A second Galilean revolution?

Gilles Dowek

A second Galilean revolution?

Gilles Dowek

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

The concept of structuring concept

In the history of sciences, some apparently independent questions often have received the same answer. For example, the questions "What does make apples fall off trees?", "What does make planets revolve around the Sun?", "What does make matter hold together in a star?" all have received the same answer: "gravity". When this happens, we can say that the concepts used in this answer structure a field of knowledge. For example, the concept of gravity structures mechanics.

Throughout history, many questions could receive a satisfactory answer only after a structuring concept has been discovered, often in the attempt at solving another problem. For example, the discovery of the concept of electron has permitted to give a satisfactory answer to the question "What is electric current?", but it has been discovered attempting at answering the question "What are cathode rays?".

It seems that nowadays the concept of algorithm permits to give satisfactory answers to many questions that had so far received only unsatisfactory ones.

What is an airport?

We often study a phenomenon by describing it "by means of equations". For example, to answer the many questions that can be asked about the motion of a mass in a mass-spring system—that of the position of the mass at a given moment in the future, or that of the period of its oscillations—, we first describe the systems by means of the equation

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

However, for some phenomena, this description by means of equations is more difficult. For instance, imagine we are trying to solve the problem of the maximal number of aircraft that can simultaneously be on the runway of an airport, used according to the following rules:

- an aircraft that is on the runway can leave it, taxiing to a hangar,
- an aircraft on flight can land on the runway, if no aircraft is already on this runway.

In the same way, the position of the mass at any moment could be described by a real number, the state of this runway can be described by a natural number: the number of aircraft on this runway.

And it is not difficult to prove that, if the number of aircraft on the runway is zero in the initial state, and if we exclude the case where two aircraft land exactly at the same time, then, at every future moment, the number of aircraft on the runway will be equal to zero or to one. Indeed, starting from the state "zero aircraft on the runway", the system can remain in this state or evolve to the state "one aircraft on the runway", and from the state "one aircraft on the runway", the system can remain in this state or evolve to the state "zero aircraft on the runway".

However, unlike in the case of the mass-spring system, to establish this result, we have not formulated the problem by means of equations, even if we have formulated it in a mathematical way.

The mathematical description of a full airport is more complex, because it must take into account that a runway can be used not only to land, but also to take off, and that, before landing, an aircraft must go through various holing patterns, obtain clearances... but the general idea remains: we first describe the set of possible states for the airport, then transition rules that define the way the airport can evolve from one state to another. These rules are algorithms that take as input an airport state and output the set of the possible successors of this states.

Studying such a system requires various methods depending on whether the set of states that are accessible from the initial state is finite or not. In the first case, it is possible of enumerate all the accessible states. In the second, it is often necessary to find invariants of the transitions, like the fact that, in the chess game, the bishop is always on a square of the same color. But, in one case and in the other, the mathematization of the phenomenon does not lead to an equational description, but to an algorithmic one.

If the answer to the question "How can a mass-spring system be described?" is "with a differential equation", then that to the question "How can an airport be described?" seems rather to be "with an algorithm".

What is the synthesis of a protein?

Since its invention, at the end of the 17^{th} century, calculus has proved its effectiveness to mathematize mechanics, then electromagnetism, economics... And it is thus tempting to see, in this formalism, the language in which the Great Book of Nature is written.

However, flipping through a biology book, we are often surprised by the scarcity of differential equations. Nevertheless, the relation between a messenger RNA sequence and a protein is as rigorous as that between the position and

the acceleration of the mass in a mass-spring system. And there is no reason why the first relation would be more recalcitrant to mathematization than the second. Actually, the relation between a messenger RNA sequence and a protein is, nowadays, as much mathematized as that between the position and the acceleration of a mass, but it is not mathematized by means of a differential equation: like an airport, this relation is mathematized by means of an algorithm, that describes the synthesis process of the protein from the information encoded in the strand of RNA.

This algorithm is not difficult to describe. A strand of RNA is a sequence of nucleotide. Each nucleotide containing a nucleobase, adenine (A), guanine (G), cytidine (C) or uracil (U). As, there are four types of nucleotides, a stand of RNA is a word in a four letter alphabet.

In turn, a protein is a sequence of amino acids. Proteins are built with twenty different amino acids, alanine (Ala), arginine (Arg)..., and a protein is therefore a word in a twenty letter alphabet. To determine the protein synthesized by a strand of RNA, we must scan it until we find a triple AUG and then continue scanning letters tree by three. To each triple of letters corresponds an amino acid, except for the three triples UAA, UAG, and UGA that signify the end of the synthesis process. This algorithm can be simply described by the following program:

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

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

let rna_to_protein l = translation (start l)
```

where `stop` is the list `[(U,A,A);(U,A,G);(U,G,A)]` of triples that signify the end of the synthesis process and `table` is the table

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

that associates an amino acid to each of the sixty one other triples.

The answer to the question "How can the synthesis of a protein be described?" seems therefore also to be "with an algorithm."

Chemistry as a precursor

An entire branch of informatics focuses on the conception of languages permiting to express algorithms. These languages can be roughly classified in two

categories. On the one hand, programming languages, such as Caml, that we have been using above, C, Java... have a rich grammar and a vocabulary, and provide many tools to the programmers to express algorithms. On the other hand, Turing machines, lambda-calculus... have a minimal grammar and vocabulary. They are therefore more difficult to use, but they are conceptually simpler. One of these languages, rewriting, permits to express an algorithm as a set of rules, that describe how to an expression transforms step by step to a result. For example, if we write numbers using the archaic notation where the number n is expressed with n strokes, the addition algorithm can be expressed with two rules

$$+y \longrightarrow y$$

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

This way, the expression $|| + ||$ transforms successively to $| + |||$, with the second rule, then to $+||||$, again with the second rule, then to $||||$, with the first rule, that can no more be transformed and that therefore is the result.

This language hence contains only two syntactic categories: the expressions, such as $(|x) + y$ that are built with constants $|$, $+$... and variables x, y... and the rule, that are formed with two expressions separated with the symbol $\longrightarrow$.

This symbol $\longrightarrow$, that is only loosely related to the symbol $\mapsto$ used in mathematics to define functions, $x \mapsto x + 2$, is, in contrast, very close to the symbol $\longrightarrow$ used in chemistry to describe reactions

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

Thus, chemistry, like, more recently biology, has for long described phenomena by means of algorithms, as noted by G. Berry and G. Boudol. This description of chemical reactions can be considered as the precursor of modern algorithmic descriptions of all kinds of phenomena.

The answer to the question "How can a chemical reaction be described?" seems therefore also to be "with an algorithm."

Grammar as a precursor

Another precursor domain in the algorithmic description of phenomena is grammar. N. Chomsky has indeed proposed in 1957 to describe the grammar of a language as an algorithm formed with rewrite rules. For example, with the following rules

$$S \longrightarrow NP\ VP$$

$$NP \longrightarrow PN$$

$$VP \longrightarrow TV\ NP$$

$$PN \longrightarrow \text{Alice}$$

$$PN \longrightarrow \text{Bob}$$

$$TV \longrightarrow \text{sees}$$

the expression S transforms successively to $NP\ VP$, $PN\ VP$, $PN\ TV\ NP$, $PN\ TV\ PN$, Alice $TV\ PN$, Alice sees PN, and Alice sees Bob, that can no more be transformed and that is therefore a result. In this simple grammar, the expression S rewrites in four sentences only. But it can, of course, be extended in order to describe a language that contains more sentences.

This formulation of grammars is more precise than the traditional formulation, where each rule is expressed in a natural language. In particular, to prove that a sentence is syntactically correct, it is enough to provide a sequence of transformations: the room left for interpretation is therefore minimal. Nevertheless, it is not too far from the traditional formulation of grammars: the three first rules, for example, can be read "the sequence of a noun phrase and a verb phrase is a sentence", "a proper name is a noun phrase", and "The sequence formed with a transitive verb and a noun phrase is a verb phrase".

The answer to the question "How can a grammar be described?" therefore also seems to be "with an algorithm."

To which questions is "an algorithm" the answer?

The questions "How can airport be described?", "How can the synthesis of a protein be described?", "How can a chemical reaction be described?", "How can a grammar be described?" all receive the same answer: "with an algorithm". And this answer is also a possible answer to the questions: "How can an economic agent be described?", "How can a natural number be described?", "How can a real number be described?", "How can a mathematical proof be described?", "How can a theory be described?"...

What is a physical law?

Thus, at a first sight, there seems to be two ways to mathematize phenomena. In mechanics, electromagnetism, economics... phenomena are described by means of differential equations. In biology, chemistry, grammar... by means of algorithms. This raises the question of the links between these to means to describe the world.

The differential equation

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

has the solution

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

that can be computed by an algorithm. In the same way, the free fall equation

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

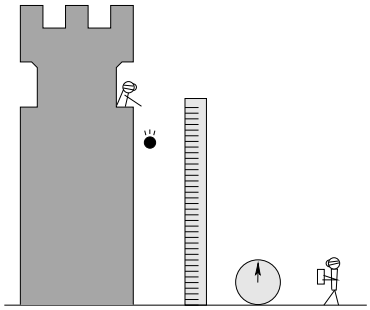
has the solution

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

that also can be computed by an algorithm.

The question "How can a physical law be described?" thus also seems to receive, at least in these two cases, the answer "with an algorithm".

In fact, each physical experiment, that is each physical system equipped with a protocol that specifies the chosen magnitudes and the observed ones, for example the experiment in which we drop an object in vacuum for a given time



and measure the distance travelled by the object during this time, defines a relation: the relation realized by this experiment. And a fundamental question is to characterize the set of relations that can this way be realized by a physical experiment.

An hypothesis, *the physical Church-Turing thesis*, postulates that the relations that can be realized by a physical experiment are exactly those that can be computed by an algorithm.

The first reason to believe that this thesis is true is that it has not be refuted so far: nobody has ever observed or built a physical system, that realizes a non algorithmic relation. Another reason is that this hypothesis can be proven from other hypotheses on nature. In particular, R. Gandy has proved that it is a consequence of three other hypotheses: that the velocity of information propagation is bounded, that the density of information is bounded, and that time and space are homogeneous.

This physical Church-Turing thesis, if it holds, explains why the laws of nature can be expressed in the mathematical language: if the laws of nature can be expressed in an algorithmic language, they can *a fortiori* be expressed in the mathematical language.

If this hypothesis is true, the answer to the question "How can a physical law be described?" also is "with an algorithm" and the difference between the two ways to describe phenomena dwindles.

The differential equations

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

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

are means to define the functions

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

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

that can be computed by algorithms.

This remark has lead M. Pour El and I. Richards to raise the question of the algorithmic nature of the solutions of differential equations. In the general case, their result is negative: some differential equations have solutions that cannot be described by an algorithm. But their counter-example is very artificial, as it uses limit conditions that are algorithmic, but whose derivative is not. It raises, on its turn, the question to characterize the differential equations whose solutions are algorithmic. Many differential equations, and probably all those used in physics, are in this case.

The differential equations then appear as a way, among others, to define algorithms and, in the end, the two means to describe the natural phenomena: with differential equations and with algorithms should not be opposed: the equational description seems, on the contrary, to fit perfectly in the more general scheme of the algorithmic description.

The algorithmic description of phenomena

Thus, this possibility of describe natural phenomena algorithmically seems to give a solution to "Galileo's problem": why is the Great Book of Nature written in the mathematical language? It also explains why these phenomena can be simulated *in silicio*. It also permits to extend the realm of phenomena that can be described. In particular, we can describe this way much more complex systems than a mass-spring system, or a free falling object, for instance a whole airport.

Describing, by means of equations a shape as complex as that of the Guggenheim museum in Bilbao

seems to be very difficult. And even more using such a description to determine if this building stands or not. Nevertheless, all this becomes possible, and it has been made, if we use an algorithmic description of this building.

Contemplating this building, we can ask ourselves if the reason why many buildings in the past were parallelepipeds with a prism sitting on top, is not just a consequence of the instruments we have been then used to describe, design, and compute these buildings.

This algorithmic description of phenomena also permits to ask new questions. For example, economics often shows the existence, under some hypotheses, of

equilibria. The algorithmic approach to economics permits to investigate the amount of computation needed to compute such an equilibrium. If this amount is linear in the number of agents, then determining this equilibrium requires, to each agent, a amount of computation that is independent of the number of agents. If, in contrast, it is exponential in the number of agents, then it is hopeless to expect that, even distributing the computing load, the agents will succeed in computing this equilibrium.

A second Galilean revolution?

A question that naturally arises is why this revolution occurs now, while we have been knowing algorithms for millenia.

If we have been knowing algorithms for ages, we have been writing programs only for a few decades. It is only a few decades ago, that we started expressing algorithms in programming languages. And it is this revolution that has provided the languages that we use today to algorithmically describe the phenomena.

Like the Galilean revolution—the use of the mathematical language to describe physical phenomena—the use of an algorithmic language to describe these phenomena is, in the first place, a revolution of the language in which science is written.