Une deuxième révolution galiléenne ?

Gilles Dowek

Science is what we understand well
enough to explain to a computer.
Art is everything else we do.
D.E. Knuth

Le concept de concept structurant

Dans l'histoire des sciences, des questions, apparemment indépendantes, ont souvent reçu une même réponse. Par exemple, les questions « Qu'est-ce qui fait tomber les pommes des arbres ? », « Qu'est-ce qui fait tourner les planètes autour du Soleil ? », « Qu'est-ce qui assure la cohésion des étoiles ? » ont toutes reçu la même réponse : « La gravité ». Quand cela se produit, nous pouvons dire les concepts utilisés dans cette réponse structurent un champ de la connaissance. Par exemple, le concept de gravité structure la mécanique.

Au cours de l'histoire, de nombreuses questions n'ont pu recevoir une réponse satisfaisante qu'après la découverte d'un concept structurant, souvent lors de la tentative de répondre à une autre question. Par exemple, la découverte du concept d'électron a permis de donner une réponse satisfaisante à la question « Qu'est-ce que le courant électrique ? », alors qu'il a été découvert pour répondre à la question « Que sont les rayons cathodiques ? ».

Il semble que le concept d'algorithme permette aujourd'hui de répondre à de nombreuses questions qui n'avaient jusqu'alors reçu que des réponses insatisfaisantes.

Qu'est-ce qu'un aéroport ?

Nous étudions souvent des phénomènes en les « mettant en équation ». Par exemple, pour répondre à de nombreuses questions qui se posent à propos du mouvements d'une masselotte oscillant au bout d'un ressort — celle de la position de la masselotte à une date future, ou celle de la période de ses oscillations —, nous commençons par mettre ce phénomène en équation :

$$m\ddot{x} = -kx$$

Pour certains phénomènes, cependant, cette mise en équation est plus difficile. Par exemple, imaginons que nous cherchions à répondre à la question du nombre maximal d'avions qui peuvent se trouver sur la piste d'un aéroport utilisée selon les règles suivantes :

– un avion qui est sur la piste peut la quitter, en roulant vers un garage,

– un avion qui est en vol peut atterrir sur la piste, si celle-ci est libre.

De même que la position de la masselotte à un instant donné pouvait être décrite par un nombre réel, l'état de cette piste peut être décrit par un nombre entier : le nombre d'avions qui s'y trouvent. Et il n'est pas difficile de montrer que, si le nombre d'avions sur la piste est nul à l'instant initial, et si nous excluons le cas où deux avions atterrissent exactement au même instant, alors, à chaque instant, le nombre d'avions sur la piste sera toujours égal à zéro ou à un. En effet, à partir de l'état « 0 avion sur la piste », le système peut rester dans ce même état ou évoluer vers l'état « 1 avion sur la piste », et à partir de l'état « 1 avion sur la piste », le système peut rester dans ce même état ou évoluer vers l'état « 0 avion sur la piste ».

Toutefois, contrairement au cas de la masselotte oscillant au bout d'un ressort, pour établir ce résultat nous n'avons pas mis le phénomène en équation, même si nous l'avons mathématisé.

La description mathématique d'un aéroport entier est plus complexe, car il faut tenir compte du fait qu'une piste peut être utilisée non seulement atterrir, mais aussi pour décoller, qu'avant d'atterrir un avion doit passer par diverses zones d'attentes, obtenir une autorisation, ... mais l'idée générale demeure : nous décrivons d'abord l'ensemble des états possibles de l'aéroport, puis les règles de transition qui définissent la manière dont l'aéroport peut évoluer d'un état à un autre. Ces règles sont des algorithmes qui prennent en argument un état de l'aéroport et qui retournent l'ensemble des états qui peuvent lui succéder.

L'étude d'un tel système dépend beaucoup du fait que l'ensemble des états accessibles à partir de l'état initial est fini ou infini. Dans le premier cas, il est possible d'énumérer tous les états accessibles. Dans le second, il est souvent nécessaire de trouver des invariants des transitions — comme le fait que le fou, aux échecs, ne change jamais de couleur. Mais, dans un cas comme dans l'autre, la mathématisation du phénomène n'aboutit pas à une description équationnelle, mais à une description algorithmique.

Si la réponse à la question « Qu'est-ce qu'une masselotte oscillant au bout d'un ressort ? » est « Une équation différentielle », alors la réponse à la question « Qu'est-ce qu'un aéroport ? » semble davantage être « Un algorithme ».

Qu'est-ce que la synthèse d'une protéine ?

Depuis son invention, à la fin du XVII$^{\text{e}}$ siècle, le calcul différentiel a montré son efficacité pour mathématiser les phénomènes mécaniques, puis électromagnétiques, économiques, ... Il est donc tentant de voir, dans ce formalisme, le langage dans lequel le Grand Livre de la Nature est écrit.

Toutefois, en feuilletant un livre de biologie, nous ne pouvons manquer d'être surpris par la rareté des équations différentielles. Pourtant, la relation en une séquence d'ARN messager et une protéine est aussi rigoureuse que la relation entre la position et l'accélération d'une masselotte oscillant au bout d'un ressort. Et celle-là ne semble avoir aucune raison d'être plus rétive à la mathématisation que celle-ci. En fait, la relation entre une séquence d'ARN et une protéine est, aujourd'hui, aussi mathématisée que la relation entre la position et l'accélération d'une masselotte, mais elle n'est pas mathématisée sous la forme d'une équation différentielle : comme un aéroport, cette relation est mathématisée sous la forme d'un algorithme, qui décrit le processus de synthèse de la protéine à partir de l'information codée dans le brin d'ARN.

Cet algorithme n'est pas difficile à décrire. Un brin d'ARN est une suite de nucléotides. Chaque nucléotide contenant une base azotée — l'adénine (A), la guanine (G), la cytidine (C) ou l'uracile (U) —, il y a quatre types de nucléotides et un brin d'ARN est donc un mot dans un alphabet à quatre lettres. Une protéine, quant à elle, est une suite d'acides aminés. Il y a vingt acides aminés qui entrent dans la composition des protéines — l'alaline (Ala), l'arginine (Arg), ... —, et une protéine est donc un mot dans un alphabet à vingt lettres. Pour déterminer la protéine synthétisée par un brin d'ARN il faut commencer par le parcourir, jusqu'à trouver le triplet AUG. Puis on continue en lisant les lettres trois par trois. À chaque triplet de lettres correspond un acide aminé, sauf pour les trois triplets UAA, UAG et UGA qui marquent la fin de la synthèse de la protéine. Cet algorithme peut se décrire simplement par le programme suivant

```
let rec debut l = match l with
| A::U::G::_ -> l
| _::r -> (debut r)
|[] -> []

let rec traduction l = match l with
| x::y::z::r -> if List.mem (x,y,z) stop
                then []
                else (List.assoc (x,y,z) table)::(traduction r)
| _ -> []

let arn_to_proteine l = traduction (debut l)
```

où `stop` est la liste `[(U,A,A);(U,A,G);(U,G,A)]` des triplets qui marquent la fin de la synthèse de la protéine et `table` est la table

```
[((G,C,U),Ala);((G,C,C),Ala);((G,C,A),Ala);((G,C,G),Ala);((C,G,U),Arg); ...
```

qui associe un acide aminé à chacun des soixante et un autres triplets.

La réponse à la question « Qu'est-ce que la synthèse d'une protéine ? » semble donc également être « Un algorithme. »

Le caractère précurseur de la chimie

Une branche entière de l'informatique est consacrée à la conception de langages permettant d'exprimer des algorithmes. Ces langages peuvent grossièrement se classer en deux catégories. D'un coté, *les langages de programmation*, comme Caml — que nous avons utilisé ci-dessus — C, Java, ... ont une grammaire et un lexique riches, qui donnent de nombreux outils aux programmeurs pour exprimer leurs algorithmes. D'un autre coté, les machines de Turing, le lambda-calcul, ... ont une grammaire et un lexique minimaux. Ils sont, de ce fait, plus difficiles à utiliser, mais conceptuellement plus simples. L'un de ces langages, *la réécriture*, permet d'exprimer un algorithme sous la forme d'un ensemble de règles, qui décrivent la manière de transformer une expression en un résultat. Par exemple, si nous écrivons les nombres dans la notation primitive dans laquelle le nombre n est noté par n bâtons, l'algorithme de l'addition peut s'exprimer avec deux règles

$$+y \longrightarrow y$$

$$(|x) + y \longrightarrow x + (|y)$$

Ainsi l'expression $|| + ||$ se transforme-t-elle successivement en $| + |||$ — avec la seconde règle —, puis en $+||||$ — avec la seconde règle à nouveau — puis en $||||$ — avec la première règle — qui ne peut plus se transformer et qui est donc le résultat.

Ce langage ne comporte donc que deux catégories syntaxiques : les expressions, telles $(|x)+y$ qui sont formées de constantes — $|$, $+$, ... — et de variables — x, y, ... — et les règles, qui sont formées de deux expressions séparées par le symbole $\longrightarrow$.

Ce symbole $\longrightarrow$, qui n'est relié que de manière lointaine au symbole $\mapsto$ utilisé en mathématique pour définir une fonction — $x \mapsto x + 2$ —, est, en revanche, très proche du symbole $\longrightarrow$ utilisé en chimie pour décrire les réactions

$$2H_2 + O_2 \longrightarrow 2H_2O$$

La chimie, comme la biologie plus récemment, décrit donc, et depuis longtemps, des phénomènes de manière algorithmique et comme l'ont souligné G. Berry et G. Boudol, cette description des réactions chimiques peut être considérée comme le précurseur des descriptions algorithmiques modernes de phénomènes de toutes sortes.

La réponse à la question « Qu'est-ce qu'une réaction chimique ? » semble donc également être « Un algorithme. »

Le caractère précurseur de la grammaire

Un autre domaine précurseur dans la description algorithmique des phénomènes est la grammaire. N. Chomsky a en effet proposé en 1957 de décrire la grammaire d'une langue comme un algorithme formé de règles de réécriture. Par exemple avec les règles suivantes

$$S \longrightarrow GN\ GV$$

$$GN \longrightarrow NP$$

$$GV \longrightarrow VT\ GN$$

$$NP \longrightarrow \text{Pierre}$$

$$NP \longrightarrow \text{Paul}$$

$$VT \longrightarrow \text{écoute}$$

l'expression S se transforme successivement en $GN\ GV$, $NP\ GV$, $NP\ VT\ GN$, $NP\ VT\ NP$, Pierre $VT\ NP$, Pierre écoute NP et Pierre écoute Paul, qui ne peut plus être transformée et qui est donc un résultat. Dans cette grammaire simple l'expression S ne se réécrit qu'en quatre phrases. Mais elle peut, bien entendu, être enrichie de manière à décrire une langue qui en comprend davantage.

Cette formulation des grammaires est plus précise que la formulation traditionnelle, dans laquelle chaque règle est exprimée en langue naturelle. En particulier, pour montrer qu'une phrase est correcte, il suffit de donner une suite de transformations : l'interprétation est donc réduite au minimum. Cependant, elle ne s'éloigne pas trop de la description traditionnelle : les trois premières règles, par exemple, peuvent se lire « La séquence d'un groupe nominal et d'un groupe verbal est une phrase », « Un nom propre est un groupe nominal » et « La séquence formée d'un verbe transitif et d'un groupe nominal est un groupe verbal ».

La réponse à la question « Qu'est-ce qu'une grammaire ? » semble donc également être « Un algorithme. »

À quelles questions la réponse est-elle « Un algorithme » ?

Les questions « Qu'est-ce qu'un aéroport ? », « Qu'est-ce que la synthèse d'une protéine ? », « Qu'est-ce qu'une réaction chimique ? », « Qu'est-ce qu'une grammaire ? » reçoivent donc une même réponse : « Un algorithme ». Et cette réponse est également une réponse possible aux questions : « Qu'est-ce qu'un agent économique ? », « Qu'est-ce qu'un nombre entier ? », « Qu'est-ce qu'un nombre réel ? », « Qu'est-ce qu'une démonstration mathématique ? » « Qu'est-ce qu'une théorie ? », ...

Qu'est-ce qu'une loi physique ?

Superficiellement, il semble donc y avoir deux manières de décrire des phénomènes de manière mathématique. En mécanique, en électromagnétisme, en économie, ... les phénomènes semblent se décrire dans le langage des équations différentielles. En biologie, en chimie, en grammaire, ... dans celui des algorithmes. Cela amène à s'interroger sur les liens entre ces deux langages.

L'équation différentielle

$$m\ddot{x} = -kx$$

a comme solution la fonction

$$x(t) = x_0 \cos(\sqrt{\frac{k}{m}}t + \phi)$$

qui peut être calculée par un algorithme. De même, l'équation de la chute des corps dans le vide

$$m\ddot{x} = mg$$

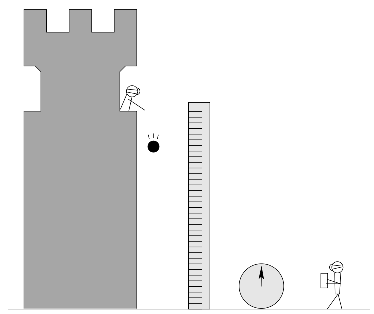
a comme solution la fonction

$$x(t) = \frac{1}{2}gt^2$$

qui, elle aussi, peut être calculée par un algorithme.

La question « Qu'est-ce qu'une loi physique ? » semble donc également recevoir, au moins dans ces deux cas, la réponse « Un algorithme ».

En fait, chaque expérience physique, c'est-à-dire chaque système physique muni d'un protocole qui spécifie les grandeurs choisies et observées, par exemple l'expérience qui consiste à laisser tomber un objet dans le vide pendant un temps donné et à mesurer la distance parcourue



définit une relation : la relation réalisée par cette expérience. Et une question scientifique fondamentale est de caractériser l'ensemble des relations qui peuvent être ainsi réalisées par une expérience de physique.

Une hypothèse, *la forme physique de la thèse de Church*, est que les relations qui peut être ainsi physiquement réalisées sont les exactement les relations qui peuvent être calculées par un algorithme.

La première raison que nous avons de croire cette hypothèse vraie est qu'elle n'a pas, à ce jour, été réfutée : personne n'a observé ou construit de système physique qui réalise une relation non algorithmique. Une seconde raison est que cette hypothèse peut être démontrée à partir d'autres hypothèses sur la nature. En particulier, R. Gandy a montré que cette hypothèse est une conséquence de trois autres hypothèses : une hypothèse de finitude de la vitesse de transmission de l'information, une hypothèse de finitude de la densité de l'information et une hypothèse d'homogénéïté de l'espace et du temps.

Cette forme physique de la thèse de Church, si elle est vraie, explique, pourquoi les lois de la nature peuvent être exprimées en langage mathématique : si les lois de la nature peuvent être exprimées en langage algorithmique, alors elles peuvent *a fortiori* l'être en langage mathématique.

Si cette hypothèse est vraie, la réponse à la question « Qu'est-ce qu'une loi physique ? » est donc également « Un algorithme ». La différence entre les deux manières de décrire un phénomène de manière mathématique semble donc se réduire. Les équations différentielles

$$m\ddot{x} = mg$$

$$m\ddot{x} = -kx$$

sont des moyens de définir les fonctions

$$x(t) = \frac{1}{2}gt^2$$

$$x(t) = x_0 \cos(\sqrt{\frac{k}{m}}t + \phi)$$

qui peuvent être calculées par des algorithmes.

Cette remarque a amené M. Pour El et I. Richards a se poser la question du caractère algorithmique des solutions des équations différentielles. Dans le cas général, leur résultat est négatif : certaines équations différentielles ont des solutions qui ne peuvent pas être décrites par un algorithme. Mais le caractère très artificiel de leur contre-exemple — qui utilise une condition aux limites algorithmique, mais dont la dérivée ne l'est pas — pose à son tour le problème de la caractérisation des équations différentielles dont les solutions sont algorithmiques. De nombreuses équations différentielles, et probablement toutes celles utilisées en physiques, sont dans ce cas.

Les équations différentielles nous apparaissent donc comme un moyen, parmi d'autres, de définir des algorithmes et les deux manières de décrire les phénomènes naturels : par des équations ou par des algorithmes ne s'opposent finalement pas : la description équationnelle semble au contraire s'intégrer parfaitement dans le schéma plus vaste de la description algorithmique.

La description algorithmique des phénomènes

Cette possibilité de décrire algorithmiquement les phénomènes naturels semble donc donner une solution au « problème de Galilée » : pourquoi le Grand Livre de la Nature est-il écrit en langage mathématique ? Elle explique également pourquoi ces phénomènes peuvent être simulés *in silicio*. Elle permet également d'étendre le champ des phénomènes que nous pouvons décrire. En particulier, nous pouvons décrire ainsi des systèmes beaucoup plus complexes qu'une masselotte oscillant au bout d'un ressort, ou en chute libre dans le vide, par exemple des aéroports entiers.

Mettre en équation une forme aussi complexe que celle du musée Guggenheim de Bilbao

semble très difficile. Et utiliser une telle description pour déterminer si ce bâtiment tient debout ou non semble plus difficile encore. En revanche, cela devient possible, et a été réalisé, si nous utilisons une description algorithmique de ce bâtiment. En observant la forme de ce bâtiment, nous pouvons nous demander si la raison pour laquelle de nombreux bâtiment du passé sont des parallélépipèdes surmontés d'un prisme, n'est pas dû aux instruments que nous utilisions alors pour décrire, dessiner et calculer ces bâtiments.

Cette idée de description algorithmique des phénomènes a également permis de poser de nouvelles questions. Par exemple, la théorie économique montre souvent l'existence, sous certaines hypothèses, de maxima. La démarche algorithmique en économie permet de s'interroger sur la quantité de calculs nécessaire pour déterminer un tel maximum. Si cette quantité est linéaire en le nombre d'agents, il se peut que déterminer ce maximum ne demande, à chaque agent, qu'une quantité de calculs indépendante du nombre d'agents. Si, en revanche, elle est exponentielle en le nombre d'agents, alors il est illusoire d'espérer que, même en se répartissant le travail, les agents arrivent à calculer ce maximum.

Une deuxième révolution galiléenne ?

Une question qui se pose naturellement est celle de la raison pour laquelle cette révolution se produit aujourd'hui, alors que nous connaissons des algorithmes depuis des millénaires.

Si nous connaissons des algorithmes depuis la nuit des temps, cela fait en revanche à peine quelques décennies que nous écrivons des programmes, c'est-à-dire que nous exprimons ces algorithmes dans des langages aussi formels que les langages de programmation, et c'est cette révolution qui a fourni les langages que nous utilisons aujourd'hui pour décrire algorithmiquement les phénomènes.

Comme la révolution qu'a constituée, à l'époque de Galilée, l'utilisation du langage mathématique pour décrire les phénomènes physiques, la révolution que constitue l'utilisation d'un langage algorithmique pour décrire ces phénomènes est, d'abord, une transformation du langage dans lequel la science s'écrit.