zones consacrées à des tâches particulières : par exemple l'aire visuelle du cortex s'occupe de la vision.
S'il est possible de repérer des "zones par fonction", il n'est tout de même pas possible de trouver des "zones par images". Il n'y a pas, par exemple, une cellule ou un groupe de cellules fixement localisées qui s'active, parce que votre grand-mère vient de passer dans votre cône visuel. A un niveau supérieur de fonctionnement (par analogie avec les niveaux d'un ordinateur), il doit y avoir des symboles, qui sont exécutés par des groupes de neurones.
Plusieurs questions se posent à cet égard. Notamment :

1. Quelle forme devrait avoir un groupe de neurones qui constitue un symbole ? Est-elle compacte ou ressemble-t-elle à un filament (ou une toile ou autre chose) qui s'étend dans de nombreuses zones du cerveau ?
2. Un seul neurone peut appartenir à plusieurs symboles ? A combien ? Et combien de neurones faut-il pour un symbole ?
3. Les symboles sont-ils les mêmes (ou presque) dans tout cerveau ?
4. Deux mêmes symboles sont-ils situés au même endroit dans deux cerveaux différents ?
5. Le chevauchement de complexes de neurones formant plusieurs symboles est-il le même dans tous les cerveaux ?

On pourrait aussi se demander si un symbole est enclenché par l'activation d'une même zone neuronale, ou si plusieurs zones codent pour un même symbole. Les difficultés s'ajoutent si nous revenons à notre discours du contexte et du langage. Si le symbole est le langage, il est assez clair qu'il ne peut être

probablement pas activé tout seul, mais que son activation déclenche l'activation d'autres symboles.

Nous ne proposons pas de réponse (Hofstadter donne quelques pistes), mais nous aimerions souligner que l'activation des symboles ne doit pas être trop rigide.
Nous sommes capables d'imaginer l'absurde : si je vous demande de penser à la mer, il est probable que vous activiez l'image d'un poisson. Mais si je vous demande d'activer l'image d'une girafe qui nage le long de la fosse des Mariannes, vous pouvez le faire, bien que cela ait l'air drôle. Il n'y a rien qui vous empêche d'imaginer l'absurde et l'imprevu, puisque c'est une des nos prérogatives, un des nos atouts, de savoir improviser, d'être souples au niveau des symboles. Une guêpe *Sphex* qui creuse un trou pour y pondre ses oeufs, part chasser un grillon, puis elle le pose, paralysé par son venin, à côté du trou, et elle rentre pour contrôler qu'il n'y ait pas d'intrus. Enfin, elle sort, elle pousse le grillon dans le trou, elle le bouche et elle part. Si un expérimentateur déplace le grillon de quelques centimètres, la guêpe sorte du trou, repositionne le grillon et recommence l'inspection. Vous pouvez répéter à plusieurs reprises l'expérience : la guêpe sortira, repositionnera le grillon, et recommencera. Jamais une idée nouvelle lui viendra à l'esprit : la guêpe n'a justement pas de souplesse dans l'activation de ses symboles.

En ce qui concerne les ressemblances entre cerveaux, on peut penser que des personnes assez similaires (par culture, par langue, par consanguinité) auront des symboles assez similaires.

Hofstadter propose la métaphore suivante : imaginez qu'on vous donne une carte de la CEE, avec toutes les caractéristiques géologiques, mais sans les noms des pays, les autoroutes, les aéroports, les parcs nationaux... L'Europe sans l'homme. Maintenant on vous demande de la compléter avec les bonnes données. Il est vraisemblable que vous arriviez assez bien à remplir la partie autour de votre région d'origine, ainsi que celles que vous avez visitées dans le passé. Ainsi il sera très probable que je sache situer Lausanne, Lugano et Berne, que je connaisse à peu près le nord de l'Italie, la France et l'ouest de l'Allemagne, mais je vais être très approximatif sur Rome, la Sardaigne et l'Espagne, et je vais probablement écrire n'importe quoi sur l'Ukraine et la Pologne. Appelons cette version de la CEE, une EEC : imaginons maintenant de pouvoir voyager dans cette EEC imaginaire. Pour qu'on ne se perde pas, on vous donne une carte de la CEE ordinaire. Certains accidents peuvent se produire : par exemple dans la visite de ma EEC, la carte sera correcte sur la forme de la Sardaigne, mais il y a des chances que l'île se retrouve surpeuplée, et que Sassari et Cagliari soient marquées à des kilomètres de distance de leur

site véritable [4].

L'exemple est là pour nous indiquer qu'une partie des nos symboles sont probablement universels ; il est par exemple peu probable que Rome soit absente de votre carte de la EEC, même si certains, qui sont comme moi très peu doués en géographie, pourraient la mettre à la place de Naples ou même de Milan.

Or des aberrations, des erreurs, et mêmes des absurdités peuvent habiter nos EEC : ce sont des convictions erronées. Au fond, nos rêves ressemblent beaucoup à ce genre de mondes imaginaires. Le langage et la culture ont une influence importante, maistous les hommes ont des points en commun !

Finalement, Hofstadter propose que le "moi" peut être considéré comme un symbole : un symbole qui peut jouer le rôle de ce qu'on appelle souvent "âme" ou "esprit", et qui loin d'être "magique", appartient au monde des objets sensibles.

4.2 La transcendance de soi : un mythe contemporain

Un des thèmes les plus débattus en philosophie est la transcendance de soi. Il s'agit de l'idée selon laquelle un individu peut sortir de soi-même, c'est-à-dire sortir du système qui est le siège de sa propre existence. Bien sûr, un individu peut parler de soi-même : il devient alors son méta-soi. Si c'est à cela que revient la transcendance, alors elle ne diffère guère du langage de la théorie des nombres. Le langage aussi peut parler de lui-même et devenir son propre méta-language. Or, ce méta-langage engendre des propositions indécidables, comme on l'a vu, et nécessite du méta-(méta-langage). Puisqu'on est forcé d'avancer infiniment, nous savons qu'il restera toujours des propositions indécidables. Cela veut dire qu'un langage, aussi bien qu'un individu, peut parler de lui-même, mais il ne peut absolument pas *sortir de lui-même*. Les conséquences intéressantes sont de deux types : d'un côté nous rejetons comme inconsistante tout approche mystique vouée à sortir du système : c'est n'est qu'une absurdité. De l'autre, nous pouvons voir qu'un système peut parler de soi en étant son méta-langage, son méta-(métalangage) et son méta-(méta-(méta-(méta-(...(méta-langage))))), mais nous sommes assez rapidement confrontés à nos limites de computation (ou d'abstraction si on veut recourir à un niveau de description plus élevé). Si un ordinateur peut accéder à une puissance proche de celle du cerveau, et à des récurrences (des images de soi) aussi profondes que nous, il n'y a aucune raison pour que ces capacités ne soient pas en mesure d'engendrer une intelligence artificielle.

4. Essayez de dessiner un endroit que vous pensez connaître à peu près, et allez ensuite le comparer avec la vraie carte géographique.

Une récursion assez compliquée peut donc donner lieu à un système intelligent.

La métaphore suivante proposée par Hofstadter est celle d'un crabe, qui est capable de jouer à la flûte n'importe quel théorème des mathématiques. Il prend un bout d'écriture dans le langage de la théorie des nombres et il le joue, en obtenant des mélodies. Ces compositions sont belles ou laides, en fonction du fait que les assertions de la théorie des nombres sont vraies ou fausses, mais le crabe n'en sait rien. Il comprend la beauté de façon intuitive, et il n'a pas la moindre idée des mathématiques. L'idée derrière cette métaphore est d'introduire la possibilité d'une connaissance intuitive, c'est-à-dire de la connaissance d'une réponse, sans savoir comment cette réponse a été générée. Ce discours aboutit à la thèse de Church-Turing, qui affirme que

> Supposons qu'un être sensible soit capable de trier les nombres en deux classes (par exemple les pairs et les impairs, les premiers et le non-premiers), que sa méthode ait toujours une réponse dans un temps fini, que la réponse soit toujours la même pour un même nombre. DE PLUS CETTE MÉTHODE PEUT ÊTRE COMMUNIQUÉE À L'AIDE DU LANGAGE. Alors il existe un programme récursif capable d'accomplir la même tâche.

On a ici la version faible de la thèse : une version plus forte s'obtient en enlevant la phrase en lettres MAJUSCULES.

Hofstadter défend plutôt la thèse version forte : il analyse des cas connus de capacités mathématiques hors du commun (le mathématicien Ramanujan, certains cas de "savants idiots") en mettant en évidence que loin d'être magique, leur méthode repose tout simplement sur une puissance de calcul hors du commun. Preuve en est que ces génies de mathématiques ont besoin d'un temps très court afin d'effectuer des calculs complexes, mais que leur temps de calcul s'allonge avec la complexité de la tâche. Ce sont des personnes plus douées que la moyenne, mais pas différemment douées. Tout cela nous mène à la réflexion suivante : si tous les processus cérébraux dérivent d'un substrat calculable, alors peu importe que le substrat soit neural ou électronique. L'intelligence peut surgir des deux : elle repose sur la création des symboles, sur l'organisation de haut niveau, plutôt que sur le type d'implémentation.Toute capacité peut s'écrire comme un programme, mais le hardware peut varier sans conséquences.

Il y maintenant quelque malentendu à discuter : il s'agit de la thèse de T. Roszak avec ses versions faibles.

> Les ordinateurs sont ridicules ainsi que, de façon générale, toute discipline scientifique

En général, cette affirmation découle de l'idée que les nombres et l'exactitude sont une menace pour les valeurs humaines. Notamment en ce qui concerne :

1. l'irrationalité de l'homme contre la rationalité de la machine.
2. l'existence de la beauté, une qualité insaisissable pour un ordinateur.
3. l'âme, qui distingue l'homme de toute intelligence artificielle imaginable.
4. et tout dernièrement les *qualia*, qui sont en somme une thèse de Church-Turing de l'âme version faible.

(La quatrième thèse est un ajout personnel.)

En ce qui concerne l'irrationalité, le problème se situe au niveau d'un manque de compréhension des ordinateurs ainsi que d'une confusion dans les niveaux de description. C'est certainement vrai qu'un ordinateur ne se trompe pas dans ses procédures de base. Pareillement, un neurone ne se trompe pas dans son déclenchement une fois qu'il dépasse le niveau de seuil. Par contre, l'activité neuronale peut engendrer un comportement irrationnel. Rien n'empêche, pareillement, d'écrire un programme qui imprime des lettres au hasard, ou des phrases choisies parmi un gros lot qui sont parfois vraies, parfois fausses, sans raison particulière. C'est là un comportement de haut niveau totalement irrationnel.

En ce qui concerne la beauté des choses il faut rappeler que nous sommes en train de parler de capacités de décision, dans la thèse de Turing, qui aboutissent à un résultat dans un temps fini. Mais nous savons, grâce au théorème de Gödel, qu'il y a des affirmations de la théorie des nombres pour lesquelles on ne peut pas savoir si le temps de calcul est fini. Il conviendra alors de séparer les propriétés décidables en temps fini (qu'on appellera syntaxiques), de tests qui ne peuvent pas aboutir dans un temps fini (qu'on appellera sémantiques).

La beauté n'est certainement pas syntaxique : nous pourrions caractériser nos propriétés syntaxiques comme celles qui possèdent une signification intrinsèque, alors que les propriétés sémantiques n'ont de signification que si elles sont plongées dans un ensemble de relations (potentiellement infinies) avec d'autres objets. Ces relations sont prises en compte en passant à des niveaux descriptifs plus élevés, et sont donc résistantes à la compréhension. Néanmoins, l'écriture en langage formel est possible et fait du sens, et il est donc imaginable de construire tout cela sur une machine.

5 Intelligence artificielle

5.1 Test de Turing

Les machines peuvent-elles penser ? C'est la question proposée par Turing, à laquelle on va essayer de répondre. Il propose aussi un test, capable de décider si une machine pense, ou si elle ne pense pas. Le test se joue à trois, dans la version originale de Turing : un homme A, une femme B, et un interrogateur, dont le sexe n'a pas d'importance.
L'interrogateur interagit à l'aide d'un clavier avec A et B, et il ne le voit pas. Il doit réussir à deviner lequel des deux est l'homme et lequel est la femme. L'interrogateur peut proposer toutes sortes de questions, et en principe A essaye de le tromper, alors que B essaye de l'aider. La meilleure stratégie consiste (pour B) à ne pas mentir.

Maintenant une machine prend la place de A dans le jeu. L'interrogateur se trompera-t-il dans cette version dans l'identification de la femme, aussi souvent que dans la version standard ? Autrement dit : une machine peut-elle penser ?

Un ordinateur, on pourrait croire, est fort doué dans certains domaines, et pas du tout dans d'autres. Imaginez alors de poser une question concernant un calcul compliqué : vous remarquez que la réponse prend un temps long et qu'elle est fausse. Est-ce que cela permet de conclure quelque chose sur la nature du répondant ? S'il s'agit d'un humain, on peut croire qu'il a fait une faute de calcul, mais s'il s'agit d'une IA, plusieurs raisons pourraient expliquer sa défaillance. Il pourrait s'agir d'une erreur, d'un malfonctionnement, d'une astuce du programmeur pour nous tromper dans le test, de quelque type de humour robotique. Une vraie IA pourrait vous donner une autre réponse si vous lui reposez la même question plus tard.

Vous seriez étonnés de savoir qu'il existe des machines qui ont passé un test de Turing *version faible*. C'est-à-dire que des utilisateurs qui ne savaient pas qu'ils communiquaient avec des machines ont été trompés par celles-ci. Or le type de machines en question sont des ordinateurs qui ont un gros répertoire de réponses, et qui choisissent celle qui semble être la meilleure en fonction de la question. Il y a notamment des programmes capables de simuler le discours d'un paranoïaque (PARRY), et d'un psychiatre utilisant une "thérapie non directive". Aucune machine n'a tout de même réussi le test de Turing alors que l'interlocuteur était au courant d'effectuer un test sur l'IA.

On peut notamment remarquer que les programmes trompeurs ne sont pas intelligents, selon un consensus unanime : en général, ils ne comprennent pas grand'chose à ce qu'ils disent, et on peut le remarquer en les faisant jouer l'un contre l'autre.
Une question plus intéressante se pose au niveau de l'identification d'une ori-

ginalité dans la réflexion d'une machine. Par exemple, un ordinateur qui joue aux echecs mieux qu'un humain, est-il original dans le jeu ? Remarquez qu'en général un ordinateur ne calcule pas toutes les combinaisons possibles depuis une position donnée : lorsqu'on vous dit qu'un ordinateur prévoit mille coups, cela veut dire qu'il développe certaines stratégies jusqu'à milles coups. Ce n'est d'ailleurs pas très utile en soi ; ce qui est important, c'est de choisir les bonnes directions à calculer, et pour faire cela l'ordinateur possède des "idées générales" sur la valeur de pièces, la position, le contrôle du centre...
A l'époque de la rédaction de GEB, Hofstadter ne savait pas que les ordinateurs réussiraient à battre l'homme ; il s'étonnait que cela soit possible au jeu de dames. Les discussions qu'on peut faire sont néanmoins semblables : notamment, on peut se demander si le mérite de la victoire de l'ordinateur est dû à lui-même ou à son programmeur (à l'auteur de la victoire ou à son méta-auteur). Le discours se complique si l'on considère qu'un ordinateur peut battre son propre programmeur aux echecs.

En général, la réponse de Hofstadter est la suivante : si vous pouvez clairement identifier, dans le code de l'ordinateur, où se situe la décision de faire ce choix, alors il n'y a pas d'originalité dans la machine (il n'y-en a que dans le programmeur). Mais s'il n'est plus possible de le faire, c'est-à-dire que la machine possède des symboles de haut niveau, alors elle est vraiment originale.

Ce genre de problème (la construction d'un symbole abstrait) peut se voir tout aussi bien dans certains animaux. Par exemple, un chien qui voit un os, alors qu'un filet métallique le sépare de son but, tendra à courir jusqu'au filet et aboyer. Si vous creusez un trou quelques mètres à côté, certains chiens arriveront à comprendre la stratégie gagnante, alors que d'autres pas. Le concept de "manger l'os", qui se réduit au problème du plus court chemin possible, a été abstrait à un niveau supérieur. De tels problèmes se posent sans cesse devant nous, et nous sommes capables de les résoudre : peut-être qu'une machine intelligente doit être excellente dans l'abstraction, la capacité de se regarder en train d'effectuer la tâche, pour enfin revenir l'accomplir en ayant une idée meilleure du travail à effectuer. C'est un peu la différence entre la guêpe *Sphex*, extrêmement efficace dans sa tâche, mais totalement aveugle au niveau supérieur d'abstraction, et l'homme qui regarde la guêpe *Sphex*.

Un exemple assez drôle de simulation de langage a été obtenu par Hofstadter lui-même, grâce à un programme qui connaît la grammaire. Il nous propose de décider parmi neuf phrases, dont six ont été engendrées par l'ordinateur et trois sont des propositions tirés de discours réels. J'en donne trois exemples :

– L'émission spontanée de propos peut être considérée comme la substitution réciproque d'un matériau sémiotique (doublage) à un produit dialogique sémiotique au cours d'une réflexion dynamique.

– Une attitude sera souvent adoptée par les serfs d'une nation déchirée par des conflits.
– Par ailleurs, les Prix Nobel seront atteints. Du même coup, en dépit de la conséquence, les Prix Nobel qui seront atteints seront parfois atteints par une femme.

Vous avez trouvé l'intrus ? La réponse est au fond de la page. [5]

5.2 Intelligence artificielle : avenir

Il y a une capacité intéressante qui est propre aux individus doués d'intelligence (nous), qui est la production de variantes mentales de ce qui se passe vraiment. C'est la capacité d'imaginer ce qui aurait pu nous arriver si. Il apparaît que cette tâche est spécialement complexe, notamment parce que seulement certaines variations font véritablement du sens. Nous pouvons tout à fait imaginer ce qui aurait donné le dernier championnat de football si l'arbitre avait sifflé ce penalty dans le match qui a décidé le classement. Nous pouvons aussi imaginer ce que serait devenue notre vie si on avait épousé cette fille qui nous plaisait il y a maintenant dix ans, mais cela paraît bien plus difficile. Et il n'est pas tout à fait question de la quantité de choses qu'on change, ou de la distance temporelle que nous sépare des événements : ce discours paradoxal nous en donne l'idée :

– Mon oncle a failli être président des Etats-Unis.
– Vraiment ?
– Oui, il a été capitaine de la vedette lance-torpille 108 ! (J.F.K. l'a été de la vedette lance-torpille 109).

En fait, la similitude parmi deux concepts, permet de les ranger dans des classes, et de construire des symboles. Mais on ne peut pas ranger n'importe quel objet dans n'importe quelle classe ! En fait, la construction des classes est plutôt délicate.

La reconnaissance des patterns, ainsi que la capacité de choisir ce qui est important parmi les similitudes entre des données, est l'un des traits importants qui caractérisent l'intelligence. Un problème bien connu est celui de *pattern recognition*, proposé par Bongard (voir figure 5 et 6). On y trouve des formes et des dessins, rangés dans deux groupes de six cases. La question qu'on est amené à résoudre est : "En quoi les deux groupes de six sont différents ?" [6] [7]

Dernièrement, Hofstadter s'est intéressé avec Melanie Mitchell au problème de la reconnaissance des formes grâce au programme *Copycat*, qui est capable de trouver des analogies parmi des groupes d'objets. Dans cette approche, on

5. C'est la première affirmation qui a été tirée de la revue *Art-Language*, les autres sont des phrases composées par l'ordinateur. L'auteur de cette recension se demande par ailleurs si cela prouve quelque chose au niveau de la valeur du programme, ou plutôt de l'absence

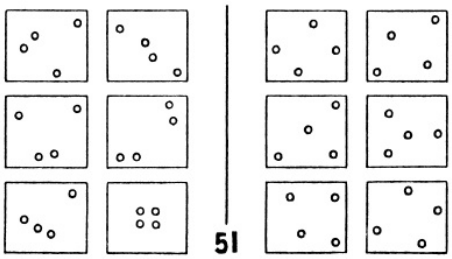


FIGURE 5 – Pattern recognition : en quoi les figures de droite et celles de gauches sont différentes ?

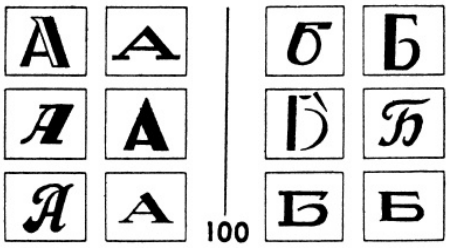


FIGURE 6 – Pattern recognition : en quoi les figures de droite et celles de gauches sont différentes ?

reconnaît l'idée sous-jacente de symboles de haut niveau, qui est nécessaire afin d'interpréter des problèmes du type "Bongard".

On est en train, dans un certain sens, de mécaniser la créativité : on pourrait objecter que c'est là une absurdité. La créativité est justement l'opposé du mécanique. En effet, ceci est vrai seulement au niveau (élevé) de la créativité. La programmabilité est cachée à des niveaux inférieurs, en bas de la chaîne de construction. Il se peut, comme certains le pensent, qu'un degré d'aléatoire soit nécessaire (et utile) dans la créativité, mais ce n'est pas au détriment de la mécanicité (dans le sens de programmabilité).

Le niveau le plus difficile à atteindre, en ce qui concerne la reconnaissance des concepts, est probablement le langage naturel. Les symboles sont très nombreux et de type sémantique ; la signification d'un terme est indissoluble du contexte dans lequel il est émis, c'est-à-dire du langage (ou d'une partie du langage) lui-même. Dernièrement, un informaticien qui en ce moment travaille dans l'écriture de programmes qui savent parler, disait qu'il est possible, quoi que difficile, de créer un programme capable d'écrire correctement en français, qui respecte les règles grammaticales et syntaxiques. Il ne comprend néanmoins pas ce qu'il dit : "Antoine mange une tarte" et "une tarte mange Antoine" sont parfaitement équivalents du point de vue de ce programme, puisqu'ils sont bien conçus grammaticalement.

Hofstadter termine le chapitre par une liste de questions/réponses qui sont intéressantes à la fois par leur portée (certaines sont toujours des questions ouvertes) et par leur intérêt historique.

5.2.1 Dix questions et dix réponses

Voici donc les questions : les réponses proposées par Hofstadter sont résumées dans la forme "DH :", alors que les réflexions du critique sont données dans la forme "TM :". Les réponses données par Hofstadter sont un condensé de l'original ; elles sont sujettes donc à interprétation dans la mesure du choix que j'ai pu faire sur ce que j'ai jugé être plus significatif en résumant ses positions. Ca ne correspond d'ailleurs pas forcement aux idées actuelles de Hoftsadter, mais uniquement à son point de vue en 1979.

1. Un ordinateur pourra-t-il écrire de la belle musique ?
   DH : Oui, mais pas avant longtemps. Il faudra qu'une IA éprouve des émotions avant de pouvoir écrire de la musique capable de susciter des

de contenu de certains articles de linguistique.

6. Faible entropie à gauche, grande entropie à droite.

7. Lettre A contre lettre B : cette tache peut vous sembler plus simple que la 51, mais ce n'est pas forcément le cas pour une machine.

émotions. La musique est un langage qu'on peut comprendre seulement si on a éprouvé des émotions, et si on a vécu les sensations qu'on veut transmettre.
TM : C'était le point en 1979, à l'époque de la publication de GEB. Entre temps nous avons vécu l'arrivée de la musique *techno*, qui est partiellement écrite par ordinateur, et que l'homme n'est souvent pas capable d'exécuter. La tâche revient à l'ordinateur. Je pense que DH à partiellement raison puisqu'il fait directement référence à la musique classique. Néanmoins la musique *techno*, qui est certainement moins émotive (ou du moins possède une émotivité plus primitive) est là. Je pense que cela nous indique quelque chose d'important : la technologie (et l'IA quand on y arrivera) influencent fortement notre esthétique et notre conception des émotions. Nous habitons un monde dans lequel les bruits mécaniques, rythmés et réguliers ont une place de plus en plus importante, et ceci se répercute sur notre conception de ce qu'est la musique. On peut s'attendre à ce que le jour où l'on parviendra à l'IA, elle aura un sens esthétique différent du notre, et qu'un consensus élargi émergera alors au niveau de ce que sont l'harmonie et la beauté.

2. Les émotions seront-elles expressément programmées dans une machine ?
   DH : Non, c'est ridicule. Les émotions sont des comportements émergents, qui surgissent de l'interaction complexe entre nos pensées. L'IA fera forcément de même.
   TM : Il n'y a rien à ajouter. C'est un point de vue profond et réfléchi, surtout en 1979.

3. Un ordinateur pensant pourra-t-il additionner rapidement ?
   DH : Pas forcément. Nous sommes composés de sous-systèmes qui calculent de façon très précise, mais nous ne sommes par forcément de très bons calculateurs. Pour l'IA, le même raisonnement s'applique.
   TM : Je partage ce point de vue, mais je trouve que cela soulève une autre question. En effet, nous imaginons l'IA comme une sorte de surhumain, un individu surdoué dans ce que les machines savent faire d'habitude et qui de plus est capable d'éprouver des émotions. C'est probablement pour cela que nous trouvons séduisante l'IA : c'est l'idée d'un ordinateur qui possède à la fois la créativité humaine et la puissance de calcul d'un ordinateur. C'est tout à fait vraisemblable que la vraie IA n'aura pas forcément ces compétences ; la question est alors de savoir ce que nous aimerions qu'une IA sache faire.

4. Des programmes d'échecs seront capables de battre n'importe quel humain ?
   DH : Non. Il existera peut-être des IA capables de le faire, mais ce sera

forcément des programmes capables de faire aussi d'autres choses. Ils devront être aussi intelligents que nous, à tout niveau, pour y parvenir.
TM : DH à tout simplement tort. Le meilleur joueur au monde c'est un ordinateur, et il n'est de loin pas une IA intelligente. Cette question est intéressante surtout pour sa portée historique. A moins de trente ans de la publication GEB est démentie de façon évidente ; le *partisan* de l'IA, quelqu'un dont la position pouvait passer pour extrême à l'époque, n'a pas sur-estimé, il a sous-estimé le développement des ordinateurs. Je trouve que c'est très indicatif.

5. Sera-t-il possible d'intervenir sur une IA en bricolant le "bas-niveau" pour lui ajouter plus de créativité, d'amour pour le football, ou de sens artistique ?
DH : Non. Ce serait pareil que penser d'intervenir dans un cerveau pour y ajouter un côté "bon cuisinier" ou "animaliste". Ca ne marche pas.
TM : Totalement d'accord. En fait, cette question est proche de celle sur l'addition.

6. Pourra-t-on régler l'IA pour qu'elle se comporte comme vous, ou comme moi, ou à mi-chemin entre nous deux ?
DH : Non. C'est de nouveau un problème de niveaux de complexité. Une IA aura un comportement émergent, non prévu, et non réglable.
TM : Je ne suis pas sûr. Je crois que DH à raison si on nous demande de changer une IA déjà existante et bien formée. Mais il est tout à fait possible d'obtenir un humain qui ressemble à moi : c'est mon père, mon fils, et dans une certaine mesure mon ami. Des animaux domestiques aussi. Cela fonctionne parce que nous sommes en interaction avec l'environnement et entre nous. Une IA devrait probablement apprendre (c'est une des directions de développement, pensons aux algorithmes génétiques) et il serait alors raisonnable de s'attendre à que leur "père" ait une influence importante sur leur comportement.

7. Une IA aura un "coeur" ou ne sera-t-elle qu'une addition de "boucles stupides et de séquences d'opérations triviales ?"
DH : Au niveau de base il n'y aura que des boucles triviales. Mais une vraie IA sera si profonde qu'en la regardant "de loin" elle paraîtra avoir un "coeur".
TM : Cette question est délicate, et elle a à faire avec deux problèmes. D'un côté, notre cerveau est aussi un ensemble de "boucles stupides". Les électrons porteurs du courant n'ont rien d'intentionnel. Ils n'ont pas de "coeur". Mais nous sommes très sûrs d'en avoir un. Je pense que cette question du "coeur" est un peu la grande question de la "conscience" version romantique. Je pense que nous avons une conscience, et qu'elle

est totalement compatible avec le fait que notre cerveau est formé, en somme, de neurones, de synapses et de courant. L'indépendance des niveaux est une question délicate. Et je pense que ceux qui trouvent les boucles "stupides" et les électrons " idiots" ont souvent une idée très approximative de ce qu'est un électron, et de comment il fonctionne. Ce n'est au fond pas si stupide que ça.

8. Les programmes IA deviendront-ils "super-intelligents" ?
   DH : Je ne sais pas. Il n'est pas certain que nous soyons capables de comprendre une super-intelligence, de communiquer avec, ou que l'idée même d'une super IA ait un sens. Je pense que si notre perception avait été totalement différente, nous aurions probablement eu un tout autre type d'intelligence. Une IA avec notre intelligence se posera probablement les mêmes questions, et il sera intéressant de connaître ses réponses.
   TM : C'est une des meilleures réflexions du livre, et ce n'est pas peu.
9. Les IA seront identiques aux humains ? Il n'y aura pas de différence ?
   DH : Le corps de l'IA aura une grande influence sur son cerveau. Probablement il sera assez différent de nous, peut-être même très différent. C'est une des raisons pour lesquelles la reconnaissance d'une IA pourrait être une tâche ardue.
   TM : Je trouve cette réponse convaincante.
10. Le jour où l'on atteindra l'IA, on aura compris ce que sont l'intelligence, la conscience, le libre arbitre et le "Moi" ?
    DH : Plus ou moins. Ca dépend de ce que l'on veut dire avec "comprendre". Comprenez-vous la musique de Bach ? C'est grâce à vos études ou c'est par les émotions que cela vous donne de l'écouter ? Et comprenez-vous la relativité restreinte ? On peut faire les calculs, mais une connaissance intuitive est une autre question. L'intelligence et la conscience ne seront probablement pas connaissables de manière intuitive, mais nous pouvons comprendre les gens, ce qui est un accès de plus haut niveau à la signification de ces problèmes.
    TM : Je trouve la réponse peu convaincante, mais je crains que toutes celles que j'ai développées jusque là soient encore moins convaincantes que celles de DH.

5.3 Boucles étranges

Une boucle étrange est un système qui, pour se comprendre, "sort du système", mais qui est obligé de faire recours à lui-même pour continuer son travail. Par exemple, les juges d'un pays donné sont sensés "sortir du système" et juger une situation qui concerne la société. Mais que se passe-t-il si les juges

eux-mêmes étaient amenés à se défendre en étant l'un des deux pôles d'une dispute ? C'est là une situation assez bizarre, où un système doit parvenir à se juger lui-même. La perception de soi fonctionne de façon analogue : c'est nous que nous percevons. Comment puis-je juger de ma propre pensée, si le juge est la pensée elle-même ? C'est un casse-tête qui rappelle étonnamment le problème de Gödel.

Mais, en même temps, le théorème de Gödel suggère (suggérer n'est pas prouver !) qu'il pourrait y avoir un mode d'observation de haut niveau de la conscience, qui utilise des concepts qui n'existent pas aux niveaux inférieurs. Certaines choses n'ont simplement pas d'explication au bas-niveau, quelque soit la longueur et la lourdeur de ce que l'on dit. Rappelons que si cette analogie est valable, cela ne nous plonge pas dans le mystère total. Il y a des explications sensées du théorème de Gödel, qui reposent sur les rapports complexes entre méta-niveaux. Hofstadter avance l'dée que les problèmes de la conscience et du libre arbitre, reposeraient sur des boucles étranges, des situations dans lesquelles un niveau élevé influence un bas-niveau, tout en étant influencé par celui-ci.

L'exposition d'estampes est une métaphore de ce que Hostadter appelle un "tourbillon de Gödel", un lieu où tous les niveaux de description se mêlent. Comme le tableau le suggère, ce lieu n'appartient vraiment à aucun niveau et il est blanc. Nous ne pouvons pas le compléter de l'extérieur : c'est impossible en étant en-dehors du système. Si on était dans le système, ce lieu n'existerait simplement pas. C'est peut-être la métaphore la plus suggestive, mais aussi la plus brumeuse de GEB.

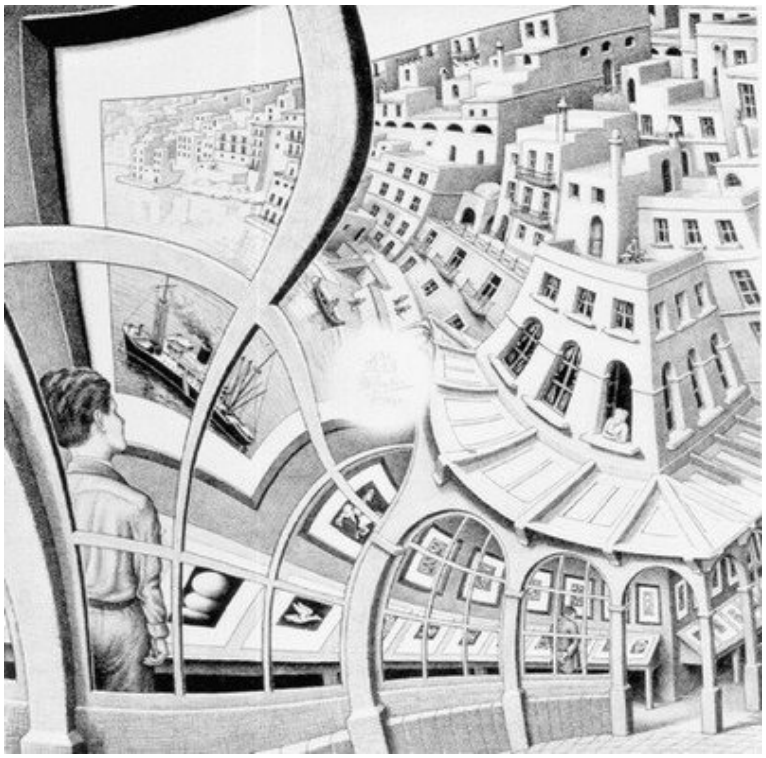


FIGURE 7 – Boucles étranges et auto-reference dans "Exposition d'estampes" de M.C. Escher.

Notes biographiques et éditoriales

Douglas Hofstadter (New York, 1945 – vivant), est un universitaire américain, professeur de sciences cognitives et d'informatique, professeur adjoint d'histoire et de philosophie des sciences, philosophie, littérature comparée et psychologie à l'université de l'Indiana à Bloomington, où il dirige le Centre de Recherche sur les Concepts et la Cognition (source wikipedia). Avec ce livre, il a gagné un prix Pulitzer dans la catégorie non-fiction.

Douglas Hofstadter, "Gödel Escher Bach, les brins d'une guirlande éternelle", InterEditions, Paris, 1996, 883 pages). Traduit par J. Henry et R. French. Preface à l'edition francaise de Douglas Hofstadter. Illustrations de M.C. Escher, R. Magritte, D. Hofstadter.

Première édition anglaise : "Gödel Escher Bach, an eternal golden braid", New York, 1979.

Thomas Mueller. Octobre 2008.