# Unfolding Cognitive Capacities

Jacques Dubucs

Institut d'Histoire et de Philosophie des Sciences et des Techniques (CNRS/Paris I/ENS)

13, rue du Four, 75006 Paris, France
jacques.dubucs@cnrs-dir.fr

http://www-ihpst.univ-paris1.fr/

ABSTRACT. As regards cognitive capacities, the point of view of classical Artificial Intelligence has been much challenged by the so-called emergentist point of view. This paper attempts to outline, on the basis of logical considerations dealing with practical feasibility, a general theory of incompressible unfoldings that is consonant with an old Leibnizian stance rather with the contemporary theory of complexity. I defend a variant of emergentism according to which any process that leads to endow a system with cognitive capacities is such an incompressible unfolding.

## 1 Unfolding as a Problem

The basic assumptions of the traditional Artificial Intelligence have been much criticized in recent times. A lot of well-known examples, observations and *Gedankenexperimente* have established that it is not enough, to behave appropriately in the real world, to be equipped with some complex set of instructions and to follow them blindly. To sum up this now familiar point, to enjoy intelligence or cognitive capacities cannot be equated with the mere execution of a computer programme, for intelligence or cognitive abilities are *emergent* properties that only can arise from a long process of 'learning' or 'adaption' involving the interactions of multiple simple components.

To make this vague idea more precise, it is usefull to refer to the parallelism between design and understanding. On one hand, to design a device capable of having some target behavior in certain circumstances amounts to conceiving and realizing a whole mechanism able to produce the desired output in presence of these circumstances or partially caused by them. On the other hand, to understand a given system one has to find the mechanism that explains its behavior by throwing light on the transitions that lead from the initial conditions of the system to its future states and that permit to predict them. The task of design is to conceive a mechanism that produces desired outcomes that are known in advance (it is like retrospective explanation, except that the outcomes to explain are actual in the case of explanation, while they are inactual in the case of design). The task of predictive understanding is to discover the way of functioning of the given system in order to foresee its future unknown configurations. A crucial issue, both for design and for understanding, has to do with the basic repertory that is available. To design a robot able to easily move in a room cluttered by furniture, we are of course not licensed to call on *easy walkers* or *dexterous artefacts* ; and we rather have to make do with ubiquitous unintelligent components like sensors or photoelectric cells. In the same way, we are not licensed, to predict the effects of ingesting opiates, to make appeal to unpackaged entities like *virtus dormitiva* ; and we rather have to set down our explanations in the elementary language of neuropharmacology.

People in the tradition of classical Artificial Intelligence have been convinced that the gap between whole intelligent behavior and elementary devices at hand could be easiely filled by *planning*. One divides the problem of obtaining the target behavior into a manageable series of subgoals until elementary goals are reached, each of them being easily performable by a specific (modular) basic device acting under the controll of a central 'administrator'. It has become nowadays commonplace that this conception of planned action is generally not suitable and that the target behaviors are more efficiently achieved in a bottom-up way, by leaving very elementary and unspecific devices freely interacting each with the others and with their environment. But it has not been remarked that epistemology of predictive understanding encounters a problem that is exactly parallel, namely the gap between, on one hand, gross input/output correlations that merely restate what is to be explained without supplementary predictive or explanatory value and, on the other hand, elementary transition laws describing step-by-step the evolution of the microstructure that is involved. In other words, the central problem of design, namely the problem of breaking down the target behavior of the whole into micro-operations to be performed by the parts, echoes the dual problem of extracting predictions concerning the future of the macro-system from the knowledge of the micro-transitions of its parts. The aim of this paper is to explain how the second problem can be resolved and how that resolution can be applied to the problem of designing devices endowed with cognitive abilities.

Let $S$ the class of the possible states of a macro-system, $G$ the class of the « desirable » mappings from $S$ to $S$ (of course, not absolutely desirable, but considered as such by the designer). For a particular targeted mapping $F$ in $G$, the problem of design is to find mappings $f_1$, …, $f_n$ that represent available elementary transitions from a state of a component to a following one and that are able, when composed together, to give $F$ in $G$. Inversely, the problem of prediction consists in computing the global mapping $F$ that results from the successive applications and compositions of the elementary $f_i$, which are supposed already known. One can think of the process by which the elementary, local transitions $f_i$ eventually reveal their global joint effect, namely the associated mapping $F$ (or the value of that mapping for the initial conditions), as an *unfolding* process. There are, in some sense, no more information in the mapping $F$ than in the

« folder » { $f_1$, ..., $f_n$ } from which $F$ arises. That information is simply contained in the folder in a compressed way. Design is compressing, prediction is decompressing.

Firstly, let us note how encompassing the notion of unfolding is. Derivation in formal systems can be viewed as the process of unfolding the mathematical content of the axioms by means of the progressive application of the inference rules. Running a computer programme can be viewed as unfolding the content implicit in its instructions. The growth of leaves from buds is also, litterally, unfolding, as well as the development of the whole plant from seed.

Secondly, one has to distinguish between real unfolding, namely the actual development of an item from corresponding germs, and epistemic unfolding, namely the intellectual process that leads to the knowledge of the result from the knowledge of the starting elements. The two processes are sometimes strictly parallel. Mathematical knowledge, if deployed in the manner Bolzano recommended, namely in such a way that theorems are demonstrated by only using the propositions to which they owe their truth (*wissenschaftliche Darstellung*), is a famous example of that parallelism.[1] In other cases things are different, depending on the cognitive capacities of the knowers. According to Leibniz, Adam's life is an example of possible discrepance. Of course, Adam has lived in a manner exactly conform withat what God had planned in creating him, namely in deciding to realize *that* possible Adam and to make him coming into existence. To ourselves, as creatures equipped with a finite, limited power of analysis, the only means of unfolding Adam's notion is to wait for Adam's real life and to look to what he does and what happens to him. Even if Adam's notion or folder implicitely contains everything which happens to him, including what happens to his relatives and to his offspring, we have to wait for the progressive real unfolding of that notion to learn what is implicit in it. In other words, real unfolding is *incompressible* for us and there is no shortcut to knowledge of what happens before it happens. God's epistemic situation is quite different, for He is endowed with an infinite power of analysis : to Him, there is no real question of unfolding whatever, as He sees everything *toto simul*.

Another manner of restating Leibniz's point in a wider, non theological context is to say that there are items such that the exhaustive knowledge of all the elements out of which they arise, along with the exhaustive knowledge of all the rules governing the composition and interaction of these elements, are not convertible into an anticipatory knowledge of these very items : there are no other means, to know their final outcome, than to wait that that outcome really *appears*.

---

[1] Cf [Dubucs-Lapointe 2003] and [Dubucs-Lapointe *in press*]

On the other hand, contemporary complexity theory assumes that the complexity of the unfolded product entirely lies in the complexity of its compressed folder and even that the the size of that folder can be taken as a measure of the complexity of the result. For example, according to Kolmogoroff's seminal idea, the complexity of a string of successive 0's and 1's can be equated to the length of the shortest computer programme able to generate it. Thus, only the complexity of the folder matters and no significance at all is attributed to the complexity of the unfolding process itself : to knowing the folder amounts to knowing the unfolded object that is contained in it in embryonic form, for unfolding never conveys or requires new information.

To discuss the correctness of that complete abandonment of the leibnizian stance, let us keep aside the boring problems that most of the time cloud the argument, namely the problems that have to do with indeterminism or incomputability. I will therefore assume that these side-problems do not arise, i.e. that we have a correct and complete « analytic » knowledge of the basic laws that govern the interactions between the constituents of the system we speak about, and that that knowledge is entirely convertible (maybe by a suitable discretization of evolving equations) into an algorithmic knowledge. Then, the main question is if, under these conditions, there is still any room for the leibnizian idea of incompressible unfolding. I will argue, contrary to the *communis opinio*, that the answer to that question should be positive.

## 2 Unfolding, Predicting, Hyper-predicting

To clarify the argument, let me take a very simple example of kolmogorovian unfolding. The programme

```
10    n := 0
20    Print n
30    n := n+1 mod 2
40    Goto 20
```

generates, when processed, the infinite string *01010101 …*

One can consider that string of digits as the successive states of some changing item $x$ that has the value *0* at its initial state. The programme is the folder of the item and the processing of the programme, including the printing process, is its diachronical unfolding. An interesting feature of that unfolding lies in the fact that shortcuts are possible, permitting prediction. Clearly, the value of $x$ at its $n$-th state is *0* if $n$ is even and it is *1* in the other case. Thus, we can mentally jump to any state of $x$ to know the value of $x$ at that state.

The notion of prediction can be understood in two different senses. In a general sense, to predict (correctly) an event simply amounts to know that the event will occur before it occurs. Prediction in that sense is generally possible as soon as the requirements stated above concerning correctness, completeness and computability of knowledge are satisfied. For in that case it suffices to have at hand computers that run fast enough to get the result before the event happens. Technical advances in computing machinery insure that that general kind of prediction is or could be achieved in most cases, provided the available knowledge meets the three stated requirements just stated. Then, Leibniz's dictum that unfolding is sometimes definitely incompressible and that the best we can do is to wait for the reality, so to speak, seems to be refuted in a deterministic and computable world. Nevertheless, that dictum can be vindicated in another way, namely by referring to a stronger sense of the very notion of prediction. The refutation that has just been outlined for Leibniz's dictum rests on the common idea that the progress in the speed of the computing devices is indefinitely open, and then that unfolding by computer is in principle able to go quicker than the real unfolding of the things themselves. But such an unfolding by computer follows exactly the same track as the real process itself. Shortening is not obtained by a shortcutting the real way, but by running the same process quicker. There is no question of jumping to the end of the process, but only of following the process step-by-step at a greater speed. Therefore, Leibniz's dictum is at best weakened, but not strictly speaking refutated. It would worth to survive if some qualification is added, stating for example that there are such phenomena that the only way of predicting them is to *simulate* by means of computer devices the unfolding process that rise to them. To really undermine Leibniz's dictum, much more is required, namely a cogent argument establishing that prediction is always possible in a stronger sense : not by quick simulation of the unfolding real process, but by bypassing parts of it and jumping more directly to its end. Let me term *strong prediction* or *hyper-prediction* this kind of shortcutting. There are plenty of cases in which prediction can be converted into hyper-prediction. The example above is one of them : one can bypass the unfolding process of $x$ to obtain directly its value at whatever stage of its unfolding. If, and only if, this manoeuver is always possible will, then Leibniz's dictum be undermined. The defenders of the dictum have therefore to show that hyper-prediction is not generally possible, even in a deterministic and computable universe.

# 3 Perspectives Towards a Vindication of Leibniz Dictum

First, a definite mathematical content can certainly attributed to the claim that hyper-prediction is not always available. That mathematical content could be made explicited by defining a non empty class $L$ of recursive functions that are only computable by a step-by-step process (or for which this is the quickest way to compute them) ; namely, *f* belongs to *L* iff the (best) computation of *f(n)* requires the successive computation of *f(0), f(1), … , f(n-1)*. Because of a lack of space, I cannot here enter into the details of the current progress on that track here. Let me just say, to give an idea of the difficulty of the task, that some « natural » inhabitants of the class *L* are, in fact, not suitable. For instance, one could think of the string of the digits of π as a natural candidate for membership to *L*, for it is *prima facie* clear that the value of the *n*-th digit of π cannot be computed before the value of the previous digits have been computed. Intuitively, the digits of π have to be computed in a successive manner and one fails to perceive that it could be possible to compute the value of, say, the thousandth digit of π without to getting beforehand the initial segment *141 592* … Unfortunately, a famous result of computational mathematics obtained in 1995 by Bailey, Bowen and Plouffe [2] has shown that this intuition is misleading.

Thus, instead of entering into the details of the mathematical definition of *L*, let me give some general idea of the (bad) reason why Leibniz's dictum has eventually felt out of favor. This situation essentially comes from the predominance of the standpoint of the « in principle » point of view that has been advocated since the work of Turing and other logicians in the 1930's. The analysis of the notion of algorithmic computability they provided was as so successfull (it has been considered as a « miracle », for the resulting notion was absolute, namely robust and independent of any reference to a particular formal system) that it is has been taken as the ultimate reference to characterize computational capacities. Now it is plainly clear that, for a Turing machine, there cannot exist any sensible difference between prediction and hyper-prediction. The only question that matters for a Turing machine is if it is in principle able to compute some output, and any further question concerning the way or the speed of that computation is of no significance. That standpoint underlies also Kolmogoroff's theory of complexity, that measures the complexity of a string by the size of the shortest computer programme able to produce it, whatever the duration of that production could be : what only matters is the initial folder, not the unfolding process. To vindicate Leibniz's idea we therefore have to bring into play considerations relating to practical or human feasibility, that is, we have to take into account of the frontier between those processes which not only can be carried out *in principle*, but also can be humanly achieved and those which, although they can be carried out *in principle*, cannot be humanly achieved.[3] If one adopts that perspective, it becomes self-evident that the only predictions we can actually do are the predictions for which we can by-pass parts of the development of the appropriate system. As regards the other ones, there is no other way, as Leibniz said or should have said,

[2] [Bailey-Borwein-Plouffe 2004]

[3] Cf [Dubucs 2002]

than to wait for the result of the unfolding or to simulate it.

Returning to the problem from which we started from, the most plausible conjecture concerning cognitive abilities is that, when a system is endowed with them, there are not only *novel*, in the sense that it makes no sense to attribute them to specific components of the system, but also *anomal*, in the sense that it is impossible to explain why the system has these capacities or to hyper-predict that it will have them merely on the only basis of the laws that govern the behavior of its components.

# References

[Bailey-Borwein-Plouffe 1995] D. Bailey, P. Borwein & S. Plouffe, A Rapid Computation of Various Polylogarithmic Constants », in Pi : A Source Book (eds. L. Berggren, J. Borwein & P. Borwein), Springer Verlag, 3th edition, 2004.

[Dubucs 2002] J. Dubucs, Feasibility in Logic, Synthese 132(2002), pp.213-237.

[Dubucs-Lapointe 2003] J. Dubucs and S. Lapointe, Preuves par excellence, Philosophiques 30(2002), pp 219-234..

[Dubucs-Lapointe *in press*] J. Dubucs and S. Lapointe, Bolzano's Alleged Explicativism, Synthese, *im press*.