Computability theory and Bell non-locality

Antonio Acín[1], Ariel Bendersky[2], Gonzalo de La Torre[1],
Santiago Figueira[2] y *Gabriel Senno*[1]

[1]ICFO - The Institute of Photonic Sciences.
[2]Computer Science Department, FCEyN, University of Buenos Aires.

International Symposium on New Frontiers in Quantum Correlations
(ISNFQC18)
Kolkata, January 2018

Abstract

Two results relating computability theory and Bell non-locality:

- Inputs: *The computability loophole*.

Abstract

Two results relating computability theory and Bell non-locality:

- Inputs: *The computability loophole*.
- Outputs: *Computability + non-locality* $\implies$ *signaling*.

The computability loophole

The CHSH game

$(x, y) \in_R \{0,1\}^2$

x y

$a \in \{0,1\}$

$b \in \{0,1\}$

$$a \oplus b = xy$$

The CHSH game

$(x, y) \in_{\mathrm{R}} \{0,1\}^2$

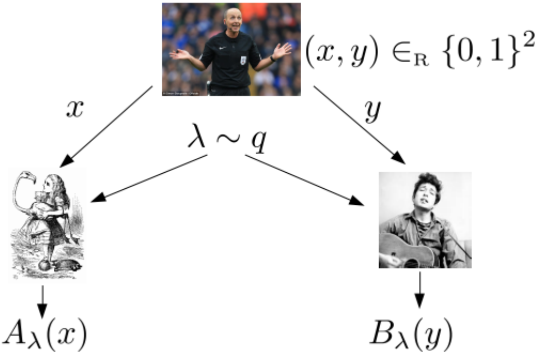
x $\quad$ y

$a \in \{0,1\}$ $\qquad$ $b \in \{0,1\}$

$$a \oplus b = xy$$

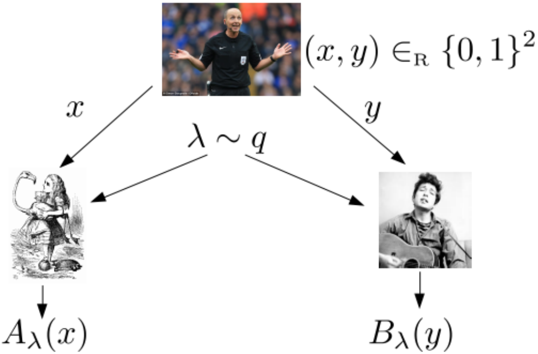
$$p_{win}^{\text{CHSH}} := \frac{1}{4} \sum_{a,b} \left[p(a \neq b | 1,1) + \sum_{(x,y) \neq (1,1)} p(a = b | x, y) \right]$$

Local strategies



$$p(a, b|x, y) = \sum_{\lambda} q(\lambda)\delta_{a=A_\lambda(x)}\delta_{b=B_\lambda(y)}.$$

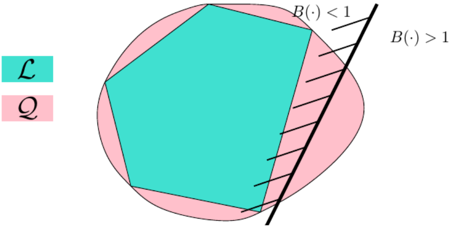
Local strategies



$$p(a, b|x, y) = \sum_\lambda q(\lambda)\delta_{a=A_\lambda(x)}\delta_{b=B_\lambda(y)}.$$

$$p_{win}^{\mathrm{CHSH}} \leq \frac{3}{4}, \text{ for every local strategy.}$$

Bell inequalities



- The CHSH inequality

$$p(=|P_1,P_1)+p(=|P_1,P_2)+p(=|P_2,P_1)+p(\neq|P_2,P_2)\leq 3$$

  is an example of a Bell inequality.
- In general,

$$\sum_{a,b,x,y} B_{a,b,x,y}\,p(a,b|x,y)\leq B_l.$$

Quantum strategies

- Quantum strategy:

$$p(a,b|x,y) = \langle\psi|\Pi_a^x\Pi_b^y|\psi\rangle$$

with $|\psi\rangle \in \mathcal{H}$, $\sum_a \Pi_a^x = \sum_b \Pi_b^y = \mathbb{I}_{\mathcal{H}}$ and $[\Pi_a^x, \Pi_b^y] = 0$.

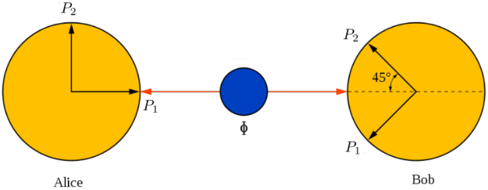
Quantum strategies

- Quantum strategy:

$$p(a,b|x,y) = \langle\psi|\Pi_a^x \Pi_b^y|\psi\rangle$$

with $|\psi\rangle \in \mathcal{H}, \sum_a \Pi_a^x = \sum_b \Pi_b^y = \mathbb{I}_{\mathcal{H}}$ and $[\Pi_a^x, \Pi_b^y] = 0$.

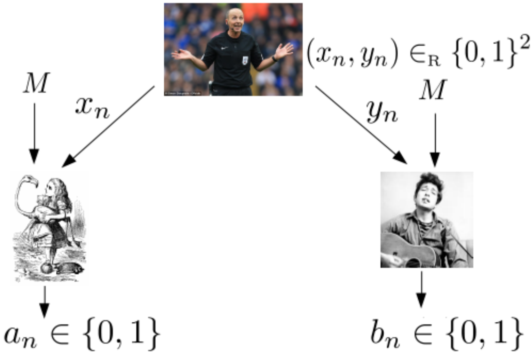
- For the CHSH game, preparing $|\psi^-\rangle =: \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ and measuring in the following spin directions:



it is easy to see that, $p_{win}^{\text{CHSH}} = \cos^2(\pi/8) \approx 0,85$.

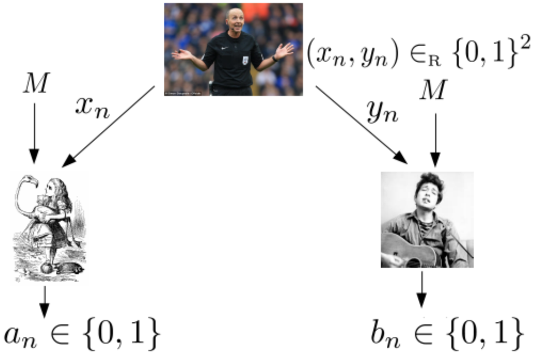
Memory scenario

$$M = [(x_1, y_1), \dots, (x_{n-1}, y_{n-1})]$$



Memory scenario

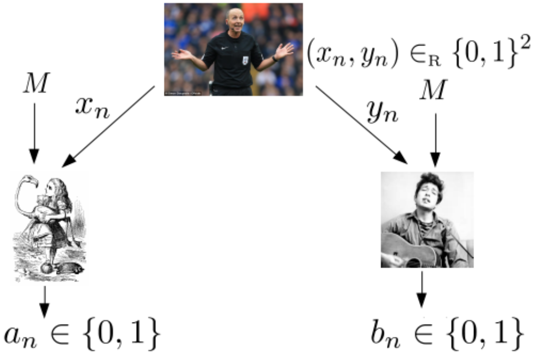
$$M = [(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})]$$



- Local strategies:
  $p(a_n, b_n | x_n, y_n) = \sum_\lambda q(\lambda) \delta_{a_n = A_\lambda(x_n, M)} \delta_{b_n = B_\lambda(y_n, M)}$.

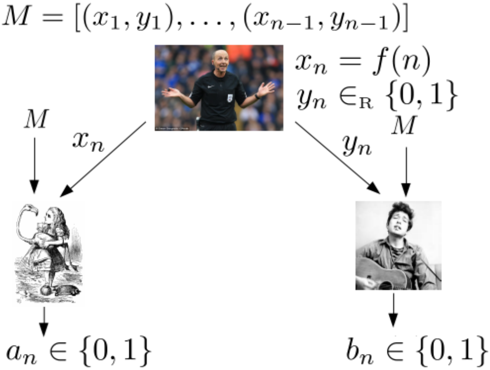
Memory scenario

$$M = [(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})]$$



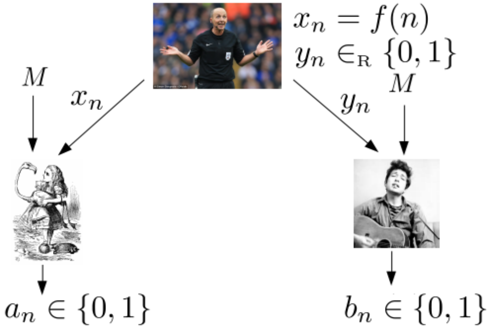
- Local strategies:
  $p(a_n, b_n | x_n, y_n) = \sum_\lambda q(\lambda) \delta_{a_n = A_\lambda(x_n, M)} \delta_{b_n = B_\lambda(y_n, M)}$.
- $p_{win}^{\text{CHSH}} \leq 3/4$ [Barrett et al., Phys. Rev. A 66, 042111, 2002].

Memory scenario with computable inputs



Memory scenario with computable inputs

$$M = [(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})]$$



Theorem ([Bendersky, Senno, de la Torre, Figueira and Acín. **PRL** 116, 230402, 2016])

If the referee in the CHSH game with memory chooses (at least) one of the players' questions using a computable function $f : \mathbb{N} \to \{0, 1\}$*, there is a perfect local strategy (independent of* f*).*

Computable functions

- A function $f : \mathbb{N} \to \{0, 1\}$ is said to be *computable* if there is a program $\mathcal{P}$ that on input n outputs $f(n)$.

Computable functions

- A function $f : \mathbb{N} \to \{0, 1\}$ is said to be *computable in (deterministic) time $O(T)$* for some $T : \mathbb{N} \to \mathbb{N}$ if there is a program $\mathcal{P}$ that on input n outputs $f(n)$ after (at most) $c \cdot T(\log n)$ computational steps for some c and suff. large n.

Computable functions

- A function $f : \mathbb{N} \to \{0, 1\}$ is said to be *computable in (deterministic) time $O(T)$* for some $T : \mathbb{N} \to \mathbb{N}$ if there is a program $\mathcal{P}$ that on input n outputs $f(n)$ after (at most) $c \cdot T(\log n)$ computational steps for some c and suff. large n.
- A class of computable functions $\mathcal{C} = \{f_0, f_1, \ldots\}$ is said to be *computably enumerable* if there is a program $\mathcal{P}$ that on inputs n outputs (the code of) a program that computes f_n.

Computable functions

- A function $f : \mathbb{N} \to \{0, 1\}$ is said to be *computable in (deterministic) time $O(T)$* for some $T : \mathbb{N} \to \mathbb{N}$ if there is a program $\mathcal{P}$ that on input n outputs $f(n)$ after (at most) $c \cdot T(\log n)$ computational steps for some c and suff. large n.
- A class of computable functions $\mathcal{C} = \{f_0, f_1, \dots\}$ is said to be *computably enumerable* if there is a program $\mathcal{P}$ that on inputs n outputs (the code of) a program that computes f_n.
- For every computable $T : \mathbb{N} \to \mathbb{N}$, the class $\mathcal{C}_T$ of functions computable in time $O(T)$ is computably enumerable. This include the well-known complexity classes P, NP, EXP, BQP.

Computable functions

- A function $f : \mathbb{N} \to \{0, 1\}$ is said to be *computable in (deterministic) time $O(T)$* for some $T : \mathbb{N} \to \mathbb{N}$ if there is a program $\mathcal{P}$ that on input n outputs $f(n)$ after (at most) $c \cdot T(\log n)$ computational steps for some c and suff. large n.
- A class of computable functions $\mathcal{C} = \{f_0, f_1, \dots\}$ is said to be *computably enumerable* if there is a program $\mathcal{P}$ that on inputs n outputs (the code of) a program that computes f_n.
- For every computable $T : \mathbb{N} \to \mathbb{N}$, the class $\mathcal{C}_T$ of functions computable in time $O(T)$ is computably enumerable. This include the well-known complexity classes P, NP, EXP, BQP.
- But, the class of all computable functions is *not* computably enumerable.

Predicting computable functions

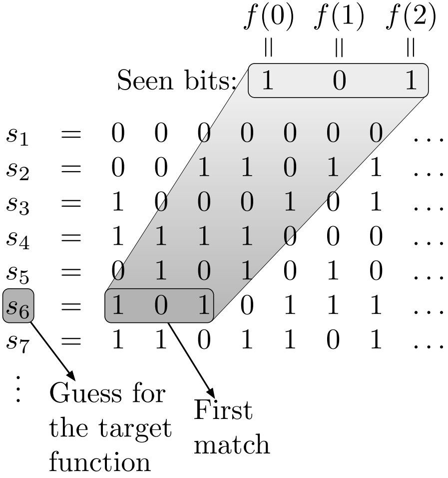
Functions in computably enumerable classes can be predicted in the following sense:

For every computably enumerable class $\mathcal{C}$ of computable functions there is a program $\mathcal{P}$ (called a *predictor for* $\mathcal{C}$) such that for every $f \in \mathcal{C}$,

$$(\exists n_0)(\forall n \geq n_0)\ f(n) = \mathcal{P}([f(0), \ldots, f(n-1)])$$

Predicting computable functions

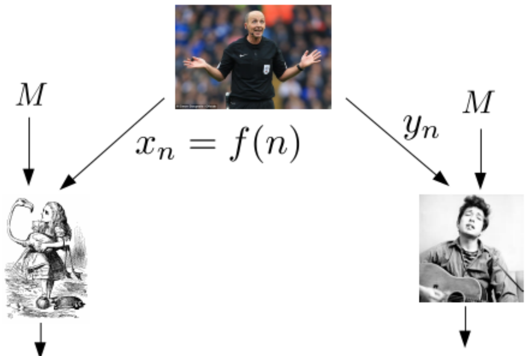
The predictor works as follows,



Perfect local strategy

Let $T : \mathbb{N} \to \mathbb{N}$ be some computable function, $f \in \mathcal{C}_T$ and $\mathcal{P}$ a predictor for $\mathcal{C}_T$.

$$M = [(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})]$$



0

$$\begin{cases} 1 & \text{if } \mathcal{P}([x_1, \ldots, x_{n-1}]) = 1 \wedge y_n = 1 \\ 0 & \text{o.w.} \end{cases}$$

Technical details of the model

- Provided $f \in \mathcal{C}_T$, if Bob uses a predictor $\mathcal{P}$ for $\mathcal{C}_T$, after finitely many rounds he will start predicting x_i correctly.

Technical details of the model

- Provided $f \in \mathcal{C}_T$, if Bob uses a predictor $\mathcal{P}$ for $\mathcal{C}_T$, after finitely many rounds he will start predicting x_i correctly.
  - However, how many rounds it will take him is uncomputable.

Technical details of the model

- Provided $f \in \mathcal{C}_T$, if Bob uses a predictor $\mathcal{P}$ for $\mathcal{C}_T$, after finitely many rounds he will start predicting x_i correctly.
  - However, how many rounds it will take him is uncomputable.
- Nevertheless, the number M of prediction errors he make until that time is small.

Technical details of the model

- Provided $f \in \mathcal{C}_T$, if Bob uses a predictor $\mathcal{P}$ for $\mathcal{C}_T$, after finitely many rounds he will start predicting x_i correctly.
  - However, how many rounds it will take him is uncomputable.
- Nevertheless, the number M of prediction errors he make until that time is small.
  - If $f(n)$ has a k-bits program, $M \leq O(\log(k))$.

Technical details of the model

- Provided $f \in \mathcal{C}_T$, if Bob uses a predictor $\mathcal{P}$ for $\mathcal{C}_T$, after finitely many rounds he will start predicting x_i correctly.
  - However, how many rounds it will take him is uncomputable.
- Nevertheless, the number M of prediction errors he make until that time is small.
  - If $f(n)$ has a k-bits program, $M \leq O(\log(k))$.
- Also, the prediction algorithm is (almost) as efficient as f.

Technical details of the model

- Provided $f \in \mathcal{C}_T$, if Bob uses a predictor $\mathcal{P}$ for $\mathcal{C}_T$, after finitely many rounds he will start predicting x_i correctly.
  - However, how many rounds it will take him is uncomputable.
- Nevertheless, the number M of prediction errors he make until that time is small.
  - If $f(n)$ has a k-bits program, $M \leq O(\log(k))$.
- Also, the prediction algorithm is (almost) as efficient as f.
  - If f is computable in $O(T)$ time, then $\mathcal{P}$ runs in $O(T \cdot \log(T))$.

Loophole

- In the context of non-locality experiments, this result can be seen as a loophole, i.e. an experimental situation allowing non-communicating classical devices to generate statistics violating a Bell inequality.

Loophole

- In the context of non-locality experiments, this result can be seen as a loophole, i.e. an experimental situation allowing non-communicating classical devices to generate statistics violating a Bell inequality.
  - Can it be closed?

Loophole

- In the context of non-locality experiments, this result can be seen as a loophole, i.e. an experimental situation allowing non-communicating classical devices to generate statistics violating a Bell inequality.
  - Can it be closed?
- Famous loophole-free experiments of 2015 use QRNGs. If we assume quantum theory, they are free from the loophole (with probability 1). However,

Loophole

- In the context of non-locality experiments, this result can be seen as a loophole, i.e. an experimental situation allowing non-communicating classical devices to generate statistics violating a Bell inequality.
  - Can it be closed?
- Famous loophole-free experiments of 2015 use QRNGs. If we assume quantum theory, they are free from the loophole (with probability 1). However,
  - Rather circular to assume a non-local theory when testing non-locality.

Loophole

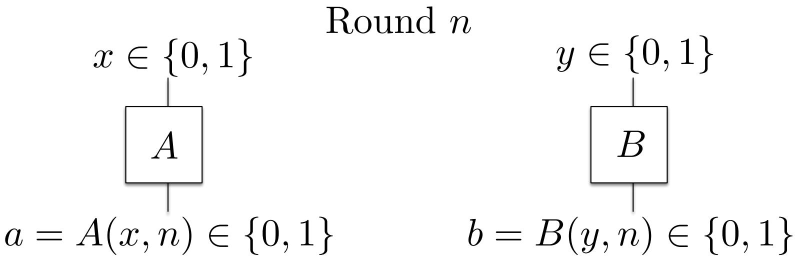
- In the context of non-locality experiments, this result can be seen as a loophole, i.e. an experimental situation allowing non-communicating classical devices to generate statistics violating a Bell inequality.
  - Can it be closed?
- Famous loophole-free experiments of 2015 use QRNGs. If we assume quantum theory, they are free from the loophole (with probability 1). However,
  - Rather circular to assume a non-local theory when testing non-locality.
  - Experimentally unverifiable.

Loophole

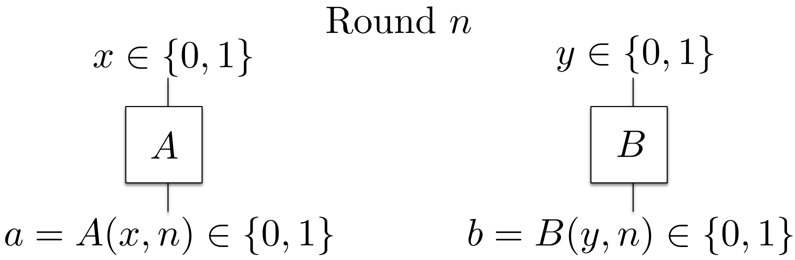
- In the context of non-locality experiments, this result can be seen as a loophole, i.e. an experimental situation allowing non-communicating classical devices to generate statistics violating a Bell inequality.
  - Can it be closed?
- Famous loophole-free experiments of 2015 use QRNGs. If we assume quantum theory, they are free from the loophole (with probability 1). However,
  - Rather circular to assume a non-local theory when testing non-locality.
  - Experimentally unverifiable.
- Nevertheless, the result I will talk about next implies that, under reasonable assumptions, the outputs from QRNGs are, in fact, uncomputable.

Computable non-locality allows for signaling

Deterministic boxes in a CHSH scenario

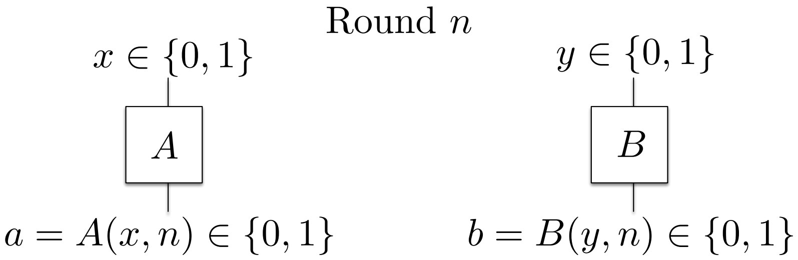


Deterministic boxes in a CHSH scenario



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

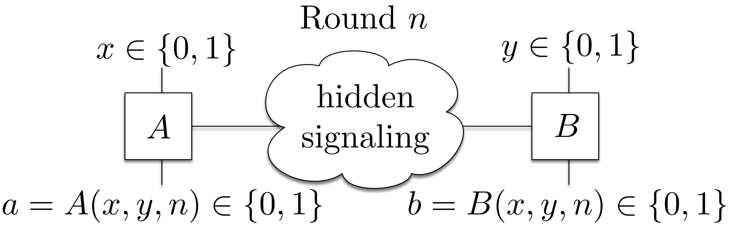
Deterministic boxes in a CHSH scenario



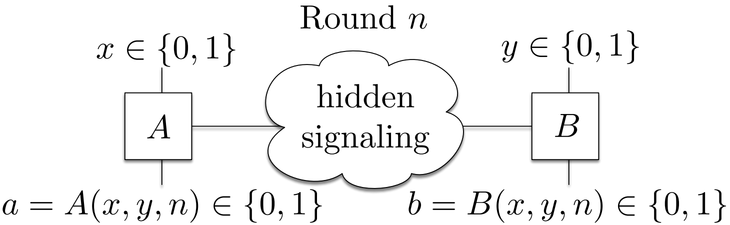
$$p(a,b|x,y) := \lim_{n\to\infty} \frac{|\{i \le n \mid x_i = x, y_i = y, a_i = a, b_i = b\}|}{|\{i \le n \mid x_i = x, y_i = y\}|}.$$

We will say that A and B are *non-local* if, when the inputs are chosen uniformly at random, p violates a Bell inequality with probability 1.

Determinism $\wedge$ non-locality $\implies$ violation of Parameter Independence



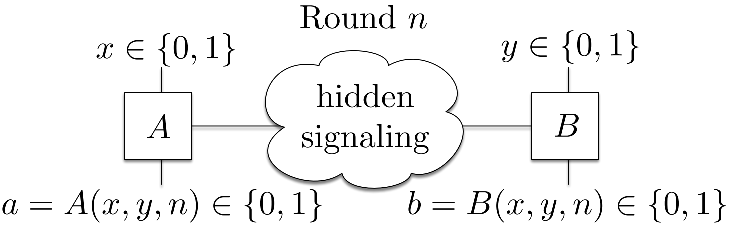
Determinism $\wedge$ non-locality $\implies$ violation of Parameter Independence



Lemma

If A and B are non-local,
$\exists^\infty n\ [\exists x\ A(x,0,n) \neq A(x,1,n) \vee \exists y\ B(0,y,n) \neq B(1,y,n)]$.

Determinism $\wedge$ non-locality $\implies$ violation of Parameter Independence



Lemma

If A and B are non-local,
$\exists^\infty n\ [\exists x\ A(x,0,n) \neq A(x,1,n) \vee \exists y\ B(0,y,n) \neq B(1,y,n)]$.

Observation

Violation of Parameter Independence doesn't imply signaling (e.g.: Bohmian mechanics, Toner & Bacon, etc).

Using that hidden signaling for communicating

W.l.o.g., let's assume that

$$\exists^{\infty} n \exists y \; B(0, y, n) \neq B(1, y, n). \tag{1}$$

Observation

If Alice and Bob had access to B, i.e. if they knew how to **compute** it, they could easily communicate: they just wait for the ns that verify (1) and, with the right choice of y_n, Bob can tell x_n. Thus, we assume B is hidden (it's Nature's secret). Can it be kept that way?

Using that hidden signaling for communicating

W.l.o.g., let's assume that

$$\exists^\infty n \exists y \; B(0,y,n) \neq B(1,y,n). \tag{1}$$

Observation

If Alice and Bob had access to B, i.e. if they knew how to **compute** it, they could easily communicate: they just wait for the ns that verify (1) and, with the right choice of y_n, Bob can tell x_n. Thus, we assume B is hidden (it's Nature's secret). Can it be kept that way?

Main Result

We give a protocol which, if B is a **computable function**, allows Alice to send a message to Bob with the sole knowledge of a bound on the computational complexity of B.

Learnability in the limit

A class computable functions $\mathcal{C}$ is *learnable in the limit* if there exists a program $\mathcal{P}$ (called a *learner for* $\mathcal{C}$) such that for every $f : \mathbb{N} \to \mathbb{N} \in \mathcal{C}$, there exists m such that for every $m \geq n$, on input (some coding of) $\{(0, f(0)), \ldots, (m, f(n))\}$ $\mathcal{P}$ outputs (the code of) a program that computes f.

Learnability in the limit

A class computable functions $\mathcal{C}$ is *learnable in the limit* if there exists a program $\mathcal{P}$ (called a *learner for* $\mathcal{C}$) such that for every $f : \mathbb{N} \to \mathbb{N} \in \mathcal{C}$, there exists m such that for every $m \geq n$, on input (some coding of) $\{(0, f(0)), \ldots, (m, f(n))\}$ $\mathcal{P}$ outputs (the code of) a program that computes f.

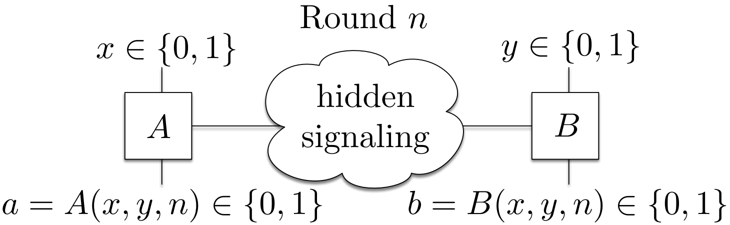
- Every computably enumerable class of computable functions is learnable in the limit.

Learnability in the limit

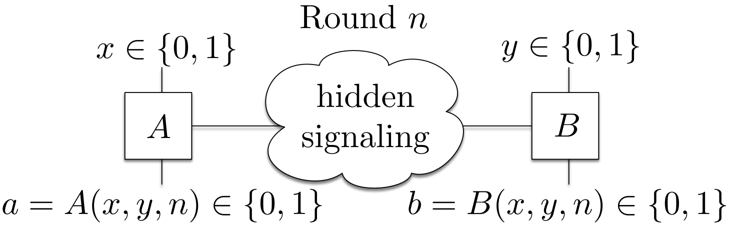
A class computable functions $\mathcal{C}$ is *learnable in the limit* if there exists a program $\mathcal{P}$ (called a *learner for* $\mathcal{C}$) such that for every $f : \mathbb{N} \to \mathbb{N} \in \mathcal{C}$, there exists m such that for every $m \geq n$, on input (some coding of) $\{(0, f(0)), \ldots, (m, f(n))\}$ $\mathcal{P}$ outputs (the code of) a program that computes f.

- Every computably enumerable class of computable functions is learnable in the limit.
- The class of all computable functions is not learnable in the limit.

Learning in the limit allows communication



Learning in the limit allows communication



Restrictions

- 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.
- For every n, Bob will only see the value of B for just one pair of inputs (x_n, y_n).
- Bob will not be able to tell when he has effectively learned B.

The protocol $\mathcal{P}(t, m, S)$

Inputs:

1. a computable non-decreasing function t
2. the size of Alice's message m
3. a sequence $S \in \{(0,0),(0,1),(1,0),(1,1),1,\ldots,m\}^\infty$

The protocol $\mathcal{P}(t, m, S)$

Inputs:

1. a computable non-decreasing function t
2. the size of Alice's message m
3. a sequence $S \in \{(0,0), (0,1), (1,0), (1,1), 1, \ldots, m\}^\infty$

On each round n:

1. if $S(n) = (x, y)$, Alice inputs x and Bob inputs y. Then, Bob uses a learner for the class of functions computable in time $O(t)$ on input $((x_{i_1}, y_{i_1}, B(x_{i_1}, y_{i_1}, i_1)) \ldots (x, y, B(x, y, n)))$ (with i_k being the past learning rounds) to update his guess $\widetilde{B}$ of a program that computes B (**Learning round**).
2. if $S(n) = i \in \{1, \ldots, m\}$, Alice inputs the ith bit of her message, a_i and Bob y s.t. $\widetilde{B}(0, y, n) \neq \widetilde{B}(1, y, n)$ and makes the output of his box his new guess for a_i. If there is no such y, he inputs 0 (**Signaling round**).

Soundness of the protocol

For $\mathcal{P}(t, m, S)$ to be sound, it suffices that the following properties hold:

1. There exists a number of round n such that for all $m \geq n$, and $x, y \in \{0, 1\}$, we have $\widetilde{B}(x, y, m) = B(x, y, m)$, i.e. the learning process converges to B.
2. For every bit i of Alice's message and for infinitely many n, $S(n) = i \in \mathbb{N}$ and $\exists y \in \{0, 1\}.B(0, y, n) \neq B(1, y, n)$.

Alternating randomly

Is not hard to see that properties 1 and 2 hold with probability 1 when S_i are independent and uniformly distributed random variables.

Alternating randomly

Is not hard to see that properties 1 and 2 hold with probability 1 when S_i are independent and uniformly distributed random variables.

Pseudorandom Nature

It is unfair to allow the players randomness when we are considering a pseudrandom Nature.

Alternating randomly

Is not hard to see that properties 1 and 2 hold with probability 1 when S_i are independent and uniformly distributed random variables.

Pseudorandom Nature

It is unfair to allow the players randomness when we are considering a pseudrandom Nature.

Can we de-randominize?

T-randomness

Definition

A sequence S is called T-random if there is no strategy computable in time $O(T)$ to win unbounded money betting successively on the symbols of S starting from with some finite initial capital.

T-randomness

Definition

A sequence S is called T-random if there is no strategy computable in time $O(T)$ to win unbounded money betting successively on the symbols of S starting from with some finite initial capital.

Observation

We can compute T-random sequences in time $O(T(n) \cdot \log(T(n)) \cdot n^3)$ [Figueira, Nies, Theo. Comp. Sys. 56, 439 (2015)].

T-randomness

Definition

A sequence S is called T-random if there is no strategy computable in time $O(T)$ to win unbounded money betting successively on the symbols of S starting from with some finite initial capital.

Observation

We can compute T-random sequences in time $O(T(n) \cdot \log(T(n)) \cdot n^3)$ [Figueira, Nies, Theo. Comp. Sys. 56, 439 (2015)].

Theorem ([Bendersky, Senno, de la Torre, Figueira, Acín. Phys. Rev. Lett. 118, 130401, 2017])

If S is T-random, properties 1 and 2 hold.

Future directions

1. Randomness amplification

Future directions

1. Randomness amplification
   - Quantum nonlocality can be used to certify randomness amplification (a task which is classically impossible [Santha and Vazirani, J. Comput. Syst. Sci. 33(1), 1986]).

Future directions

1. Randomness amplification
   - Quantum nonlocality can be used to certify randomness amplification (a task which is classically impossible [Santha and Vazirani, J. Comput. Syst. Sci. 33(1), 1986]).
   - Algorithmic randomness cannot be *computably* amplified [Miller, Adv. Math. 226(1), 373-384 (2011)].

Future directions

1. Randomness amplification
   - Quantum nonlocality can be used to certify randomness amplification (a task which is classically impossible [Santha and Vazirani, J. Comput. Syst. Sci. 33(1), 1986]).
   - Algorithmic randomness cannot be *computably* amplified [Miller, Adv. Math. 226(1), 373-384 (2011)].
   - In the signaling result, we show the outputs of non-local boxes are not computable from the inputs. What else can be said about their relative degree of uncomputability (or, more generally, about their relative level of algorithmic randomness)?

Future directions

1. Randomness amplification
   - Quantum nonlocality can be used to certify randomness amplification (a task which is classically impossible [Santha and Vazirani, J. Comput. Syst. Sci. 33(1), 1986]).
   - Algorithmic randomness cannot be *computably* amplified [Miller, Adv. Math. 226(1), 373-384 (2011)].
   - In the signaling result, we show the outputs of non-local boxes are not computable from the inputs. What else can be said about their relative degree of uncomputability (or, more generally, about their relative level of algorithmic randomness)?
2. Computability of the set of quantum correlations.
   - Very recent breaktrough results by Slofstra [arXiv:1606.03140, arXiv:1703.08618].