1 Qu'est-ce que l'intelligence artificielle ?

C'est au congrès de Dartmouth en 1956 que l'expression *intelligence artificielle* a été proposée pour désigner le domaine de recherche qui s'ouvrait alors. Le succès de cette appellation provient sans doute de ce qu'elle énonce avec une remarquable économie d'expression une problématique fondamentale : la possibilité de *concevoir une machine intelligente*. Cela ne signifie pas pour autant que tous les chercheurs de ce domaine s'accordent sur ce que l'on entend par cela. Que signifie en effet construire une machine intelligente ? On peut envisager essentiellement deux types de réponses :

1. une machine sera considérée comme intelligente si elle reproduit *le comportement* d'un être humain dans un domaine spécifique ou non
2. une machine sera considérée comme intelligente si elle modélise *le fonctionnement* d'un être humain.

Pour comprendre en quoi ces deux approches diffèrent, prenons l'exemple du jeu d'échecs. Savoir jouer aux échecs est sans aucun doute une de ces capacités humaines qui requiert de l'intelligence. On ne connaît aucun système de règles dont l'observation suffirait à l'un des deux camps pour gagner (ou tout au moins, pour annuler) contre toute défense : ce jeu n'y aurait probablement pas survécu. Réaliser un programme jouant aux échecs peut donc constituer un sujet de recherche en intelligence artificielle. Mais selon que l'on adhère à la première approche de l'Intelligence Artificielle ou à la seconde, on ne réalisera pas la même machine :

- dans le premier cas, on essaiera avant tout d'obtenir un programme efficace. Peu importe alors que la machine fasse des calculs inaccessibles à l'homme, comme explorer quelques centaines de millions (ou milliards) de positions à la seconde.
- dans le second, on essaiera d'abord de comprendre comment l'homme joue aux échecs. Pour cela, on interviewera des maîtres, on essaiera de dégager les règles plus ou moins consciemment suivies par les joueurs : tenter d'occuper le centre, de dominer une couleur de cases, etc. Le programme réalisé validera (ou non) les hypothèses qui auront été faites. En France, l'école de Jacques Pitrat privilégie cette approche.

Les informaticiens adhèrent souvent à la première interprétation de l'intelligence artificielle. On peut également noter les rapports entre cette approche et celle de l'école *béhavioriste* ou *comportementaliste* en psychologie pour laquelle seul le comportement est un sujet d'études scientifique. On désigne parfois par *ingénierie des connaissances* cette branche de l'informatique dont l'objet est de concevoir des machines reproduisant (et dépassant) les capacités humaines (et surtout celles qui semblent échapper à toute méthode, à tout algorithme). Les *sciences cognitives* par contre (ou plus exactement le *cognitivisme* qui en représente le courant principal des années 1960 à nos jours) se sont constituées en

posant comme hypothèse de base la pertinence de l'analogie entre les couples *esprit*|*cerveau* et *logiciel*|*matériel*. Daniel Andler, dans sa préface à *L'introduction aux sciences cognitives* [2], montre très clairement en quoi le paradigme du cognitivisme repose sur cette analogie.
Daniel Andler décrit ce paradigme en trois propositions :

- Le complexe *esprit-cerveau* est susceptible d'une double description : *matérielle* ou *physique* et *informationnelle* ou *fonctionnelle*. Ces deux niveaux sont largement indépendants : de même qu'un programme peut être exécuté sur des machines possédant des architectures différentes, deux individus peuvent présenter les mêmes fonctionnalités sans pour autant que leurs architectures neuronales soient identiques. Le *fonctionnalisme* est alors la notion centrale qui permet d'envisager un *monisme* (il est inutile de postuler l'existence d'un esprit irréductible à la matière) *non réductionniste* (les fonctionnalités ne sont pas identiques à leurs implémentations matérielles).
- Au niveau informationnel, le système cognitif de l'homme est caractérisé par ses *états internes* ou *mentaux* et par les processus qui conduisent d'un état au suivant. Ces états sont *représentationnels* (dotés d'un contenu renvoyant à des entités externes) c'est-à-dire sémantiquement évaluables.
- Les états ou représentations internes sont des formules d'un langage interne (le *mentalais* : mot créé par J. Fodor [6]) proche des langages formels de la logique. Les processus sont effectifs c'est-à-dire *calculables* (voir plus loin).

Il semble qu'actuellement cette dichotomie entre ces deux approches de l'I.A. devienne de moins en moins pertinente et cela pour deux raisons :

- la force brute de l'ordinateur ne suffit jamais à résoudre les problèmes les plus compliqués. Il est nécessaire d'y adjoindre des heuristiques, des méthodes de représentations des connaissances qui proviennent le plus souvent d'une analyse de la manière dont nous fonctionnons.
- le courant cognitiviste semble un peu marquer le pas. Les programmes de recherche très ambitieux qu'il s'était fixé dans les années 1950-60 n'ont pas donné les résultats escomptés. D'autres paradigmes commencent à voir le jour. Le *connectionnisme* (réseaux neuronaux) et les *algorithmes adaptatifs* (algorithmes génétiques) intéressent de plus en plus la communauté des sciences cognitives comme alternative à l'intelligence artificielle classique.

2 Historique bref et lacunaire

2.1 L'époque des automates

A quand remonte l'idée de la possibilité d'une intelligence artificielle ? Si l'on veut retenir l'émergence d'un désir de construire ou de concevoir un homme ou une intelligence artificielle comme moment fondateur, il semble qu'il faille remonter très loin. Dans l'Iliade (chant XVIII), Hephaistos (dieu forgeron) crée des femmes en or qui ont la capacité de parler, travailler, etc.

> << Des servantes s'empressaient pour soutenir le prince, toutes d'or, mais semblables à de jeunes vivantes ; elles ont un esprit dans leur diaphragme ; elles ont la voix, la force, et les immortels leur ont appris à agir. >> [*14*] p. 317

Dans la tradition juive, le Golem est un automate à forme humaine en bois ou en argile. Une inscription magique sur le front en fait un serviteur muet et obéissant. On cite souvent la *machine à calculer* de Pascal (1642) comme étant la première construction effective d'une machine réalisant ce que l'on pouvait croire l'homme, seul capable de faire. Il semble en fait que la première machine à calculer ait été construite par l'Allemand Wilhelm Schickard en 1623. Cette machine a été détruite dans un incendie mais son existence est mentionnée dans une lettre de Schickard à Kepler.

> << Certainement vous rayonnerez en voyant comment la machine mémorise d'elle même les retenues des dizaines et des centaines. >> [*27*]

La question de savoir ce que révèle de la nature de la pensée le fait que l'arithmétique (élémentaire) soit reproductible par une machine a été posée dés cette époque. Diverses réponses ont été apportées :

- Pour Pascal et Descartes, cela montre que l'arithmétique échappe au domaine du raisonnement
- Hobbes en tire la conclusion inverse : le raisonnement n'est jamais qu'un calcul dont la nature mécanique est confirmée par des constructions telles que la machine de Pascal [*2*]

La question reste pertinente de nos jours et le plus étonnant, c'est que les réponses fournies par Pascal et Descartes d'une part, Hobbes d'autre part représentent encore de nos jours deux sensibilités parfaitement identifiables dans la communauté scientifique : que révèle de la nature de la pensée le fait que l'on soit capable d'écrire des logiciels qui jouent très bien aux échecs ? Pour les uns, cela signifie que le jeu d'échecs ne nécessite pas tant d'intelligence qu'on pouvait le penser, pour les autres cela conforte l'idée que l'on pourra créer un jour une machine véritablement intelligente. Pour Descartes, si les comportements des animaux et certaines activités humaines peuvent être simulés par des machines, il

semble impossible de concevoir une machine qui reproduirait le comportement humain dans toute sa généralité. La raison principale de cette impossibilité réside pour lui dans la faculté de l'homme de converser. Nous verrons qu'Alan Turing place également dans cette faculté humaine la frontière qui permettra de dire si une machine est intelligente ou non.

> << Et je m'étais ici particulièrement arrêté à faire voir que, s'il y avait de telles machines qui eussent les organes et la figure d'un singe ou de quelque autre animal sans raison, nous n'aurions aucun moyen pour reconnaître qu'elles ne seraient pas en tout de même nature que ces animaux ; au lieu que, s'il y en avait qui eussent la ressemblance de nos corps, et imitassent autant nos actions que moralement[1] il serait possible, nous aurions toujours deux moyens très certains pour reconnaître qu'elles ne seraient point pour cela de vrais hommes. Dont le premier est que jamais elles ne pourraient user de paroles ni d'autres signes en les composant, comme nous faisons pour déclarer aux autres notre pensée. Car on peut bien concevoir qu'une machine soit tellement faite qu'elle profère des paroles, et même qu'elle en profère quelques-unes à propos des actions corporelles qui causeront quelques changements en ses organes ; comme si on la touche en quelque endroit, qu'elle demande ce qu'on veut lui dire, si en un autre, qu'elle crie qu'on lui fait mal, et choses semblables ; mais non pas qu'elle les arrange diversement pour répondre au sens de tout ce qui se dira en sa présence, ainsi que les hommes les plus hébétés peuvent faire etc. >>[2]

Les *caractéristiques universelles* de Leibniz : tout objet concret ou abstrait, tout comportement a son chiffre, est formalisable (comme on est tenté de l'exprimer au prix d'un anachronisme).

> << Les plus importantes pratiques, les plus importants tours de main dans toutes sortes de métiers et d'artisanats, demeurent encore non écrits. Mais au fond, ce ne sont que des théories, plus particulières et plus complexes, que nous pourrons également formuler par écrit. >> Leibniz in [5]

> << Assigner à chaque objet, son propre chiffre, caractéristique et déterminé. >>

Dés lors, tous les problèmes devraient pouvoir être résolus par le calcul et tous les sujets controversés trouver leur conclusion.

> << Si quelqu'un venait à douter de mes résultats, je lui dirais : << Monsieur, calculons cela ensemble >> et de la sorte, avec une plume et un encrier, nous réglerions la question >>

Les *automates de Vaucanson* (1737) : le joueur de flûte, le canard capable de battre des ailes (chaque aile était composée d'environ deux milles pièces), de manger, de déféquer, etc. Des commentaires à propos des automates construits à cette époque indiquent que l'on a pu penser qu'ils représentaient un immense progrès de la connaissance et que par eux, l'on était proche du mystère de la vie. Cela semble risible de nos jours : comment a-t-on pu tirer de telles conclusions de la réalisation de tels objets, sans aucun doute très ingénieux mais malgré tout, basés sur une technique assez pauvre ? C'est que ces automates représentaient effectivement quelques concepts fondamentaux. Avant qu'une idée soit complètement explicitée, c'est à dire, au moment où elle constitue ou contribue au paradigme que l'on n'a

pas encore dégagé puisqu'il constitue l'espace et l'horizon de toute connaissance, un objet ou une oeuvre d'art peut être ressenti comme exprimant globalement tout ce que l'on peut penser. C'est sans doute le même phénomène qui est à l'origine de l'optimisme démesuré des pionniers de l'intelligence artificielle dans les années 1950-60. D. Andler montre bien dans [2] quels concepts pouvaient être représentés directement par ces automates :

- la fameuse phrase de Galilée

  << la nature est écrite en langage mathématique >>

  illustre parfaitement la découverte extraordinaire que constitue l'intelligibilité du monde grâce aux mathématiques. Le **calcul** est tout-puissant et la mécanique est son instrument (le Dieu horloger). Les raisons manquent pour prouver que la mécanique ne peut pas tout reproduire.
- la **symbolisation** ou la possibilité du codage : il y a un rapport entre les mouvements d'aile du canard de Vaucanson et les rouages qui les causent qui n'est pas d'analogie mais qui préserve toute *l'information*.
- l'**universalité** : la machine à calculer de Pascal calcule la somme ou la différence non pas de deux nombres en particulier (ce qui n'aurait aucun intérêt) mais de deux nombres quelconques. De même les métiers Jacquard peuvent reproduire n'importe quel motif.

La Mettrie (1709-1751) publie son ouvrage le plus connu *L'homme-machine* en 1747. La thèse qui y est défendue est vigoureusement matérialiste : seule la matière existe, la pensée n'en est qu'une propriété et l'Homme est une machine.

> << Le corps humain est une machine qui remonte elle-même ses ressorts etc. >> << L'âme n'est donc qu'un vain terme dont on a point idée, et dont un bon esprit ne doit se servir que pour nommer la partie qui pense en nous >>. << Je crois la pensée si peu incompatible avec la matière organisée, qu'elle semble en être une propriété, telle que l'électricité, la faculté motrice, l'impénétrabilité, l'étendue, etc. >>. << Concluons donc hardiment que l'Homme est une Machine, et qu'il n'y a dans tout l'Univers qu'une seule substance diversement modifiée. >>

Les *métiers Jacquard* (1805) : la caractéristique la plus intéressante de ces métiers à tisser, pour ce qui nous concerne, est qu'ils peuvent être configurés pour reproduire automatiquement n'importe quel motif. A chaque motif correspond une séquence d'instructions matérialisées par des fiches perforées. Il s'agit donc d'une machine programmable universelle (dans l'univers des motifs). La *machine analytique* de Charles Babbage (1842) : après avoir conçu en 1833 une *machine à différence* capable d'effectuer des séquences d'opérations arithmétiques, Charles Babbage consacre le reste de son existence et de sa fortune à la réalisation d'une machine analytique beaucoup plus ambitieuse. Cette machine doit comporter un *magasin* contenant les données sur lesquelles la machine travaille et un *moulin* qui effectue les opérations, contrôlé par des cartes perforées. Le contrôle du calcul est programmé et, c'est là une innovation essentielle, peut dépendre de résultats intermédiaires grâce à des instructions de branchement conditionnel. L'universalité du calcul est donc (virtuellement) atteinte. Cette machine n'a jamais été achevée : elle aurait nécessitée plus de 50 000 pièces mécaniques de haute précision. L'importance de la machine analytique était comprise par ses concepteurs. Il ne s'agit donc pas d'une découverte fortuite.

> << en permettant au mécanisme de combiner des symboles généraux, on

établit un lien unificateur entre les opérations de la matière et les processus mentaux abstraits de la branche la plus abstraite de la science mathématique >> Ada Lovelace (collaboratrice de Babbage et fille de Lord Byron). << la machine analytique est capable de reproduire toutes les opérations que l'intellect effectue en vue d'obtenir un résultat déterminé, à condition que ces opérations soient elles-mêmes susceptibles d'une définition précise >> Luigi Menabrea.

2.2 Naissance de la logique moderne

Alan Turing, qui fut au 20e siècle un des principaux pionniers de l'intelligence artificielle, a commencé sa carrière par de très remarquables travaux en logique mathématique. Le lien entre intelligence artificielle et logique est très ancien. Il provient bien entendu du fait que la reproduction ou la modélisation du raisonnement humain est une des tâches majeures que s'est assignée l'IA et que c'est là aussi l'objet d'études de la logique. Jusqu'au milieu du XIXe siècle, la logique faisait partie de la philosophie. La référence à Aristote était encore inévitable. A partir de De Morgan (1806-1871) et de Boole (1815-1864) (*An Investigation of the Laws of Thought*, 1854), la logique migre vers les mathématiques. Il est possible d'envisager un *calcul* des propositions logiques. Frege (1848-1925) : la *Begriffsschrift* (1879) (idéographie, notation conceptuelle) fonde la science des langages formels. La notion de **système formel** : on peut décrire un système formel par la donnée d'un *langage* (c'est-à-dire d'un *alphabet* et de *règles* permettant de construire les expressions bien formées), d'*axiomes* (sous ensemble de l'ensemble des expressions) et de *règles* permettant d'inférer des expressions à partir d'autres. Formaliser un domaine de connaissances consiste à construire un système formel dans lequel les expressions bien formées codent les expressions pertinentes du domaine (qu'elles soient vraies ou non), les axiomes correspondent aux vérités de base et les règles d'inférence permettent de déduire toutes les vérités du domaine et elles seules.

Exemple : *un fragment de l'arithmétique de Peano.* L'*alphabet* contient la constante a et le symbole fonctionnel un-aire S. On définit les *termes* de manière récursive par :

- a est un terme
- si t est un terme, alors $S(t)$ est aussi un terme.

Les *formules élémentaires* sont de la forme : $t = t'$ où t et t' sont des termes. Les *formules* se déduisent des formules élémentaires par un usage correct des connecteurs logiques Ú, Ù, ¬ et des quantificateurs " et $. Les *expressions bien formées* sont les termes et les formules. Les *axiomes* sont :

$$" x \neg (a = S(x))$$
$$" x \neg (a = x) Þ \$ y (x = S(y))$$
$$" xy (S(x) = S(y)) Þ x = y.$$

Les *règles* ou *axiomes logiques* contiennent (entre autres)

- le *modus ponens* qui peut s'exprimer de la manière suivante : si l'on a déduit les formules F et $F Þ G$, alors on peut déduire la formule G.
- le *modus tollens* qui peut s'exprimer de la manière suivante : si l'on a déduit les formules $F Þ G$ et $\neg G$, alors on peut déduire la formule $\neg F$.

- la *règle d'universalité* : si l'on a déduit la formule " x $F(x)$ alors on peut déduire la formule $F(t)$ pour tout terme t.

Montrons que l'on peut déduire la formule ¬ ($S(a) = S(S(a))$). La règle d'universalité appliquée au premier axiome donne

$$\neg (a = S(a))$$

La règle d'universalité appliquée au deuxième axiome donne

$$S(a) = S(S(a)) \text{ Þ } a = S(a)$$

Le modus tollens appliqué aux deux formules précédentes donne

$$\neg(S(a) = S(S(a)))$$

Le système formel précédent formalise une partie de l'arithmétique élémentaire sur les entiers positifs. Pour le voir, il suffit d'interpréter la constante a par 0 et le symbole S par la notion de successeur. L'axiome " x ¬ ($a = S(x)$) signifie alors que 0 n'est le successeur d'aucun nombre. L'axiome " x ¬ ($a = x$) Þ \$ y ($x = S(y)$) signifie que tout nombre entier non nul est le successeur d'un entier. L'axiome " xy ($S(x) = S(y)$) Þ $x = y$ signifie que si deux nombres ont les mêmes successeurs alors ils sont égaux. Comme ces trois axiomes sont vrais dans N, toutes les formules que l'on peut en déduire le sont également. Par exemple, la formule

$$\neg (S(a) = S(S(a)))$$

que nous venons de démontrer prouve que 1 ¹ 2. Mais cette formalisation n'est pas *complète*, c'est-à-dire qu'il existe des vérités arithmétiques qui ne peuvent pas être déduites du système précédent. On peut montrer que c'est le cas de la formule

$$\text{" } x \neg(x = S(x))$$

qui affirme qu'aucun nombre n'est son propre successeur.

Formalisation et intelligence artificielle semblent indissociables. La raison en est qu'un programme d'ordinateur ne peut traiter que des suites de 0 et de 1, c'est-à-dire des données syntaxiques ; une condition nécessaire à l'intelligence artificielle réside donc dans cette possibilité de formaliser la connaissance et autres facultés humaines. Mais cela ne semble plus aussi clair actuellement : formaliser et réduire à du syntaxique sont-elles des propositions équivalentes ? Jusqu'à la fin du 19e siècle, ni les mathématiques ni même la géométrie n'étaient entièrement formalisées. Certaines définitions d'Euclide font encore appel à l'intuition sensible (le point, la ligne droite, etc.). David Hilbert propose une formalisation de la géométrie en 1899 (*Grundlagen der Geometrie*). En 1900, au IIe congrès international des mathématiciens, il énonce une série de problèmes ouverts dont il pense qu'ils seront au centre des préoccupations des mathématiciens dans les décennies à venir. Le deuxième problème est le suivant : formaliser les mathématiques de telle manière que tout énoncé mathématique soit décidable (voir plus loin le sens de ce mot). Zermelo propose en 1908 un système formel (Z) qui après quelques modifications apportées par Fraenkel en 1922 (ZF) permettra de décrire la quasi totalité des mathématiques telles qu'elles se font encore actuellement.

La notion de **démonstration ou déduction automatique** : si un domaine de connaissances est complètement formalisé et si les règles du raisonnement sont explicitées et syntaxiques, alors tout énoncé vrai de ce domaine doit pouvoir être produit automatiquement. Ainsi, il est très simple de produire toutes les vérités des mathématiques telles qu'on les considèrent aujourd'hui : il suffit d'appliquer

successivement toutes les règles d'inférences à tous les axiomes de ZF ainsi qu'à toutes les formules démontrées à des étapes antérieures. Un des plus gros inconvénients de cette méthode est que si un énoncé est vrai on finira bien par le savoir alors que s'il est faux, on ne le saura jamais ! Il y a une dissymétrie entre vérité et fausseté : on demande en effet rarement à un système formel d'être capable d'inférer toutes les erreurs, tous les énoncés faux de la théorie ! On dit qu'un système formel est *décidable* s'il existe une procédure permettant de décider (en un temps fini) si un énoncé se déduit des axiomes ou non. C'est un tel système que Hilbert souhaitait pour les mathématiques. Malheureusement (ou heureusement selon certains) Gödel montra en 1931 qu'un tel souhait était vain : il n'existe pas de systèmes formels décidables permettant de décrire les mathématiques (et même l'arithmétique). On peut formuler ce résultat autrement : pour tout système formel décrivant l'arithmétique, on peut construire un énoncé vrai bien que non prouvable dans ce système. C'est un argument qui est parfois utilisé pour soutenir qu'une intelligence artificielle est impossible : l'homme peut prouver plus de choses qu'un système formel (et donc qu'une machine).

Le premier théorème d'incomplétude de Gödel :

- dans tout système formel S décrivant l'arithmétique, on peut construire une proposition P qui dit (d'elle même) : << *P n'est pas prouvable dans S* >>.
- si S est consistant, c'est-à-dire s'il est impossible de prouver dans S une chose et son contraire, alors *tout ce qui est prouvable dans S est vrai*.
- supposons que S soit consistant.
  - si P était fausse alors P serait prouvable dans S et donc P serait vraie. Ce qui est absurde. Donc, *P est vraie*.
  - si P était prouvable alors P serait à la fois vraie (puisque S est consistant) et fausse (puisque P exprime que P n'est pas prouvable). *P n'est donc pas prouvable*.
- on en déduit donc que si S est un système formel consistant décrivant l'arithmétique, on peut *construire* des propositions vraies et non prouvables dans S.

D'après ce qui précède, raisonner c'est appliquer des règles, c'est-à-dire, calculer. Mais qu'est-ce qu'un *calcul* ? Qu'est-ce qu'un objet calculable ? Qu'est-ce qu'une fonction calculable ? Peut-on se mettre d'accord sur ce que l'on entend par là ? Les fonctions suivantes sont calculables :

- $f(n) = 3n^2 - 2n + 1$
- $f(n)$ = le plus grand diviseur premier de n s'il existe et 0 sinon.

Elles sont calculables parce qu'il existe un *algorithme* qui pour chaque valeur de n permet de déterminer automatiquement et en un temps fini la valeur de $f(n)$. Mais que sont les algorithmes ? Peut-on les décrire ? Est-il vrai que toute fonction définie de N dans N est calculable ? Cette question a été abordée indépendamment par plusieurs mathématiciens dans les années 30. Il n'était pas question de démontrer un théorème mais de préciser, de caractériser formellement une notion intuitive : la **calculabilité**. Plusieurs réponses apparemment très différentes ont été données par Church (en lambda-calcul), Turing (les machines de Turing - 1936), Gödel, Kleene. On a pu démontrer que toutes ces caractérisations étaient en fait

équivalentes, en dépit des différences de formulation. Cela conduit à ce qu'on appelle la *thèse de Church-Turing* :

> *la notion de calculabilité est parfaitement caractérisée par la notion de machine de Turing (par exemple).*

Encore une fois, il s'agit là d'une thèse et non d'un théorème. Cette thèse n'est pas remise en question de nos jours.

Le modèle de calcul le plus simple est la *Machine de Turing*, créée par Alan Turing en 1936. Ce modèle est encore utilisé chaque fois qu'il s'agit de démontrer des résultats concernant la calculabilité. Une machine de Turing est constituée d'un ruban infini des deux côtés divisés en cases contenant chacunes une lettre de l'alphabet S = {0, 1, *Blanc*}. Une tête de lecture-écriture pointe vers l'une des cases du ruban. Un ensemble fini Q décrit l'ensemble des états dans lequel peut se trouver la machine. On réserve la notation q_{init} (resp. q_{fin}) pour désigner l'état initial de la machine (resp. son état final). Un ensemble fini d'instructions permet de décrire la dynamique de la machine. Ces instructions sont de la forme :

$$q, x \text{ ® } y, q', d$$

où :

- q et q' sont des éléments de Q, c'est-à-dire des états de la machine
- x et y sont des éléments de S et
- d est un élément de l'ensemble $\{D, G\}$.

La dynamique de la machine est alors décrite par la règle suivante : si la machine est dans l'état q et si la tête de lecture pointe vers une case contenant la lettre x alors la machine écrit y à la place de x, elle passe dans l'état q' et la tête de lecture-écriture se déplace d'une case dans la direction indiquée par d (droite pour D et gauche pour G). On dit qu'une machine M calcule la fonction f si chaque fois que la bande contient l'entier n (écrit en binaire) et qu'on lance la machine M (la tête de lecture pointant vers la première case de n et la machine étant dans l'état q_{init}), la machine M finit par s'arrêter sur l'état q_{fin} en ayant inscrit l'entier $f(n)$ sur le ruban. Une manière d'exprimer la thèse de Church-Turing est de dire : toute fonction *intuitivement* calculable peut être effectivement calculée par une machine de Turing. Cela peut paraître surprenant et il est impossible de le démontrer. On peut par contre montrer que toute fonction calculable par un programme Pascal est calculable par une machine de Turing (et réciproquement). Une autre manière encore de formuler la thèse de Church-Turing : toute fonction intuitivement calculable peut être effectivement calculée par un programme Pascal (ou Prolog !). Bien entendu, on suppose toujours que l'on dispose d'autant de temps et d'espace mémoire qu'il est nécessaire.

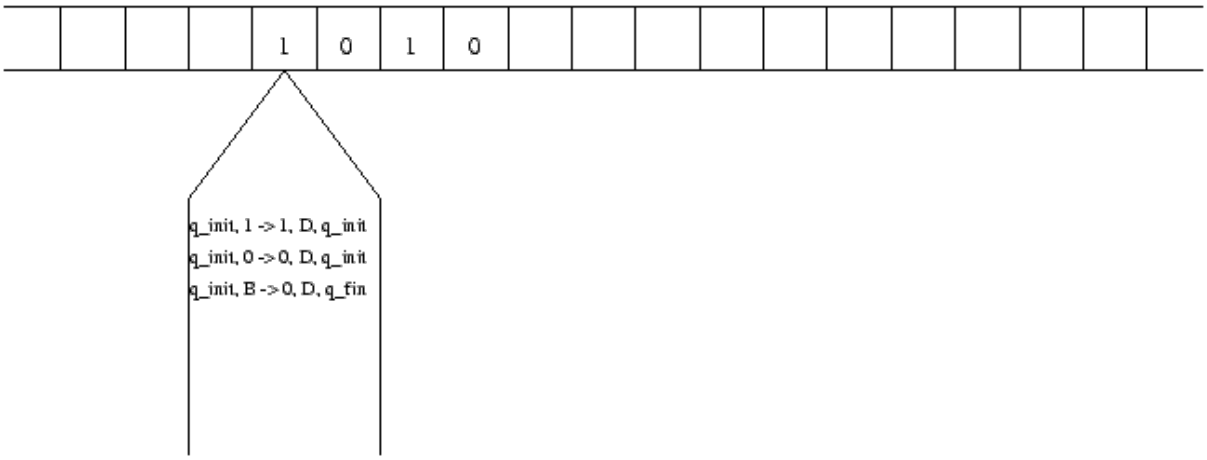

Figure 1 : Une machine de Turing qui multiplie par 2

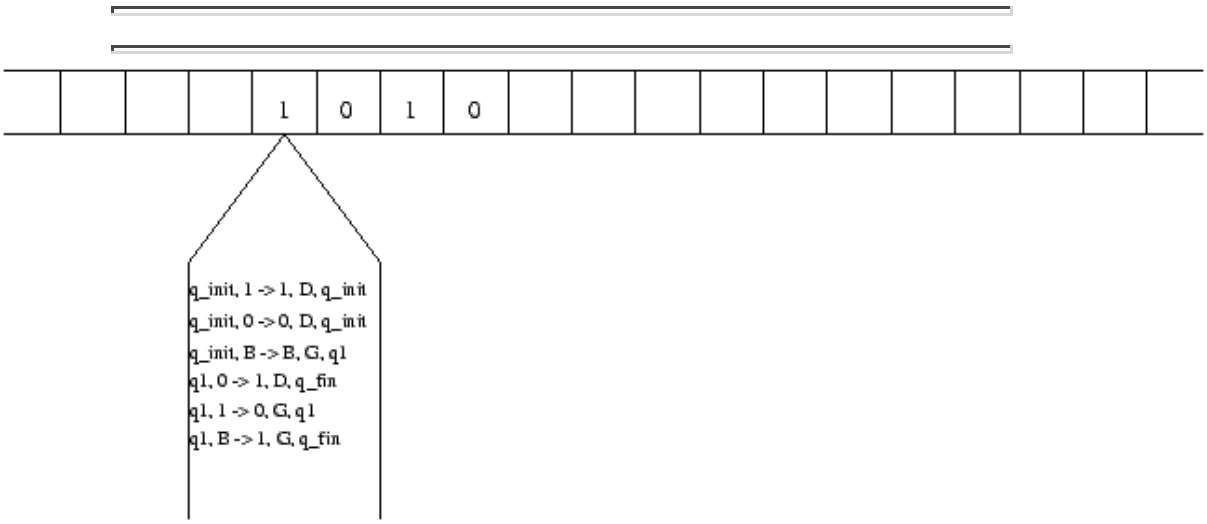

Figure 2 : Une machine de Turing qui ajoute 1

Machines spécialisées et machines universelles : les machines ci-dessus sont spécialisées. Elles calculent une fonction et une seule. Il est possible de construire une machine de Turing universelle, c'est-à-dire capable de calculer toutes les fonctions calculables. La construction repose sur le résultat suivant : il est possible d'énumérer effectivement toutes les machines de Turing. C'est-à-dire qu'il est possible de coder toute machine de Turing T par un entier n de telle manière qu'étant donné une machine de Turing T il est possible de calculer son code n et qu'étant donné un entier n il est possible de construire la machine dont il est le code. On note T_n la machine de Turing dont le code est l'entier n. On considère alors la machine de Turing U définie par :

$$U(nBp) = T_n(p).$$

La machine U interprète la première partie de la donnée comme étant le code de la machine (spécialisée) qu'elle doit simuler et la deuxième partie comme étant la donnée à lui transmettre. Finalement, n représente le *programme* de U tandis que p représente la *donnée*. Cette machine est *universelle* puisqu'elle peut *simuler* toutes les machines de Turing T_n. Ce qui précède peut paraître très compliqué. Il y a pourtant au moins deux idées essentielles à en retirer.

- Une seule machine est suffisante pour calculer tout ce qui est calculable.
- Pour une machine de Turing universelle, il n'y a pas de différence de nature entre programme et donnée. Cette idée a été reprise (ou réinventée) par Von Neumann lorsqu'il a défini l'architecture des premiers ordinateurs : programmes et données doivent être stockés dans une mémoire unique. C'est encore le cas pour les ordinateurs que nous utilisons.

2.3 L'intelligence artificielle après 1950

Les idées dégagées dans le paragraphe précédent constituent le fondement théorique de l'intelligence artificielle. Les comportements humains ne sont rien d'autre que le résultat d'un calcul portant sur des données plus ou moins complexes et tout calcul peut être simulé par une machine de Turing universelle. Tous les comportements humains peuvent donc être simulés par une telle machine. Jusqu'au milieu des années 40, cette thèse est purement théorique : il n'est pas question de la valider par des machines réelles. La construction inachevée de la machine analytique de Babbage montre les limites de la technologie mécanique. Les progrès de la technologie électrique puis électronique vont modifier le statut de cette thèse : il devient possible de la valider expérimentalement. L'histoire de l'informatique et des ordinateurs n'est pas le sujet de ce cours. L'ouvrage *Une histoire de l'informatique* de Philippe Breton [4] est très bien documenté et nous y renvoyons le lecteur intéressé. Signalons quand même quelques dates importantes :

- l'ENIAC, conçu aux Etats-Unis pendant la seconde guerre mondiale est le dernier grand calculateur plutôt que le premier ordinateur. Il fut achevé en novembre 1945. Cette machine pesait 30 tonnes, avait une surface au sol de 160 mètres carrés, comprenait 17468 tubes à vides, 70000 résistances, 10000 capacités, 1500 relais et 6000 commutateurs manuels. L'exécution des programmes était très rapide (200000 instructions par seconde) mais la programmation était beaucoup plus laborieuse : il fallait positionner à la main des milliers de commutateurs. L'ENIAC pêchait par manque d'universalité : il était impossible de mémoriser les programmes autrement qu'en configurant les commutateurs.
- John Von Neumann décrit dans un texte d'une dizaine de pages en juin 1945 (First Draft of a Report on the EDVAC) quelques principes qui sont encore à la base des ordinateurs actuels. L'idée essentielle est que les instructions et les données doivent être stockées en mémoire, l'unité centrale se chargeant d'appliquer les premières aux secondes. Cette *architecture* avait été décrite par Alan Turing dans son article sur les machines universelles et il est probable que Von Neumann s'en est inspiré.
- le tout premier langage de programmation fut conçu par Alan Turing. Il contenait une cinquantaine d'instructions que l'ordinateur (Mark1) traduisait automatiquement en binaire. Le langage FORTRAN a été écrit entre 1953 et 1956, le LISP (principal langage de l'IA jusqu'à PROLOG, créé par McCarthy) en 1956, l'ALGOL et le COBOL en 1960,le PL1 et PASCAL en 1964, PROLOG en 1974 (langage de PROgrammation en LOGique conçu par Colmerauer et Kowalski), ...
- les transistors sont mis au point en 1947 ; ils remplacent les lampes à vide dans les années 50. Le premier prototype de circuit intégré est créé en 1958 et le premier micro-processeur en 1971. Le premier micro-ordinateur est commercialisé en 1975 et le ``PC'' d'IBM date de 1981.

En 1950, Alan Turing écrit *Computing machinery and intelligence*, un article dans lequel il propose un test, communément appelé maintenant le **test de Turing**, qui devra permettre de décider si une machine est intelligente ou non. Il énonce également une série d'objections à l'intelligence artificielle qu'il réfute une à une. Le *test de Turing* est basé sur le jeu de l'imitation.

> << C'est un jeu à trois joueurs, un homme (A), une femme (B) et un interrogateur (C) qui peut être de l'un ou l'autre sexe. L'interrogateur est dans une pièce, isolé des deux autres joueurs. L'objet du jeu pour l'interrogateur est de déterminer lequel des deux autres joueurs est l'homme et lequel est la femme. Il ne les connaît que par des étiquettes X et Y et à la fin du jeu il dit soit << X est A et Y est B >>, soit << X est B et Y est A >>. L'interrogateur est autorisé à poser des questions à A et B comme :
>
> > C : X peut-il me dire quelle est la longueur de ses cheveux ?
>
> Maintenant, supposons que X est A. A doit alors répondre. L'objectif de A est de faire en sorte que C fasse une fausse identification. Sa réponse pourrait donc être :
>
> > Je suis coiffée à la garçonne et mes mèches les plus longues font à peu près 20 cm'.
>
> Afin que la voix ne puisse pas aider l'interrogateur, les réponses devraient être écrites, ou mieux, dactylographiées. Le meilleur aménagement serait que les deux pièces communiquent par téléscripteur. Ou alors, les questions et réponses peuvent être relayées par un intermédiaire. L'objet du jeu pour le troisième joueur (B) est d'aider l'interrogateur. Sa meilleure stratégie est probablement de donner des réponses vraies. Elle peut ajouter des choses comme
>
> > << Je suis une femme, ne l'écoutez pas ! >>
>
> mais cela ne lui profitera en rien car l'homme peut faire des remarques similaires. Nous posons maintenant la question : Que se passerait-il si une machine prenait la place de A dans le jeu ?. L'interrogateur se trompera-t-il aussi souvent lorsque le jeu est joué de cette manière que lorsqu'il est joué entre un homme et une femme ? Ces questions remplacent la question originale ``les machines peuvent-elles penser?'' >>.

Le grand mérite de ce test est d'opérationnaliser la question : les ordinateurs peuvent-ils penser ? Il permet de se passer d'une définition pour l'instant inaccessible de ce qu'est l'intelligence. Deux questions distinctes se posent naturellement :

- pourra-t-on réaliser un jour une machine qui passe ce test avec succès ?
- si oui, pourra-t-on dire que cette machine est intelligente ?

Les partisans de ce qu'on appelle l'IA forte répondent ``oui'' aux deux questions. Une très nombreuse littérature a été consacrée à ce test ainsi qu'aux réponses que l'on peut apporter aux questions précédentes. Voir par exemple ci-dessous l'argument de la ``Chambre chinoise'' de Ronald Searle qui entend prouver que la réponse à la deuxième question doit être ``non''. Dans le cours de l'article, Alan Turing fait également la prédiction suivante :

> << Je crois que d'ici 50 ans, il sera possible de programmer des ordinateurs, avec une capacité mémoire d'à peu près 10^9, de façon qu'ils jouent si bien au jeu de l'imitation qu'un interrogateur moyen n'aura pas plus de 70% de chances de procéder à l'identification exacte après 5 minutes d'interrogation. >>

Il reste 4 ans pour réaliser cette prédiction. Il semble que l'on en soit très loin même si le temps d'interrogation est petit et le taux d'erreur admissible assez grand. Mais est-ce si sûr ?

Les neuf objections envisagées par Turing contre la possibilité de l'I.A.

- *l'objection théologique*

  > << Penser est une fonction de l'âme immortelle de l'homme. Dieu a donné une âme immortelle à tout homme et à toute femme, mais à aucun autre animal ni à aucune machine. Aucun animal ni aucune machine ne peut donc penser. >>

- *l'objection de l'autruche* (littéralement : de la tête dans le sable)

  > << Le fait que les machines pensent aurait des conséquences trop épouvantables. Il vaut mieux croire et espérer qu'elles ne peuvent pas le faire. >>

- *l'objection mathématique* : cette objection est basée sur le théorème d'incomplétude de Gödel de 1931. Quelque soit le système formel S (pourvu qu'il soit assez puissant), il existe des énoncés vrais (et que l'on peut montrer tels) que S ne peut pas produire. L'homme peut donc prouver plus de choses qu'une machine. Cette objection a particulièrement été étudiée par Lucas. La réfutation la plus courante consiste à dire que si S est un système formel qui ne peut pas produire l'énoncé vrai E, on peut facilement construire un système formel S' qui peut produire toutes les vérités produites par S ainsi que E. Il suffit de prendre $S' = S \cup \{E\}$. D'une part, il n'existe pas de vérités échappant à tout système formel et d'autre part, il n'est nullement exclus que l'homme n'ait pas les mêmes limites.
- *l'objection de la conscience* : écrire un sonnet ou composer un concerto nécessite de ressentir ce que l'on fait, c'est-à-dire au minimum, d'en avoir conscience. Or une machine ne possède pas de conscience. Mais comment s'assurer qu'un homme ressent ce qu'il dit, qu'il possède une conscience ? A ce propos, Hoffstadter et Dennett citent l'histoire des deux sages taoïstes :

  > << Deux sages étaient debout sur un pont enjambant une rivière. L'un dit à l'autre : << j'aimerais être un poisson, ils sont si heureux ! >> Le second répondit : << Comment savez-vous si les poissons sont heureux ou non ? Vous n'êtes pas un poisson. >> Et le premier répondit : << Mais vous n'êtes pas moi, alors comment savez-vous si je sais ce que ressentent les poissons ? >> >>

- *les arguments provenant de diverses incapacités*

  > << je vous concède que vous pouvez fabriquer des machines qui fassent tout ce que vous avez mentionné, mais vous ne serez jamais capable d'en construire une qui fasse X >>

L'avantage de cette objection, c'est qu'elle est modulable dans le temps : X est toujours quelque chose que l'on ne sait pas faire à un moment donné.

- *l'objection de Lady Lovelace*

  > << La Machine Analytique n'a pas la prétention de créer quoi que ce soit. Elle peut faire tout ce que nous savons lui ordonner de faire. >>

  Cette objection est devenue très classique. L'homme crée la machine, il a donc une supériorité sur elle. Il est pourtant facile de programmer une machine afin qu'elle réalise des tâches que son programmeur ne sait pas forcément faire (comme bien jouer aux échecs, par exemple).
- *l'objection de la continuité du système nerveux* : le système nerveux est continu, les machines sont discrètes. Un système analogique affichera peut être des résultats qu'un système symbolique sera incapable de prédire, mais un système symbolique semble capable de simuler n'importe quel système analogique.
- *l'objection du caractère non formalisable du comportement* : comment énoncer les quelques règles que je suis en conduisant une auto ? Mais on confond souvent *règles de conduite* et *lois du comportement*. Il peut être impossible de dégager des règles, cela ne prouve pas que le comportement n'obéit pas à des lois. Il peut être de même très difficile de retrouver un programme à partir de quelques-uns de ses calculs.
- *l'objection de la perception extrasensorielle* : si la télépathie est possible, l'objection est sérieuse ! Pour distinguer l'homme de la machine, il suffit en effet que l'interrogateur demande << quelle carte ai-je dans la main ? >>. Je ne sais pas si Turing prenait cette objection au sérieux.

Ces objections et leurs réfutations ont suscité de nombreux travaux dont la bibliographie donne quelque références.

La chambre chinoise de Searle Le philosophe anglais John Searle décrit dans l'article ``*Minds, Brains and Programs*'' publié en 1980 une expérience de pensée (*Gedankenexperiment*) qui selon lui permet de réfuter l'assertion défendue par les partisans de l'IA forte selon laquelle une machine satisfaisant avec succès au critère de Turing doit être considérée comme intelligente. Cette expérience peut être décrite de la manière suivante :

> John Searle est enfermé dans une pièce ne communiquant avec l'extérieur que par un guichet et contenant un (très) gros livre dans lequel est écrit une succession de questions et de réponses pertinentes à ces questions, questions et réponses étant rédigées en *chinois*. Searle précise qu'il ne connaît rien au chinois et que l'anglais est sa langue maternelle. Un expérimentateur lui transmet des messages par le guichet, tantôt en anglais, tantôt en chinois. Searle répond directement aux messages rédigés en anglais alors que pour ceux rédigés en chinois, il est obligé de consulter le livre jusqu'à trouver une question *identique* au message ; il recopie alors la réponse associée. Searle fait alors remarquer que pour l'expérimentateur extérieur, les messages transmis en chinois sembleront aussi pertinents que ceux transmis en anglais. Et pourtant Searle connaît l'anglais alors qu'il ne connaît rien au chinois. De manière analogue, on ne peut pas déduire du fait qu'un programme passe avec succès le test de Turing qu'il comprenne de quoi il est question.

L'argument semble très fort à première vue. On lui fait souvent l'objection suivante

: s'il est vrai que Searle ne connaît pas le chinois, que peut-on dire du système composé de Searle et du livre ? Les partisans de l'IA forte diront que ce système connaît le chinois. La preuve en est qu'il est capable de répondre de manière pertinente à n'importe quelle question rédigée en chinois. Et il semble que l'on ait pas avancé d'un pouce !

Premières réalisations :

- 1956 : *Logic Theorist* de Newell, Shaw et Simon. Démonstrateur automatique de théorèmes.
- 1959 : *GPS* (General Problem Solver) de A. Newell et H. A. Simon. Le premier logiciel qui s'attaque à la résolution de problèmes en général et non dans quelques domaines particuliers. Le nom du logiciel est un peu optimiste en regard des résultats.
- 1958 : Newell et Simon prédisent qu'avant 10 ans, un programme d'IA aura :
  - battu le champion du monde d'échecs
  - démontré un théorème important en mathématiques
- 1956 : premier traducteur Anglais-Russe

Cette première décennie a vu la réalisation de très nombreux travaux en I.A. qui auront surtout permis de prendre la mesure de la difficulté de la tâche ! Les premiers résultats sont en effet assez décevants et le décalage entre les déclarations plus qu'optimistes et les résultats est très frappant. Hubert Dreyfus, philosophe américain, a étudié avec un oeil extrêmement critique le développement de l'intelligence artificielle et la communauté de ses chercheurs. Son livre *Intelligence artificielle - Mythes et limites* [5] contient à la fois une analyse et un témoignage très intéressants. L'un des problèmes sur lequel nombre des premiers travaux ont achoppé est celui de l'*explosion combinatoire*. On définit la *complexité* en temps d'un algorithme comme le nombre d'étapes que requiert son exécution *en fonction de la taille des données* qu'il doit traiter. Lorsque cette complexité dépend linéairement ou polynomialement de la taille des données, l'algorithme est généralement *praticable*. C'est à dire qu'on peut l'implanter sur un ordinateur et espérer obtenir une réponse dans un délai raisonnable. Lorsque cette complexité est exponentielle, l'algorithme n'est plus praticable. Or la plupart des algorithmes qui résolvent naïvement un problème d'intelligence artificielle ont une complexité exponentielle. Il est très facile par exemple de concevoir un algorithme qui joue aux échecs de manière optimale. Il suffit, à partir d'une configuration du jeu, de calculer *toutes* les suites possibles (il n'y en a qu'un nombre fini) et de choisir la meilleure d'entre elles. On voit bien que cette solution est pratiquement irréalisable.
Ces premières difficultés ont amené les chercheurs en IA à dégager un certain nombre de principes généraux :

- un algorithme doit combiner les approches *combinatoires* (utilisant la puissance de calcul de l'ordinateur) et *heuristiques* (utilisant des connaissances sur le problème à résoudre)
- il faut distinguer dans un algorithme ce qui est connaissance sur le problème de ce qui est méthode de résolution (*approche déclarative*)
- les programmes d'IA doivent incorporer des modules d'*apprentissage* qui leur permettront d'améliorer leurs performances automatiquement (plus facile à dire qu'à faire !)

Les années 60 et 70 : période faste pour l'IA qui voit la réalisation de très nombreux projets d'importance. Les projets sont plus ciblés que lors de la décennie précédente, des principes méthodologiques sont précisés et surtout, l'IA bénéficie

de l'extraordinaire augmentation de la puissance des ordinateurs liée à la miniaturisation et à l'intégration des composants.

- 1970 : SCHRDLU, logiciel conçu par Terry Winograd. Il simule la manipulation de blocs géométriques (cubes, cylindres, pyramides, ...) posés sur une table. Le logiciel génère automatiquement des plans (<< Pour déplacer le cube bleu sur le sommet du cylindre jaune, je dois d'abord enlever la pyramide qui se trouve sur le cube et ...>>) et est muni d'une interface en langage naturel.
- Les systèmes experts, parmi lesquels :
  - 1969 : DENDRAL : analyse des résultats d'une spectrographie de masse
  - 1974 : MYCIN : diagnostic de maladies infectieuses
- 1977 : MACSYMA (logiciel de calcul formel)
- ...
- le 29 août 1994, le programme CRESS GENIUS 2.9 bat avec les noirs le champion du monde Gary Kasparov.

La liste précédente est loin d'être exhaustive, même en ce qui concerne les principales réalisations. Nous n'avons en particulier pas mentionné les nombreux travaux sur l'approche ``neuronale'' de l'IA. Nous renvoyons au chapitre sur le connexionnisme et à la bibliographie pour de plus amples informations. On assiste actuellement à la fois à un reflux de l'IA sur la scène médiatique (les trop grands espoirs sont toujours déçus) et à une banalisation des techniques élaborées lors des décennies précédentes (systèmes experts, représentation des connaissances, reconnaissance des formes, robotique, ...). Dés le début des années 80, l'intelligence artificielle a fait son entrée dans le monde économique, et chaque mois voit l'annonce de nouvelles réalisations. Pourtant, aussi spectaculaire que soit chaque nouvelle avancée, on garde souvent un sentiment de déception tant le vocable d'intelligence artificielle porte d'exigences en lui. Un logiciel a battu le champion du monde Gary Kasparov, soit, mais c'était un jour ou il n'était pas au mieux de sa forme ! Un logiciel arrive à reconnaître la parole, soit, mais pour un locuteur unique et encore, à condition qu'il ne soit pas enrhumé ! L'intelligence artificielle a-t-elle une borne ? Si c'est le cas, et si on arrive à la caractériser, on aura appris quelque chose d'essentiel sur ce qu'est l'intelligence. Et sinon, on risque d'être toujours déçu par ses réalisations :

> << le dernier roman de GENIUS SCRIPTOR est complètement raté. Où est l'inventivité de ses premières années ? On a déjà lu tout cela. Et si le style reste admirable, le récit tourne à vide etc. >>

3 Les domaines de l'intelligence artificielle

- Les **systèmes experts** : un système expert est un logiciel capable de simuler le comportement d'un expert humain effectuant une tâche précise. Il s'agit là d'un domaine où le succès de l'intelligence artificielle est incontestable et cela est sans doute dû au caractère très ciblé de l'activité que l'on demande de simuler. Le logiciel MYCIN (1974) par exemple est capable de fournir un très bon diagnostic dans la mesure où il est spécialisé dans certains types de leucémies.
- Le **calcul formel** (opposé au calcul numérique) traite des expressions symboliques. Par exemple, calculer la valeur d'une fonction réelle en un point est du calcul numérique alors que calculer la dérivée d'une fonction numérique est du calcul formel. Les logiciels actuellement proposés sur le marché (MATHEMATICA, MAPLE, ...) sont capables d'effectuer tous les calculs formels (et bien d'autres) exigés il y a peu des étudiants des premières années des cycles scientifiques.
- La **représentation des connaissances** : si l'on veut qu'un logiciel soit capable de manipuler des connaissances, il faut savoir les représenter symboliquement. C'est là un des secteurs les plus importants de la recherche en intelligence artificielle.
- La **simulation du raisonnement humain**. Les hommes sont capables de raisonner sur des systèmes incomplets, incertains et même contradictoires. On tente de mettre au point des logiques qui formalisent de tels modes de raisonnement (logiques modales, temporelles, floue, non monotones, etc.).
- Le **traitement du langage naturel**. Qu'il s'agisse de traduire un texte dans une autre langue ou de le résumer, le problème crucial à résoudre est celui de sa compréhension. On pourra dire qu'un logiciel comprend un texte lorsqu'il pourra le représenter sous une forme indépendante de la langue dans laquelle il est écrit : c'est une tâche très difficile mais, bien que l'on soit très loin des enthousiasmes des premiers temps, des progrès significatifs ont d'ores et déjà été réalisés. On commence à trouver des traducteurs automatiques d'autant meilleurs qu'ils traitent d'un domaine spécialisé et des logiciels documentaires non triviaux.
- La **résolution de problèmes** : représentation, analyse et résolution de problèmes concrets. Les jeux fournissent une bonne illustration de ce domaine : le champion du monde de Backgammon est un programme depuis quelques années déjà et cela sera vraisemblablement aussi le cas pour le jeu d'échecs dans peu de temps. Le jeu de Go résiste beaucoup plus aux efforts des programmeurs de jeux.
- la **reconnaissance de la parole** : les progrès sont beaucoup plus lents qu'on ne l'imaginait mais constants. On est encore loin de pouvoir produire un logiciel capable de reconnaître les paroles d'un locuteur quelconque et cela essentiellement parce que la compréhension d'un mot, d'une phrase requiert beaucoup d'informations extra-langagières (le contexte, la connaissance du monde dans lequel nous vivons interviennent de manière fondamentale). Un Dictaphone automatique a

malgré tout été proposé dans le commerce en 1994 mais il ne fonctionne que si le locuteur sépare chacun des mots et n'effectue aucune liaison.
- La **reconnaissance de l'écriture** : même la reconnaissance de l'écriture dactylographiée n'est pas un problème facile (bien qu'on commence à trouver sur le marché des logiciels très performants). L'écriture manuscrite pose des problèmes autrement plus ardus : cela n'est pas étonnant dans la mesure où cette tâche peut nous poser à nous aussi des problèmes insolubles. Certains chercheurs essaient de reconstituer le mouvement de la main à partir du texte qu'elle a écrit afin de comprendre ce qui a été écrit : recherches prometteuses ?
- La **reconnaissance des visages** : longtemps considéré comme un des problèmes les plus difficiles de l'intelligence artificielle, il semble que l'on obtienne des résultats intéressants en utilisant des réseaux neuronaux.
- la **robotique**. Il y a déjà longtemps que des robots industriels ont fait leur apparition dans les usines. On appelle robot de la première génération, ceux qui sont capables d'exécuter une série de mouvements préenregistrés. Un robot de la deuxième génération est doté de moyens de perception visuel lui permettant de prendre certaines décisions. Un robot de la troisième génération, objet des recherches actuelles, doit acquérir une plus grande autonomie comme se déplacer dans un environnement inconnu. On est encore loin du robot domestique ou ménager !
- l'**apprentissage**. On a compris très tôt qu'un logiciel devrait avoir des capacités d'apprentissage autonome pour pouvoir être véritablement qualifié d'intelligent. Douglas Lenat travaille actuellement à la constitution d'une gigantesque base de données censées contenir toute la pragmatique, c'est-à-dire toutes les connaissances souvent implicites partagées par les humains d'un même groupe indispensables à la communication. Il est impensable de saisir manuellement toutes ces informations et Lenat souhaite adjoindre un module d'apprentissage à sa base de données lui permettant de travailler seule, c'est-à-dire de collecter des informations nouvelles dans des textes ou par discussion avec des humains et de réorganiser seule l'architecture de ses connaissances : utopie ? A suivre.
- les **réseaux neuronaux**. Un réseau de neurones formels est un modèle rudimentaire du cerveau humain, chaque cellule neuronale étant décrite comme une fonction à seuil possédant une sortie et dont les entrées sont reliées à d'autres neurones. Il est pourtant possible d'effectuer des tâches non triviales à l'aide de tels réseaux (la reconnaissance des formes et en particulier des visages en étant l'exemple le plus frappant). Ces réseaux partagent plusieurs propriétés importantes avec le cerveau humain : répartition de l'information sur l'ensemble du réseau (où se trouve la mémoire dans le cerveau ?), programmation non explicite (nous ne savons pas non plus ce que nous savons), etc. Le connectionnisme, théorie cognitive dans laquelle les explications sont recherchées au niveau neuronal, peut-elle prendre le pas sur le cognitivisme ?
- les **systèmes complexes adaptatifs** : on regroupe sous ce vocable les

algorithmes génétiques et les modèles de vie artificielle. Il s'agit là, énoncé de manière abusivement succincte, d'étudier comment des populations soumises à des lois simples et naturelles convergent naturellement vers des formes organisées.