Non-signaling deterministic models for non-local correlations have to be uncomputable

Ariel Bendersky,[1,2] Gabriel Senno,[1,2] Gonzalo de la Torre,[3] Santiago Figueira,[1,2] and Antonio Acin[3,4]

[1]*Departamento de Computación, FCEN, Universidad de Buenos Aires, Buenos Aires, Argentina*
[2]*CONICET, Argentina*
[3]*ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels, Barcelona, Spain*
[4]*ICREA, Pg. Lluis Companys 23, 08010 Barcelona, Spain*

Quantum mechanics postulates random outcomes. However, a model making the same output predictions but in a deterministic manner would be, in principle, experimentally indistinguishable from quantum theory. In this work we consider such models in the context of non-locality on a device independent scenario. That is, we study pairs of non-local boxes that produce their outputs deterministically. It is known that, for these boxes to be non-local, at least one of the boxes' output has to depend on the other party's input via some kind of hidden signaling. We prove that, if the deterministic mechanism is also algorithmic, there is a protocol that, with the sole knowledge of any upper bound on the time complexity of such algorithm, extracts that hidden signaling and uses it for the communication of information.

PACS numbers: 03.67.-a, 03.65.Ud

Non-locality is one of the defining properties of quantum mechanics, stating that remote systems can have correlations beyond what's expected from classical systems. If one wants to account in a deterministic manner for the non-local correlations that quantum mechanics predicts one must allow for the existence of some kind of 'signaling' mechanism that links distant measurement choices and outcomes. These mechanisms are said to violate *parameter independence* [1]. But, since quantum correlations are non-signaling, such signaling mechanism must be restricted to the so-called hidden variables, and not reach the phenomenological level.

Some examples of deterministic non-local theories are: the hidden variable model with communication of Toner and Bacon [2] and, more prominently, Bohmian mechanics [3]. For those models that use classical communication to mimic non-locality, one can in fact study the amount of communication needed (see, for example, [4–6]).

In all these theories, although the outputs at each round of a Bell test are determined given the inputs and the hidden variables, the sequence of such variables is chosen randomly. In this article we consider the class of deterministic models for non-local correlations in which the hidden variables are not chosen randomly but pseudorandomly, that is, by means of an algorithmic process, and study whether this has operational consequences. Note that the sequence of hidden variables for a given experiment is experimentally unaccessible. Here, we wonder whether imposing these sequences to be computable has any operational consequences when considering only the properties observed in the Bell experiment. This is equivalent to considering boxes that on each round of a Bell test define their outputs by evaluating a computable function of all the inputs and the number of round.

Our main result is to prove that deterministic models reproducing non-local correlations must be uncomputable to exclude that those correlations can lead to observable signaling. In other words, for any deterministic and computable model that uses hidden signalling to reproduce non-local correlations, we provide a protocol that, by having access only to the observed variables in the Bell test and with the knowledge of any upper bound on the model's time computational complexity, is able to extract the hidden signaling and use it for communication.

There are a few previous results in this direction. First, it is argued in [7] that the possibility to algorithmically compress the outputs of measurements over certain bipartite quantum states would allow for signalling. Here however we obtain our result in a device-independent scenario, that is, without assuming quantum mechanics. Second, in [8], computability of the outputs implying signaling is proven for the PR-box and for any non-local boxes violating the chained Bell inequality or winning any pseudotelepathy game (non-local games having a quantum strategy which wins with probability one). Our result is that this is true for *any* non-local correlations. We provide an explicit communication protocol. Finally, the question of the computability of the sequences of outputs, although without relating it to the possibility of signaling, has also been studied for contextuality scenarios [9], through the localization of *value indefinite* observables [10].

This paper is organized as follows: first we introduce the scenario that we are considering. Then we briefly review the tools from computability theory that we need to resort to in order to prove our main result. Finally, we present our results.

The scenario. – We consider a standard Bell scenario. For the sake of simplicity, we present our results for the simplest Clauser-Horne-Shimony-Holt (CHSH) Bell test [11] where we have two parties, Alice and Bob, each

one with a box that has a binary input and a binary output. The extension to other scenarios is straightforward.

Our goal is to study deterministic and computable models that reproduce non-local correlations. This means that there are computable functions $A, B : \{0,1\}\times\{0,1\}\times\mathbb{N}\to\{0,1\}$, representing Alice's and Bob's boxes respectively, such that $A(x,y,n)$ [resp. $B(x,y,n)$] defines the output at the n-th round of Alice's [resp. Bob's] box when Alice's input is x and Bob's input is y. See Fig. 1 for a schematic representation. As for the non-locality of the boxes, we formalize it through the following definition:

Definition 1. *A pair of boxes A, B with binary inputs and outputs is* non-local *iff whenever the sequences of inputs $(x_i)_{i\in\mathbb{N}}$ and $(y_i)_{i\in\mathbb{N}}$ are independent tosses of a fair coin, the sequences of outputs $(a_i)_{i\in\mathbb{N}}$ and $(b_i)_{i\in\mathbb{N}}$ are such that*

$$p(a,b|x,y) := \lim_{n\to\infty} \frac{4\cdot\#\{i<n \mid (x_i,y_i,a_i,b_i)=(x,y,a,b)\}}{n}$$

violates a Bell inequality with probability 1.

FIG. 1: Schematic representation of the scenario considered. Two distant observers, Alice and Bob, run a Bell test by implementing measurements on two systems. The observed correlations are described by a hidden-signaling mechanism plus computable functions determining the outputs given the inputs at each round n.

Note that Definition 1 is general enough to cover the usual non-deterministic scenario as well.

As we said in the introduction, because we are looking at deterministic boxes generating non-local correlations, their outputs have to depend on each other's input. Since the boxes are computable, this is the only information they need to share, as any other necessary data can be computed from the inputs. It is important to note that, although it seems that our toy model is signaling, and therefore it would not come as a surprise that Alice can signal to Bob, this is not the case. The model uses signaling for its internal workings but this does not necessarily imply that Alice and Bob can send information to each other. For instance, if one does not impose that the functions A and B are computable, one can perfectly simulate quantum mechanics without violating the no-signalling principle. Bohm's theory [3] constitutes an example of this simulation.

It is easy to see that, if the dependence between distant inputs and outputs happens in only finitely many rounds, the boxes are essentially local. Therefore, we have that:

Lemma 1. *If A and B are a pair of deterministic non-local boxes, then for infinitely many values of n, there is x such that $A(x,0,n)\neq A(x,1,n)$ or there is y such that $B(0,y,n)\neq B(1,y,n)$.*

In the following, A and B are pair of computable deterministic non-local boxes and, without loss of generality, we make the next assumption: for infinitely many n,

$$\exists y\in\{0,1\} \text{ such that } B(0,y,n)\neq B(1,y,n), \quad (1)$$

or in other words, for infinitely many values of n, the value of x can be determined from the output of B with the suitable choice of y. Therefore, if Alice knew how to compute B, they could trivially signal from Alice to Bob (Alice just inputs her message, and in the rounds in which Bob's output depends on Alice's input, he can reconstruct her input). The situation we want to study is when B is unknown.

What we show next is that, with the assumption that B is a computable function, one can actually devise a protocol to transmit one-way information from Alice to Bob with the sole knowledge of some upper bound on the time computational complexity of B. Before showing the protocol, we need to introduce some concepts from computability theory.

Tools from computability theory. – The main ingredient in the protocol we are about to describe is that of *learning* a target computable function $f:\mathbb{N}\to\{0,1\}$ from a given class $\mathcal{C}$ by looking at finite prefixes of f [12], where a prefix of f of length n is defined as the finite sequence $f(0), f(1), f(2), ..., f(n-1)$. To *learn* a computable function f means to discover a program computing f. The learner chooses successive candidate programs which are consistent with the finite prefix of f already seen. During this process, the program may change until it stabilizes in a certain program computing f.

The learning procedure for $\mathcal{C}$ depends on the class $\mathcal{C}$ but not on the specific function f to be learnt. Not all classes of functions are learnable: it is well known that the class of all computable functions is not learnable [13]. On the contrary, for every computable function t, the class of functions computable in time $O(t)$ is learnable. The learnability of the class $\mathcal{C}$ of functions running in time $O(t)$ follows from the fact that such class is computably enumerable, i.e. a computer can list programs computing all functions in the class (possibly with repetitions). As shown in Fig. 2, for a target function $f\in\mathcal{C}$ the learner will make its guess by picking the first program in the enumeration whose outputs coincide with the already seen bits of f. Since one of the programs computing f is in the enumeration (every program running in time $O(t)$ is listed), and the learner only moves forward in that list (it picks the first program whose output coincides with every seen bit), at some point it will find one program computing f.

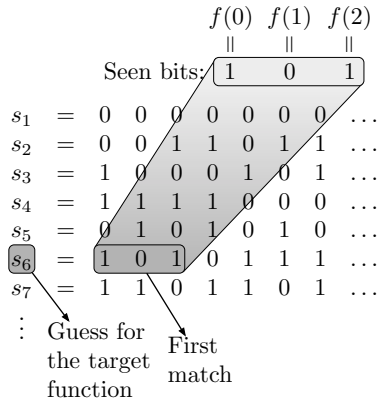

FIG. 2: Suppose we want to learn a $\{0,1\}$-valued function f and let $\{s_i\}_{i\in\mathbb{N}}$ be a computable enumeration of programs computing $\{0,1\}$-valued functions, which run in time $O(t)$. The i-th row represents the sequence $s_i(0), s_i(1), s_i(2), \ldots$. After seeing $f(0) = 1$, $f(1) = 0$ and $f(2) = 1$, the guess for the target function is defined as (the index of) the first program whose outputs match those values (in the example, the guess is s_6).

The signaling protocol. – We are now in position to present our main result: the construction of a protocol that allows two parties, Alice and Bob, to signal if the following three conditions hold: (i) their boxes are nonlocal, in the sense of Definition 1. That is, that the outputs of the boxes would lead to an estimated probability distribution that is nonlocal and no-signaling, such as those of quantum theory; (ii) the internal mechanism of the boxes is computable and is assisted by a hidden signaling mechanism; (iii) Alice and Bob know a computable time bound function t for Bob's box computation.

The key idea of the protocol is for Bob to perform a learnability scheme on the outputs of his box. Once he has learned function B, he uses the rounds n such that (1) holds to reconstruct Alice's input (see Lemma 1) and, thus, received the signal. There are three issues that we will need to deal with in this approach:

- In order for Bob to learn a program to compute the function B, he needs to know Alice's inputs x, at least until B has been learned.
- In general, Bob will not be able to tell when he has effectively learned B, as for any given prefix, there are infinitely many computable sequences reproducing it.
- For every round n, Bob has access only to the value of B for the pair of inputs (x_n, y_n) selected in the round.

To cope with the first two issues, Alice and Bob will alternate between *learning* rounds and *signaling* rounds. The former are rounds in which they know both parties inputs (they are pre-established) and are used by Bob to learn function B. The later are rounds that are used to send a message from Alice to Bob assuming B is already known. Choosing the pre-established inputs in a *sufficiently* random manner will allow them to cope with the third issue. We now give a description of the protocol and sketch the proof of its soundness. For a more detailed technical explanation, see the Supplemental Material.

As a first step of the protocol, Alice and Bob have to assume a time complexity class for the boxes' functions. This is done by choosing a computable function t that tells them how long it takes for Bob's box to compute its output on round n. To perform the alternation aforementioned, they share a sequence S whose symbols are either a pair of bits, or an integer between 1 and m, where m is the length of the message that Alice wants to communicate.

- **Learning rounds.** If the n-th symbol of S is a pair (x_n, y_n), Alice inputs x_n and Bob inputs y_n. Since Bob knows both inputs and his output $B(x_n, y_n, n)$, he uses every such round to perform a learnability scheme in order to make a better guess for his function B. Specifically, he looks for the first program p running in time $O(t)$ such that for all (x_j, y_j) in the first n symbols of S, we have $p(x_j, y_j, j) = B(x_j, y_j, j)$. Notice that the construction of the candidate program p is based only on the values corresponding to the learning rounds.
- **Signaling rounds.** If the n-th symbol of S is a number $k_n \in \{1 \ldots m\}$, Alice inputs the k_n-th bit of her message and Bob checks with his candidate program p for B if Alice's input can be inferred from his output by choosing a proper input, i.e., if there is a value of y such that $p(0, y, n) \neq p(1, y, n)$. If there is no such y, Bob inputs 0 to his box and proceed to the next round. Else, suppose $p(0, y, n) = u$ and $p(1, y, n) = 1 - u$. Bob inputs y to his box. If the output is u then Bob postulates that the k_n-th symbol of the message from Alice is 0, else he postulates that it is 1.

For this protocol to be sound, it suffices that the following properties hold:

(P_1) There exists a number of round n_0 such that for all $n \geq n_0$, Bob's candidate program p at stage n is correct, that is $p(x, y, n) = B(x, y, n)$ for all $x, y \in \{0, 1\}$. In other words, the learning process converges to B.

(P_2) For the k-th bit of Alice's message and for infinitely many n, $S(n) = k \in \mathbb{N}$ and $B(0, y, n) \neq B(1, y, n)$ for some $y \in \{0, 1\}$, i.e. the signaling mechanism happens for infinitely many rounds. For every k, after finitely many rounds, Bob's postulate for bit k of Alice's message is forever correct.

When (P_1) and (P_2) hold, the values obtained on the signaling rounds can be incorrect only for finitely many rounds (until B has been learnt). Therefore, on the long run Bob is able to identify each bit from Alice's message as the value that showed up the most for that bit.

Now, whether (P_1) and (P_2) hold depends on the choice of the shared sequence S. For example, let us consider the case in which $S(n)$ are independent and identically distributed (i.i.d.) random variables. To see that (P_1) holds we proceed by contraposition. Suppose that the learning procedure stabilizes in one of the finitely many programs p appearing before one computing B in the enumeration, and whose outputs differ from those of B in infinitely many inputs (x, y, m). This would imply that for almost all rounds n in which S dictates learning, that n is not one of the infinitely many m for which $p(x, y, m) \neq B(x, y, m)$ for some (x, y). It is easy to see that the probability of this happening when choosing the learning rounds n at random is zero. To see that (P_2) holds it suffices to observe that amongst the infinitely many n where (1) is true, the probability that S picks finitely many of them to signal the k-th bit of the message is zero.

However, letting S be i.i.d. random variables would make our argument too weak, as it would mean that Alice and Bob have access to randomness, a non-computable resource, namely a random variable, to test models of nature that are assumed to use only computable functions. On the other hand, choosing a too simple sequence for S does not work. It is not hard to see that if S is chosen such that, for instance, it indicates learning in the odd rounds and signaling in the even, the learning could converge to a program that coincides with B in almost all odd positions but, for the even positions, it outputs, say, the negation of B (this program, of course, also runs in time $O(t)$). One can then expect that some notion of computable yet sufficiently random is needed for the protocol to work. The question is: can we find a computable sequence S that does the job?

In general, no easily predictable sequence S is suitable. However, one can consider the notion of *t-randomness* [14–16], in which the degree of randomness is defined with respect to an adversary whose computing time is bounded by some computable function t. Then, the idea is to run the previous protocol with a sequence that, while being computable, "looks like a random variable" for the devices with the assumed bounded computational power. Now, the question is to operationally define what "looking like a random variable" means. Intuitively, the notion of a sequence random with respect to a time bound t can be related to the impossibility of the adversary to *predict* its symbols using a machine running in time $O(t)$.

More precisely, the adversary has a computer program that computes a function M, such that, given the first k symbols of the sequence S (i.e, $S_0 S_1 ... S_{k-1}$), it tells what part of its capital the adversary has to bet on each possible next symbol S_k. In other words, M is a betting strategy. The sequence S is called t–random if there is no such betting strategy, computable in time $O(t)$, that makes the adversary win an unbounded amount of money.

In the Supplemental Material we show how to the notion of t-randomness captures this intuition and provides a thorough proof that the protocol works even when S is a computable t-random sequence. It is important to note that, from a program for t, one can compute a t-random sequence in time $O(t(n) \cdot log(t(n)) \cdot n^3)$ (see e.g. [16]). Therefore, for our protocol to work, Alice and Bob will only need to know a computable time bound t for B, since they will be able to compute a suitable S from a program for t.

It is important to note that, without any knowledge of B, there is no a priori bound on the time it takes Bob to determine Alice's message with high enough confidence. Nonetheless, since this time is finite, there exists some finite distance for which the communication allowed by our protocol is superluminal. For instance, if it takes Bob M rounds to find out Alice's message and each round takes a time T, then if they are at a distance cTM, the message is obtained before a light signal from Alice could reach Bob.

It could be argued that imposing a bound on the time complexity of Alice and Bob's boxes (which are nothing but an abstraction of what nature is doing to choose the outputs) is a strong requirement. However, since the number of computational steps per second that can be performed by a system of mass m is upper bounded by $2mc^2/\pi\hbar$ [17], this is not only a requirement of our protocol but a reasonable physical assumption.

Discussion. – Our protocol shows that correlated systems that would have violated a Bell inequality if were used for a standard Bell test (i.e., with random inputs), can be used to signal if assumed to be computable and a time (or space) bound for their computational complexity is known in advance. The main consequence of this is that we are left with the following consequences: either Bell-violating systems cannot be computable, or if Alice and Bob guess properly a complexity class larger than the one used by the computable systems, they can signal in either way using the previous protocol. Our result implies that, under the well established assumption that no observable signaling exists, we need to accept the existence of truly unpredictable physical processes.

It is worth mentioning that our result doesn't go into conflict with the different interpretations of quantum mechanics. All of them predict random outputs, which are not allowed by our model. In the Copenhagen interpretation, the measurement process is postulated as random, whereas, for example, Bohmian mechanics is deterministic but postulates initial conditions that are randomly distributed and fundamentally unknowable.

This work was supported by the ERC CoG QITBOX, an AXA Chair in Quantum Information Science, the Spanish MINECO (Project FOQUS FIS2013-46768-P, Severo Ochoa grant SEV-2015-0522 and FPI FIS2010-14830), grants ANPCyT-PICT-2013-2011, ANPCyT-PICT-2011-0365, UBACyT 20020110100025, the Laboratoire International Associé "INFINIS", the Fundacion Cellex, the Generalitat de Catalunya (SGR875), and the John Templeton Foundation.

[1] A. Shimony, in *Quantum Concepts in Space and Time*, edited by R. Penrose and C. J. Isham (New York; Oxford University Press, 1986), pp. 182–203.
[2] B. F. Toner and D. Bacon, Physical Review Letters **91**, 187904 (2003).
[3] D. Bohm, Physical Review **85**, 166 (1952).
[4] O. Regev and B. Toner, SIAM Journal on Computing **39**, 1562 (2009).
[5] Y. Shi and Y. Zhu, SIAM Journal on Computing **38**, 753 (2008).
[6] J. Degorre, M. Kaplan, S. Laplante, and J. Roland, Quantum information & computation **11**, 649 (2011).
[7] U. Yurtsever, Complexity **6**, 27 (2000).
[8] S. Wolf, Phys. Rev. A **92**, 052102 (2015), URL `http://link.aps.org/doi/10.1103/PhysRevA.92.052102`.
[9] A. A. Abbott, C. S. Calude, J. Conder, and K. Svozil, Physical Review A **86**, 062109 (2012).
[10] A. A. Abbott, C. S. Calude, and K. Svozil, Journal of Mathematical Physics **56**, 102201 (2015), arXiv:1503.01985, URL `http://dx.doi.org/10.1063/1.4931658`.
[11] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, Phys. Rev. Lett. **23**, 880 (1969), URL `http://link.aps.org/doi/10.1103/PhysRevLett.23.880`.
[12] T. Zeugmann and S. Zilles, Theoretical Computer Science **397**, 4 (2008).
[13] E. M. Gold, Information and control **10**, 447 (1967).
[14] R. G. Downey and D. R. Hirschfeldt, *Algorithmic randomness and complexity* (Springer Science & Business Media, 2010).
[15] A. Nies, *Computability and randomness*, vol. 51 (Oxford University Press, 2009).
[16] S. Figueira and A. Nies, Theory of Computing Systems **56**, 439 (2015).
[17] S. Lloyd, Nature **406**, 1047 (2000).