BELL'S THEOREM IN AUTOMATA THEORY

MICHAEL ZIRPEL, MZ@MZIRPEL.DE

ABSTRACT. Bell's theorem is reformulated and proved in the abstract mathematical terms of automata theory, avoiding physical or ontological notions. It is stated that no pair of sequential machines can reproduce in its output data the statistical results of a quantum physical Bell test experiment if neither machine depends on the other's input.

Keywords: Bell's theorem, Bell test experiment, Bell-CHSH inequality, sequential machine, probabilistic algorithm

1. INTRODUCTION AND OVERVIEW

Bell's theorem [Bell, 1964] is a crucial topic in the foundations of physics. Despite its discovery 60 years ago, many scientific papers continue to discuss its meaning and implications (see Myrvold et al., 2024 and the references therein). Realizations of the thought experiment on which the theorem is based, the quantum physical Bell test experiment, are conducted every year with new variations (e.g., Storz et al., 2023). The Nobel Prize in Physics was awarded in 2022 to some designers of the first groundbreaking.

In the Bell test experiment, remote measurements on a pair of particles flying apart show strange nonlocal correlations, and Bell's theorem states therefore that "no physical theory, which is *realistic* and also *local* in a specified sense, can agree with all of the statistical implications of Quantum Mechanics" [Shimony, 2016]. However, the meaning of the theorem and the assumptions of its proof are not easy to understand.

To provide a new perspective, we reformulate Bell's theorem for automata theory in abstract mathematical terms, avoiding physical or ontological notions. For this purpose, the arrangement of an ideal Bell test experiment is represented by a pair of sequential machines, providing a straightforward mathematical model.[1] Each machine receives an input from a separate operator or an independent random generator, and then produces an output controlled by a probabilistic or deterministic algorithm. The theorem states that no such pair can reproduce the statistical results of a quantum physical Bell test experiment in its output data if neither machine depends on the other's input. Therefore, any computer simulation following this scheme requires information transmission between the two parties to achieve the quantum physical result statistics.

After presenting a short sketch of the physical Bell test experiment, we describe its simulation using sequential machines and prove the theorem.

[1] The idea of using computer to explain the content of Bell's theorem is not new (e.g., Gill, 2014) and inspired this paper. However, we consider abstract automata to prove a theorem.

2. Bell test experiment

For our purposes, a coarse sketch of an *ideal* Bell test experiment[2] without any physical details is sufficient.[3]

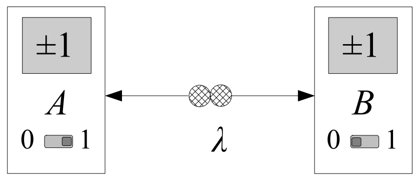

FIGURE 2.1. Bell test experiment

A single run of the experiment starts by sending a pair of particles λ from a source, each particle in another direction, to measurement devices A and B (Fig. 2.1). Each measurement device can perform one of two different measurements. A switch with two positions 0, 1, symbolizes the selection by the experimenter (or an independent random generator) of the measurement to be performed. Therefore, on each side, one of two possible quantities A_0, A_1, respectively B_0, B_1, is measured. A display shows the measurement result, -1 or $+1$, in any case.

This arrangement enables the measurement of one of the four possible pairs (A_0,B_0), (A_0,B_1), (A_1,B_0), (A_1,B_1) in each run of the experiment, from which the corresponding product A_0B_0, A_0B_1, A_1B_0, or A_1B_1 is computed, which also has the value -1 or 1. The procedure is repeated with different random measurement selections. After several runs, the mean values of the products $\overline{A_0B_0}$, $\overline{A_0B_1}$, $\overline{A_1B_0}$, $\overline{A_1B_1}$ are used to compute the expression

$$\overline{A_0B_0}+\overline{A_0B_1}+\overline{A_1B_0}-\overline{A_1B_1}$$

which in the long run should approximate the theoretical given expectation value

$$\langle E_{CHSH}\rangle = \langle A_0B_0+A_0B_1+A_1B_0-A_1B_1\rangle = \langle A_0B_0\rangle+\langle A_0B_1\rangle+\langle A_1B_0\rangle-\langle A_1B_1\rangle.$$

In probability theory, for any four random variables A_0, A_1, B_0, B_1, with the image $\{-1,1\}$ on a measure space $(\Omega,\mathscr{A})$, the absolute value of this expectation value is bounded according to the *Bell-CHSH inequality* [Clauser et al., 1969] by the value 2 (cf. App. B), i.e., for any probability measure μ on $(\Omega,\mathscr{A})$

$$|\langle E_{CHSH}\rangle| \leq 2.$$

[2]In real experiments, particle losses, detection errors, and many other factors must be taken into account, opening several loopholes to question the experimental verification of Bell's theorem. We will consider only the ideal case and trust the experimentalists who claim that all physical loopholes are closed (cf. Hensen et al. [2015]). The more metaphysical loophole, opened by *superdeterminism* with doubts about the free will of experimenters and the independence of random generators, cannot be closed (cf. Hossenfelder and Palmer, 2020).

[3]A good introduction to the physical thought experiment and the theorem is given in Bell [1981].

However, quantum theory predicts values above 2 for the measurable quantities[4] of an ideal Bell test experiment, which is confirmed by the results of countless real Bell test experiments since Aspect et al. [1982]. Therefore, the *violation* of the Bell-CHSH inequality is an *experimental fact* of quantum physics.[5]

3. SIMULATION OF THE BELL TEST WITH SEQUENTIAL MACHINES

The theory of finite probabilistic sequential machines (FPSMs)[6] is best suited to describe a computer simulation of an ideal Bell test experiment. An FPSM receives an input symbol, then changes possibly its internal state, and produces an output symbol controlled by a probabilistic or deterministic algorithm. The sets of input symbols, output symbols, and states are denoted by I, O, S.

To simulate the Bell experiment, the input and output of the FPSM can be defined straightforward. The input is a triple $i = (a, b, \lambda)$, where a and b represent the experimenters' choices, 0 or 1, and $\lambda \in \Lambda$ some not further specified properties of the particle pair[7]. The output is a pair $o = (A, B)$, where A and B represent the possible measurement results, -1 or 1, as shown in Figure 3.1.

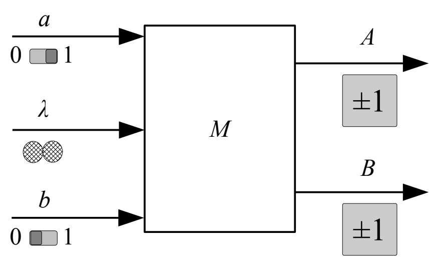

FIGURE 3.1. Simulation of the Bell test experiment using an FPSM

The algorithm of the machine is given through a *probabilistic machine function*, which determines the probability $p(o, t \mid i, s) \in [0, 1]$ to obtain output o and state t after input i in state s, with

$$\sum_{o \in O} \sum_{t \in S} p(o, t \mid i, s) = 1$$

for all $i \in I, s \in S$. For a deterministic machine, all these probabilities have a value of 0 or 1 (c.f. App. A). In Example 1, several probabilistic machine functions are listed.

Each simulation starts with an initial machine state, which is selected according to the probability distribution

$$p_0 \colon S \to [0, 1] \text{ with} \sum_{s \in S} p_0(s) = 1.$$

[4] In quantum theory the four measurable quantities are called *observables* and are represented by self-adjoint operators on a Hilbert space with the spectrum $\{-1, 1\}$. The quantum theoretical expectation value of the corresponding Bell-CHSH expression has as upper bound the *Tsirelson bound* $2\sqrt{2}$ (Cirel'son, 1980).

[5] Landau [1987] shows that observables with the spectrum $\{-1, 1\}$ violating the Bell-CHSH inquality are ubiquitous in the set of observables of any quantum system. However, a compound system taken to a pair of separated pieces is needed to perform the Bell test experiment.

[6] A brief introduction is provided in App. A.

[7] e.g. a pair of random numbers whose details are irrelevant for our further considerations

Therefore, the full FPSM is defined formally by

$$M=(I,O,S,p_0,p)=(\{0,1\}^2\times\Lambda,\{-1,1\}^2,S,p_0,p).$$

3.1. **Protocol for the Bell test simulation.** The FPSM simulation has fewer constraints than the original Bell test experiment: the output data may be totally random and not the result of four different measurements. Therefore, we must use a slightly more general notation. Furthermore, we must fix the simulation protocol to determine the statistics.

In each run of the simulation[8], the FPSM is prepared to an initial state $s\in S$ according to the probability distribution p_0 (using an independent random generator if p_0 is not deterministic).

Then, the input data is collected: An input symbol $\lambda\in\Lambda$, representing the particle data, is entered, randomly selected according to a probability distribution p_Λ with $\sum_{\lambda\in\Lambda}p_\Lambda(\lambda)=1$ using an independent random generator. Furthermore, the input symbols $a,b\in\{0,1\}$ are entered by the operators (or independent random generators) without knowledge of (or dependence on) input λ or the initial state s.[9]

After entering the input, the machine algorithm produces two output symbols $A,B\in\{-1,1\}$. The product of the output symbols $A\cdot B\in\{-1,1\}$ is recorded together with the corresponding pair of input symbols (a,b)

$$AB\mid_{a,b}.$$

After a series of runs (or multiple series for the different combinations of the values of a, b), the mean values of the recorded products for the different input symbols are computed

$$\overline{AB}\mid_{0,0},\ \overline{AB}\mid_{0,1},\ \overline{AB}\mid_{1,0},\ \overline{AB}\mid_{1,1}$$

with the value of the Bell-CHSH expression

$$\overline{E}_{CHSH}=\overline{AB}\mid_{0,0}+\overline{AB}\mid_{0,1}+\overline{AB}\mid_{1,0}-\overline{AB}\mid_{1,1}.$$

In the special case of a proper Bell test simulation, this value should approximate the theoretical given expectation value of the Bell-CHSH expression

$$\langle E_{CHSH}\rangle=\langle A_0B_0\rangle+\langle A_0B_1\rangle+\langle A_1B_0\rangle-\langle A_1B_1\rangle$$

in the long run. In general, the corresponding theoretical expression for the FPSM is a sum of conditional expectation values

$$\tilde{E}_{CHSH}=\langle AB\rangle\mid_{0,0}+\langle AB\rangle\mid_{0,1}+\langle AB\rangle\mid_{1,0}-\langle AB\rangle\mid_{1,1}$$

which can be computed by

$$\langle AB\rangle\mid_{a,b}=\sum_{A\in\{-1,1\}}\sum_{B\in\{-1,1\}}AB\cdot q(A,B|a,b)$$

with the *output probability function*

$$q(A,B\mid a,b)=\sum_{\lambda\in\Lambda}\sum_{s\in S}\sum_{t\in S}p(A,B,t\mid a,b,\lambda,s)p_\Lambda(\lambda)p_0(s).$$

The value of $\tilde{E}_{CHSH}$ lies always in the closed interval $[-4,4]$. We say *an FPSM fulfills the Bell-CHSH inequality if*

$$\left|\tilde{E}_{CHSH}\right|\leq 2,$$

[8]An alternative protocol, where this initial state preparation occurs only for the start of a series of trials, is discussed in Note 2 below.

[9]See Note 4below.

otherwise, it *violates the Bell-CHSH inequality*.

Example 1. The following table gives the probabilistic machine functions and expectation values of several simple FPSMs for the Bell test simulation. All these machine functions do not depend on internal states or λ-input, so we simply assume $S = \{s_\varnothing\}$ with $p_0(s_\varnothing) = 1$ and $\Lambda = \{\lambda_\varnothing\}$ with $p_\Lambda(\lambda_\varnothing) = 1$. Therefore, the probabilistic machine function is equal to the output probability function $q(A,B \mid a,b) = p(A,B,s_\varnothing \mid a,b,\lambda_\varnothing,s_\varnothing)$. $\delta_{x,y}$ denotes the Kronecker symbol with the values 1 if $x = y$ and 0 otherwise. [10]

	$q(A,B \mid a,b) = p(A,B,s_\varnothing \mid a,b,\lambda_\varnothing,s_\varnothing)$	$\langle AB\rangle\|_{0,0}$	$\langle AB\rangle\|_{0,1}$	$\langle AB\rangle\|_{1,0}$	$\langle AB\rangle\|_{1,1}$	$\tilde{E}_{CHSH}$
M_1	$\frac{1}{4}$	0	0	0	0	0
M_2	$\delta_{A,1}\delta_{B,1}$	1	1	1	1	2
M_3	$\delta_{A,1}\delta_{B,1-2ab}$	1	1	1	-1	4
M_4	$\frac{1}{2}\delta_{A,1}\delta_{B,1-2ab} + \frac{1}{2}\delta_{A,-1}\delta_{B,2ab-1}$	1	1	1	-1	4
M_5	$\frac{2-\sqrt{2}}{8} + \frac{\sqrt{2}}{8}(2\delta_{A,B} + 2ab - 4ab\delta_{A,B})$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$	$2\sqrt{2}$
M_6	$\frac{1}{2}\delta_{a,b}(\delta_{A,1}\delta_{B,-1} + \delta_{A,-1}\delta_{B,1}) + \frac{1}{4}\delta_{a,1-b}$	-1	0	0	-1	0

TABLE 1. Probabilistic machine functions and expectations of Example FPSMs

The output of FPSM M_1 is evenly distributed and uncorrelated random, whereas M_2 gives the constant output $(1,1)$. FPSM M_3 changes the otherwise constant output B from 1 to -1 in the case that a and b have the value 1. FPSM M_4 is an evenly weighted statistical mixture of M_3 and its negative counterpart and simulates a Popescu-Rohrlich box (cf. Popescu and Rohrlich, 1994). FPSM M_5 is a simulation of an ideal quantum physical Bell test experiment, with maximal violation of the Bell-CHSH inequality at the Tsirelson-bound. FPSM M_6 represents the EPR-Bohm experiment, where the random output is perfectly (anti-)correlated for equal input values $a = b$, but perfectly uncorrelated otherwise.
The FPSMs M_2 and M_3 are deterministic because the probabilistic machine function has only the values 0 or 1. The value of $\tilde{E}_{CHSH}$ indicates that the FPSMs M_1, M_2, M_6 fulfill the Bell-CHSH inequality, whereas M_3, M_4, M_5 violate it.

3.2. **Pairing sequential machines.** FPSM M_5 demonstrates that it is possible to simulate an ideal Bell experiment with an FPSM and obtain the same statistical results as those obtained with a quantum physical experiment. However, to for Bell's theorem, the simulation must be performed with a pair of two separated FPSMs, as sketched by the circuit diagram in Fig. 3.2).

Both FPSMs[11]

$$M^{\mathfrak{a}} = (\{0,1\}^2 \times \Lambda, \{-1,1\}, S^{\mathfrak{a}}, p_0^{\mathfrak{a}}, p^{\mathfrak{a}}),$$
$$M^{\mathfrak{b}} = (\{0,1\}^2 \times \Lambda, \{-1,1\}, S^{\mathfrak{b}}, p_0^{\mathfrak{b}}, p^{\mathfrak{b}})$$

receive the same input[12] and each one gives only a single output value $A \in \{-1,1\}$, respectively $B \in \{-1,1\}$. The machine pair can be considered as one *composite* FPSM

$$M^{\mathfrak{ab}} = (\{0,1\}^2 \times \Lambda, \{-1,1\}^2, S^{\mathfrak{a}} \times S^{\mathfrak{b}}, p_0^{\mathfrak{a}} p_0^{\mathfrak{b}}, p^{\mathfrak{a}} p^{\mathfrak{b}})$$

[10]The free web app https://bell.qlwi.de can be used to perform Bell test simulations with these FPSMs on any PC, tablet, or smartphone.

[11]We use the letters $\mathfrak{a}$,$\mathfrak{b}$ in the upper position only as label not as exponent or summation index.

[12]We assume only one λ-input for both machines and not a pair $\lambda^{\mathfrak{a}}, \lambda^{\mathfrak{b}}$ as with the particles in the original experiment. In the following proof, this assumption makes no difference.

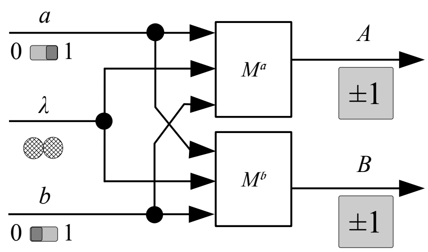

FIGURE 3.2. Simulation of the Bell test experiment using two FPSMs

which gives a pair of output values after each input. The probabilistic machine function and the initial probability distribution are the product of the corresponding components, reflecting the *independence* of the partial machines $M^{\mathfrak{a}}$ and $M^{\mathfrak{b}}$.[13] Therefore, not every FPSM can be replaced by a composite pair.

Example 2. The FPSMs M_1, M_2, M_3 of Example 1 can be replaced by a composite pair. If we assume $S^{\mathfrak{a}} = \{s_0^{\mathfrak{a}}\}$, $S^{\mathfrak{b}} = \{s_0^{\mathfrak{b}}\}$ and $s_0^{\mathfrak{ab}} = (s_0^{\mathfrak{a}}, s_0^{\mathfrak{b}})$ with $p_0^{\mathfrak{ab}}(s_0) = 1$, then $M_1 = (\{0,1\}^2 \times \{\lambda_0\}, \{-1,1\}^2, \{s_0\}, p_1, 1)$ can be replaced by $M_1^{\mathfrak{ab}}$ with the probabilistic machine functions $p_1^{\mathfrak{a}} = \frac{1}{2}$ and $p_1^{\mathfrak{b}} = \frac{1}{2}$ because $p_1 = p_1^{\mathfrak{a}} p_1^{\mathfrak{b}}$. Similarly, M_2 can be replaced by $M_2^{\mathfrak{ab}}$ with $p_2^{\mathfrak{a}} = \delta_{A,1}$ and $p_2^{\mathfrak{b}} = \delta_{B,1}$, and M_3 by $M_3^{\mathfrak{ab}}$ with $p_3^{\mathfrak{a}} = \delta_{A,1}$ and $p_3^{\mathfrak{b}} = \delta_{B,1-2ab}$.

3.3. **Functional independence from remote inputs and the Bell-CHSH inequality.** Now, we consider the case in which the machine function $p^{\mathfrak{a}}$ of the composite FPSM $M^{\mathfrak{ab}}$ does not depend on input b and the machine function $p^{\mathfrak{b}}$ does not depend on input a. In this case, the vertical connections can be removed from the circuit diagram (dotted lines in Fig. 3.3).

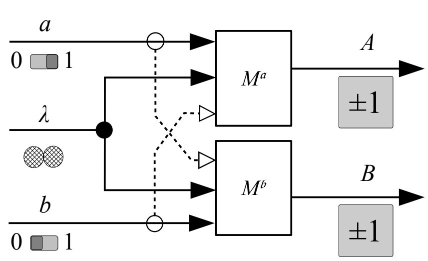

FIGURE 3.3. Simulation using two FPSMs without dependence on remote inputs

[13]The machine functions represent fixed algorithms, which control each machine, separated and independent from the other machine. However, the initial states of the machines can be prepared together, so we will allow non-product initial probability distributions in the following.

The Bell test simulation follows the protocol described in Section 3.1. This allows the joint initial state preparation of the composite machine pair according to an arbitrary probability distribution $p_0^{\mathfrak{ab}}$. Therefore, we do *not* assume $p_0^{\mathfrak{ab}} = p_0^{\mathfrak{a}} p_0^{\mathfrak{b}}$ for $M^{\mathfrak{ab}}$.

Lemma 3. *If $p^{\mathfrak{a}}$ depends not on selection input b and $p^{\mathfrak{b}}$ depends not on selection input a, i.e.,*

$$p^{\mathfrak{a}}(A, t^{\mathfrak{a}} \mid a, b, \lambda, s^{\mathfrak{a}}) = p^{\mathfrak{a}}(A, t^{\mathfrak{a}} \mid a, 0, \lambda, s^{\mathfrak{a}}) = p^{\mathfrak{a}}(A, t^{\mathfrak{a}} \mid a, 1, \lambda, s^{\mathfrak{a}}),$$

$$p^{\mathfrak{b}}(B, t^{\mathfrak{b}} \mid a, b, \lambda, s^{\mathfrak{b}}) = p^{\mathfrak{b}}(B, t^{\mathfrak{b}} \mid 0, b, \lambda, s^{\mathfrak{b}}) = p^{\mathfrak{b}}(B, t^{\mathfrak{b}} \mid 1, b, \lambda, s^{\mathfrak{b}})$$

for all $a, b \in \{0,1\}$; $\lambda \in \Lambda$;$A, B \in \{-1,1\}$;$s^{\mathfrak{a}}, t^{\mathfrak{a}} \in S^{\mathfrak{a}}$;$s^{\mathfrak{b}}, t^{\mathfrak{b}} \in S^{\mathfrak{b}}$, then any composite FPSM

$$M^{\mathfrak{ab}} = (\{0,1\}^2 \times \Lambda, \{-1,1\}^2, S^{\mathfrak{a}} \times S^{\mathfrak{b}}, p_0^{\mathfrak{ab}}, p^{\mathfrak{a}} p^{\mathfrak{b}})$$

fulfills the Bell-CHSH inequality

$$\left|\tilde{E}_{CHSH}\right| \leq 2$$

in the Bell test simulation.

Proof. The conditional expectations are

$$\langle AB \rangle \mid_{a,b} = \sum_{\lambda \in \Lambda} \sum_{s^{\mathfrak{a}} \in S^{\mathfrak{a}}} \sum_{s^{\mathfrak{b}} \in S^{\mathfrak{b}}} p_{\Lambda}(\lambda) p_0^{\mathfrak{ab}}(s^{\mathfrak{a}}, s^{\mathfrak{b}}) \langle AB \rangle_{a,b,\lambda,s^{\mathfrak{a}},s^{\mathfrak{b}}}$$

with

$$\langle AB \rangle_{a,b,\lambda,s^{\mathfrak{a}},s^{\mathfrak{b}}} = \sum_{A \in \{-1,1\}} \sum_{B \in \{-1,1\}} \sum_{t^{\mathfrak{a}} \in S^{\mathfrak{a}}} \sum_{t^{\mathfrak{b}} \in S^{\mathfrak{b}}} AB \cdot p^{\mathfrak{a}}(A, t^{\mathfrak{a}} \mid a, 0, \lambda, s^{\mathfrak{a}}) p^{\mathfrak{b}}(B, t^{\mathfrak{b}} \mid 0, b, \lambda, s^{\mathfrak{b}})$$

for all $a, b \in \{0,1\}$; $\lambda \in \Lambda, s^{\mathfrak{a}} \in S^{\mathfrak{a}}, s^{\mathfrak{b}} \in S^{\mathfrak{b}}$. The latter can be interpreted as the output product expectation value for a fixed input and initial state. Reordering gives

$$\langle AB \rangle_{a,b,\lambda,s^{\mathfrak{a}},s^{\mathfrak{b}}} = \langle A \rangle_{a,\lambda,s^{\mathfrak{a}}} \langle B \rangle_{b,\lambda,s^{\mathfrak{b}}} \tag{3.1}$$

with

$$\langle A \rangle_{a,\lambda,s^{\mathfrak{a}}} = \left(\sum_{A \in \{-1,1\}} \sum_{t^{\mathfrak{a}} \in S^{\mathfrak{a}}} A \cdot p^{\mathfrak{a}}(A, t^{\mathfrak{a}} \mid a, 0, \lambda, s^{\mathfrak{a}}) \right),$$

$$\langle B \rangle_{b,\lambda,s^{\mathfrak{b}}} = \left(\sum_{B \in \{-1,1\}} \sum_{t^{\mathfrak{b}} \in S^{\mathfrak{b}}} B \cdot p^{\mathfrak{b}}(B, t^{\mathfrak{b}} \mid 0, b, \lambda, s^{\mathfrak{b}}) \right).$$

These expressions are the output expectation values of $M^{\mathfrak{a}}$ and $M^{\mathfrak{b}}$ with fixed $a, \lambda, s^{\mathfrak{a}}$, respectively $b, \lambda, s^{\mathfrak{b}}$, and lie in the interval $[-1,1]$. So (according to App. B) the absolute value of the expression

$$E_{\lambda,s^{\mathfrak{a}},s^{\mathfrak{b}}} =$$

$$\langle A \rangle_{0,\lambda,s^{\mathfrak{a}}} \langle B \rangle_{0,\lambda,s^{\mathfrak{b}}} + \langle A \rangle_{0,\lambda,s^{\mathfrak{a}}} \langle B \rangle_{1,\lambda,s^{\mathfrak{b}}} + \langle A \rangle_{1,\lambda,s^{\mathfrak{a}}} \langle B \rangle_{0,\lambda,s^{\mathfrak{b}}} - \langle A \rangle_{1,\lambda,s^{\mathfrak{a}}} \langle B \rangle_{1,\lambda,s^{\mathfrak{b}}} \tag{3.2}$$

will be less or equal 2 for all $\lambda \in \Lambda, s^{\mathfrak{a}} \in S^{\mathfrak{a}}, s^{\mathfrak{b}} \in S^{\mathfrak{b}}$. Hence,

$$\begin{aligned}
\left|\tilde{E}_{CHSH}\right| &= \left| \langle AB \rangle |_{0,0} + \langle AB \rangle |_{0,1} + \langle AB \rangle |_{1,0} - \langle AB \rangle |_{1,1} \right| \\
&= \Big| \sum_{\lambda \in \Lambda} \sum_{s^{\mathfrak{a}} \in S^{\mathfrak{a}}} \sum_{s^{\mathfrak{b}} \in S^{\mathfrak{b}}} E_{\lambda, s^{\mathfrak{a}}, s^{\mathfrak{b}}} \, p_{\Lambda}(\lambda) p_0^{\mathfrak{ab}}(s^{\mathfrak{a}}, s^{\mathfrak{b}}) \Big| \\
&\leq \sum_{\lambda \in \Lambda} \sum_{s^{\mathfrak{a}} \in S^{\mathfrak{a}}} \sum_{s^{\mathfrak{b}} \in S^{\mathfrak{b}}} \left| E_{\lambda, s^{\mathfrak{a}}, s^{\mathfrak{b}}} \right| p_{\Lambda}(\lambda) p_0^{\mathfrak{ab}}(s^{\mathfrak{a}}, s^{\mathfrak{b}}) \\
&\leq \sum_{\lambda \in \Lambda} \sum_{s^{\mathfrak{a}} \in S^{\mathfrak{a}}} \sum_{s^{\mathfrak{b}} \in S^{\mathfrak{b}}} 2 \, p_{\Lambda}(\lambda) p_0^{\mathfrak{ab}}(s^{\mathfrak{a}}, s^{\mathfrak{b}}) = 2.
\end{aligned}$$

□

Example 4. The machine functions of the composite FPSMs $M_1^{\mathfrak{ab}}$, $M_2^{\mathfrak{ab}}$ in Example 2 are independent from any inputs, so they fulfill the Bell-CHSH inequality.

3.4. **Bell's theorem.** The validity of the Bell-CHSH inequality for any composite FPSM $M^{\mathfrak{ab}}$, defined as above, is a logical consequence of the functional independence of $p^{\mathfrak{a}}$ from input b and $p^{\mathfrak{b}}$ from input a. Therefore, the violation of this inequality implies that there is some *functional dependence* instead.

Theorem 5. *For any composite FPSM $M^{\mathfrak{ab}}$, defined as above, which violates the Bell-CHSH inequality in the Bell test simulation, the probabilistic machine function $p^{\mathfrak{a}}$ depends on the selection input b or the probabilistic machine function $p^{\mathfrak{b}}$ depends on the selection input a.*

In this case, the circuit diagram must contain at least one of the vertical connections represented by the dotted lines in Fig. 3.3.

Example 6. The composite FPSM $M_3^{\mathfrak{ab}}$ in Example 2 violates the Bell-CHSH inequality. The machine function $p_3^{\mathfrak{b}} = \delta_{B,1-2ab}$ depends on the input a.

3.5. **Notes.**

(1) The expectation values $\langle A \rangle_{a,\lambda,s^{\mathfrak{a}}}, \langle B \rangle_{b,\lambda,s^{\mathfrak{b}}}$ in the RHS of (3.1) depend only on one selection input a, respectively b. This has the consequence that expression (3.2) has only four different quantities instead of eight and fulfills the Bell-CHSH inequality (7). It is possible, to interpret these quantities as the expectations of four random variables.

(2) A similar proof can be given for a slightly different Bell test simulation protocol, where after the initial state preparation a sequence of input symbols generates a sequence of output symbols. The state after step n is the initial state for step $n+1$ with an initial state probability distribution $p_{n,\ldots}^{\mathfrak{ab}}$ [14] analog to $p_0^{\mathfrak{ab}}$. Because the proof of proposition 3 requires not a product distribution of the initial states, it is valid for each step. In total, the output expectation for all sequences is the mean value of the output expectations after each step.

(3) The proof is also valid for deterministic sequential machines (cf. App. A). However, in this case a simplified proof is possible: Because of the independence of the remote input, the output of each machine is implicitly defined by two real functions $A_0(\lambda, s^{\mathfrak{a}}) = A(0, \lambda, s^{\mathfrak{a}})$, $A_1(\lambda, s^{\mathfrak{a}}) = A(1, \lambda, s^{\mathfrak{a}})$ and $B_0(\lambda, s^{\mathfrak{b}}) = B(0, \lambda, s^{\mathfrak{b}})$, $B_1(\lambda, s^{\mathfrak{b}}) = B(1, \lambda, s^{\mathfrak{b}})$, so proposition 7 applies. This is also valid for infinite deterministic machines, e.g., analog computer or neural networks over rational or

[14] This probability distribution depends on the complete history of inputs and the initial state.

real numbers. Infinite probabilistic machines require additional measure theoretic assumptions.

(4) It is important that the Bell test simulation follows the protocols described in Section 3.1 or in Note 2, where the inputs a and b are *freely* chosen by the operators or *independent* random generators and do not depend on λ (or the initial machine states). To shed some light on the *superdeterminism loophole* (cf. Hossenfelder and Palmer, 2020) of Bell's theorem, which doubts the free will of experimenters and the independence of random generators, we can consider the case where the input λ contains the recorded output A,B of an ideal Bell test experiment together with the recorded choices a,b. The actual experimenters enter these recorded choices from λ as their own input (violating the protocol). Very simple machine functions $p^{\mathfrak{a}}, p^{\mathfrak{b}}$, which copy the recorded outputs A,B from λ, will exactly reproduce the recordings from the ideal Bell test experiment and violate the Bell-CHSH inequality, without any functional dependence on the inputs a,b. In a complete deterministic world, the recorded inputs a,b can be omitted from λ if the superdeterministic fate determines the right input corresponding to the recorded values of A,B for each run of the simulation. However, an observer not believing in such fate but knowing Bell's theorem for sequential machines can only conject a violation of the protocol or a hidden dependence of the machine functions on the remote input.

4. Conclusion

Bell's theorem for sequential machines sets limits to any computer simulation of quantum physical experiments. It shows that some data transmission between the separated parts is necessary to violate the Bell-CHSH inequality in the Bell test simulation and reproduce the statistical results of an ideal quantum physical Bell test experiment.

This sheds some light on Bell's original theorem if we add a suitable ontological hypothesis, for example: For a *realistic* physical theory, every observation can be simulated by sequential machines *and* observations on *locally* separated systems can be simulated by separated machines. Consequently, in order to explain the results of the quantum physical Bell test experiments, information transmission between the separated machines must be assumed[15]. However, no such transmission was observed in any of these experiments. Moreover, if there were hidden transmissions, they would have to be much faster than the speed of light according to the experimental results of Gisin et al., 2008, which is beyond any known physical mechanism. This is an obstacle for theories that attempt to explain fundamental physics in terms of (non-quantum) *cellular automata*, essentially lattices of sequential machines with input/output only between neighboring machines (see 't Hooft, 2016).

Perhaps Bell's theorem for sequential machines explains the famous statement that "nobody understands quantum mechanics" [Feynman, 1965]. There are different ideas about what understanding means. However, an important way to understand a process, especially in science, is to follow it step by step in an algorithmic simulation, often only in the imagination through a chain of thought. The theorem makes this impossible for the ideal Bell test experiment if the separation of parts is observed.

Finally, the theorem has value for automata theory itself. It sets some limits for networks of sequential machines that can be crossed by quantum devices, e.g., allowing cryptographic protocols as described by Ekert [1991].

[15] if superdeterminism is excluded.

References

A. Aspect, J. Dalibard, and G. Roger. Experimental test of Bell's inequalities using time-varying analyzers. *Phys. Rev. Lett.*, 49, 1982. doi: 10.1103/PhysRevLett.49.91.

J. S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1, 1964. doi: 10.1017/CBO9780511815676.004.

J. S. Bell. Bertlmann's socks and the nature of reality. *Journal de Physique*, 42, 1981. doi: 10.1017/CBO9780511815676.018.

B. S. Cirel'son. Quantum Generalizations of Bell's Inequality. *Lett. Math. Phys.*, 4, 1980. doi: 10.1007/BF00417500.

J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23, 1969. doi: 10.1103/PhysRevLett.23.880.

A. K. Ekert. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.*, 67, 1991. doi: 10.1103/PhysRevLett.67.661.

R. P. Feynman. *The Character of Physical Law*. Random House, New York, 1965.

N. Gisin, D. Salart, A. Baas, C. Branciard, and H. Zbinden. Testing spooky action at a distance. *Nature*, 454, 2008. doi: 10.1038/nature07121.

F. C. Hennie. One-Tape, Off-Line Turing Machine Computations. *Information and Control*, 8, 1965. doi: 10.1016/S0019-9958(65)90399-2.

B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526, 2015. doi: 10.1038/nature15759.

S. Hossenfelder and T. Palmer. Rethinking Superdeterminism. *Front. Phys.*, 8, 2020. doi: 10.3389/fphy.2020.00139.

L. J. Landau. On the Violation of Bell's Inequality in Quantum Theory. *Physics Letters A*, 120, 1987. doi: 10.1016/0375-9601(87)90075-2.

W. Myrvold, M. Genovese, and A. Shimony. Bell's Theorem. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2024 edition, 2024. URL `https://plato.stanford.edu/entries/bell-theorem/`.

S. Popescu and D. Rohrlich. Nonlocality as an axiom. *Found. Phys.*, 24, 1994. doi: 10.1007/BF02058098.

A. Salomaa. *Theory of Automata*. Pergamon Press, Oxford, 1969.

A. Shimony. Bell's theorem. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2016. URL `https://plato.stanford.edu/archives/win2016/entries/bell-theorem/`.

S. Storz, J. Schär, A. Kulikov, and et al. Loophole-free Bell inequality violation with superconducting circuits. *Nature*, 617, 2023. doi: 10.1038/s41586-023-05885-0.

G. 't Hooft. *The Cellular Automaton Interpretation of Quantum Mechanics*. Springer Cham, Berlin, 2016. doi: 10.1007/978-3-319-41285-6.

Appendix A. Sequential machines[16]

A *sequential machine* (SM) is a simple model of a computing device. It accepts an input and then produces an output, possibly changing its internal state. I, O, S denote the sets of *input symbols*, *output symbols* (also called input and output alphabet) and *states*. An

[16] This Section is based on Salomaa [1969]

SM is called *finite* if all sets I, O, S are finite. Finite SMs are well suited for the formal description of control devices and many physical experiments.

A *deterministic SM* (DSM) is defined by a quintuple (I, O, S, s_0, f) where $s_0 \in S$ is the initial state and the *deterministic machine function*

$$f\colon I \times S \to O \times S, (i,s) \mapsto (o,t) = f(i,s)$$

determines the resulting output o and successor state t after the input i in state s.

For a *probabilistic* SM, the resulting output symbol o and state t are randomly chosen after each input. A *finite probabilistic SM* (FPSM) is defined by a quintuple (I, O, S, p_0, p) with finite I, O, S, an *initial state probability distribution function*

$$p_0\colon S \to [0,1]$$

which gives the probability of the initial state s, with

$$\sum_{s \in S} p_0(s) = 1$$

and a *probabilistic machine function*

$$p\colon O \times S \times I \times S \to [0,1], (o,t,i,s) \mapsto p(o,t \mid i,s)$$

which gives the probability of obtaining output o and state t after input i in state s, with

$$\sum_{o \in O} \sum_{t \in S} p(o,t \mid i,s) = 1$$

for all $i \in I, s \in S$.

An FPSM (I, O, S, p_0, p) is *deterministic* if the image of the functions p_0 and p is $\{0,1\}$. In this case, a corresponding DSM (I, O, S, s_0, f) can be defined with the initial state $s_0 \in S$, uniquely determined by $p_0(s_0) = 1$, and the (total) function f, which is uniquely defined by the set of pairs $((i,s) \mapsto (o,t))$ with $p(o,t \mid i,s) = 1$. The original FPSM is *completely equivalent* to this DSM in the sense that it gives the same output o and successor state t for each input $i \in I$ in state $s \in S$ with probability 1 and different results with probability 0. Conversely, for any finite DSM (I, O, S, s_0, f), there is a completely equivalent deterministic FPSM $(I, O, S, \delta_{s_0,s}, \delta_{f(i,s),(o,t)})$, where $\delta_{x,y}$ denotes the Kronecker symbol with the values 1 if $x = y$ and 0 otherwise.

Generally, two FPSMs $M^{\mathfrak{a}} = (I, O, S^{\mathfrak{a}}, p_0^{\mathfrak{a}}, p^{\mathfrak{a}})$, $M^{\mathfrak{b}} = (I, O, S^{\mathfrak{b}}, p_0^{\mathfrak{b}}, p^{\mathfrak{b}})$, with the same input and output symbol sets, are considered *equivalent* if for each finite sequence of input symbols, the probability of each sequence of output symbols with the same length is equal. Salomaa [1969] proves that equivalence is determined by the set of input/output sequences with the finite length $l = |S^{\mathfrak{a}}| + |S^{\mathfrak{b}}| - 1$.

Many other machine types can be described as FPSMs. *Moore* and *Mealy machines* are FPSMs with special forms of the probabilistic machine function p (cf. Salomaa, 1969). A "stateless" *circuit* can be described as an FPSM with one constant state, i.e., $S = \{s_\emptyset\}$. *Finite automata*, without explicit output, can be represented by FPSMs with $O = \{0,1\}$, where output 1 indicates the acceptance of the input sequence. Any *Turing machine* that terminates in fewer steps than a fixed finite limit (i.e., any computer that gives an output after an input with a maximum delay), can be represented by an equivalent FPSM [Hennie, 1965]. Even a *quantum computer* with classical input and output I, O (and a finite set of initial states and gates) can be described as an FPSM with a probabilistic machine function, whose probabilities are determined by the laws of quantum mechanics (similar to the machine M_5 in Example 1).

APPENDIX B. BELL-CHSH INEQUALITY

Proposition 7. *Any four real numbers $A_0, A_1, B_0, B_1 \in [-1,1]$ fulfill the inequality*

$$\left|A_0B_0 + A_0B_1 + A_1B_0 - A_1B_1\right| \leq 2. \tag{B.1}$$

Proof. The expression

$$A_0B_0 + A_0B_1 + A_1B_0 - A_1B_1$$

is linear in each of the four variables. Therefore, its maximum and minimum are located on a corner of the hypercube $[-1,1]^4$, where $A_0, A_1, B_0, B_1 \in \{-1,1\}$. In this case, for all 16 possible valuations, the following equations are valid

$$A_0B_0 + A_0B_1 + A_1B_0 - A_1B_1 = A_0\left(B_0 + B_1\right) + A_1\left(B_0 - B_1\right) = \pm 2.$$

□

Proposition 8. *Four random variables $A_0, A_1, B_0, B_1 \colon \Omega \to [-1,1]$ on a measure space $(\Omega, \mathscr{A})$ fulfill for any probability measure μ on $(\Omega, \mathscr{A})$ the Bell-CHSH inequality*

$$\left|\left\langle A_0B_0 + A_0B_1 + A_1B_0 - A_1B_1 \right\rangle\right| \leq 2$$

where $\langle\ \rangle$ indicates the expectation value with the measure μ.

Proof. Let for any $\omega \in \Omega$

$$E(\omega) = A_0(\omega)B_0(\omega) + A_0(\omega)B_1(\omega) + A_1(\omega)B_0(\omega) - A_1(\omega)B_1(\omega).$$

Then, because of (B.1), $|E(\omega)| \leq 2$ for all $\omega \in \Omega$ and

$$\left|\left\langle E\right\rangle\right| = \left|\int_\Omega E(\omega)d\mu\right| \leq \int_\Omega \left|E(\omega)\right| d\mu \leq \int_\Omega 2d\mu = 2.$$

□