Quantum Kolmogorov complexity and quantum correlations in deterministic-control quantum Turing machines

Mariano Lemus[1,2], Ricardo Faleiro[1,2], Paulo Mateus[1,2], Nikola Paunković[1,2], and André Souto[2,3,4]

[1]Departamento de Matemática, Instituto Superior Técnico, Universidade de Lisboa, Av.Rovisco Pais, 1049-001 Lisboa, Portugal
[2]Instituto de Telecomunicações, Av.Rovisco Pais, 1049-001 Lisboa, Portugal
[3]Lasige, Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749-016 Lisboa, Portugal
[4]Departamento de Informática,Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749-016 Lisboa, Portugal

This work presents a study of Kolmogorov complexity for general quantum states from the perspective of deterministic-control quantum Turing Machines (dcq-TM). We extend the dcq-TM model to incorporate mixed state inputs and outputs, and define dcq-computable states as those that can be approximated by a dcq-TM. Moreover, we introduce (conditional) Kolmogorov complexity of quantum states and use it to study three particular aspects of the algorithmic information contained in a quantum state: a comparison of the information in a quantum state with that of its classical representation as an array of real numbers, an exploration of the limits of quantum state copying in the context of algorithmic complexity, and study of the complexity of correlations in quantum systems, resulting in a correlation-aware definition for algorithmic mutual information that satisfies symmetry of information property.

1 Introduction

Kolmogorov complexity is a mathematical formulation to capture the intuitive idea of the amount of information an individual object (usually a string) has. This notion was independently proposed by Kolmogorov in 1965 [12], Solomonoff in 1964 [30] and Chaitin in 1966 [7]. It defines the amount of information of an object as the number of minimal instructions to be given to a machine to reproduce that object. If the object is a string x, and the machine is a Universal Turing machine U, the Kolmogorov complexity of x is the length of the shortest program that running in U as input, outputs x. We refer to the book by Li and Vitányi [14] for a comprehensive reading of this theme.

Several aspects must be considered to extend the notion of complexity to the quantum realm, as, in this context, the objects are quantum states, and the set of potential operations is not discrete. In [35], Vitányi identifies three approaches for the quantitative definition of information in a quantum state. The first one is the description (using classical bits) of the physical apparatus that outputs the given state. In this context, the continuous set of quantum states can be approximated by Cauchy sequences of states that can be directly outputted by the machine. The second approach consist on the *qubit* description of the quantum machine outputting the state, which upper bounds the information in a given quantum state by the logarithm of the dimension of the associated Hilbert space. The third approach considers

Mariano Lemus: mlemus@tecnico.ulisboa.pt, Corresponding author
Ricardo Faleiro: ricardofaleiro@tecnico.ulisboa.pt
Paulo Mateus: paulo.mateus@tecnico.ulisboa.pt
Nikola Paunković: npaunkov@mathtecnico.ulisboa.pt
André Souto: ansouto@ciencias.ulisboa.pt

the information content in a quantum state to be the classical information content of the list of real numbers involved in a (fixed) mathematical expression for the state. Furthermore, there are additional aspects to consider, including how to handle the distinction between a machine outputting a state or a high fidelity approximation of it; and the issue of directly addressing mixed states, and therefore define the machine through density operators instead of pure states.

In the last few decades, several definitions and approaches have been proposed in the literature that explore all the aforementioned possibilities. Svozil [32], in his pioneer work, defined the quantum Kolmogorov complexity of a state as the length of the minimum classical description in a quantum Turing machine (in the machine model defined in [8]) of that state. In [34], Vitányi proposed the definition of the (quantum) Kolmogorov complexity of a pure quantum state $|\varphi\rangle$ as the minimum value of the sum of two parts: (i) the length of a classical program computing $|\psi\rangle$, an approximation of $|\varphi\rangle$, and (ii) the negative log-fidelity of the approximation of $|\psi\rangle$ to $|\varphi\rangle$.

These two first approaches considered classical descriptions for quantum states. In [6], the authors considered the possibility of using a quantum program (encoded in qubits instead of bits) to define the complexity similarly to Vitányi's. In their work, Berthiaume, van Dam, and Laplante defined the Kolmogorov complexity of a qubit string as the length of the shortest quantum bit string that, given to a quantum universal Turing machine, produces the qubit string with high fidelity. Its properties were further studied in [23, 24].

Based on a generalization of the notion of universal semi-measure, Gács [9] proposed to define Kolmogorov complexity based on density matrices. Later, in [21], the Kolmogorov complexity of a pure quantum state $|\varphi\rangle$ was defined as the length of the shortest description of a quantum circuit capable of producing such state (or an approximation of it). More recently, a new approach to define Kolmogorov complexity based on a particular type of quantum Turing machines, called deterministic controlled Turing machines was proposed [16]. In this approach, a description of a quantum state $|\varphi\rangle$ is given by a sequence of quantum gates, which transforms an initial reference state of the quantum tape into the state $|\varphi\rangle$.

Although (classical) Kolmogorov complexity has been successfully applied to several fields ranging from security, computational complexity, bio-medicine, etc [1, 2, 31], corresponding applications of its quantum versions are yet just a few. For example, in [21] the authors establish the ground to characterize quantum entanglement via Kolmogorov complexity. In [22] Mora and Briegel and Kraus studied the application of quantum Kolmogorov complexity to thermodynamics and complexity theory. In [3] and [17] the authors explore the connection between (quantum) Kolmogorov complexity and entropy applied to Brudno's theorem and information-disturbance theorem, respectively. Another example of application is in cryptography, in [18], the authors use (classical) Kolmogorov complexity to derive the security of BB84 quantum key distribution [4]. Recently, in [28], algorithmic information theory has been used in genomics applications.

The interconnection between Kolmogorov Complexity Theory and Information Theory is well-known and extensively studied in the literature [10, 33]. Concepts in one theory can find analogous versions in the other. One of the most important concepts in Information Theory is the notion of mutual information, which captures the inherent dependency of how much information is needed to explain one object having another object as a starting point. One of the most significant characteristics of (classical) mutual information is that it is symmetric, that is, $I(X, Y) = I(Y, X)$. On the other hand, for algorithmic mutual information such equality holds up to a logarithmic term of the size of the objects [13, 33, 36]. The concept is briefly mentioned in the quantum Kolmogorov case in [34]. In the context of quantum Kolmogorov complexity theory, the study of quantum algorithmic mutual information can help understand and quantify the complexity of the correlations between parts of multipartite quantum systems, since these systems can be inherently correlated. It is well-known that the correlations in quantum systems can be stronger than those present in classical systems due to the presence of entanglement [11] or the more general quantum discord [20]. Understanding quantum correlations and their relation to algorithmic mutual information measured in terms of a quantum version of Kolmogorov complexity is important, as it opens the possibility of quantifying the amount of manipulation required to create such correlations between two systems. Consequently, this connection can be useful in analyzing complexity in quantum computing, leading to more efficient algorithms, or in relating security and complexity in quantum cryptography protocols, for example.

1.1 Overview of results

The main results of this work can be summarized as follows:

- We present a natural generalization of the *deterministic-control quantum Turing machine* (dcq-TM) model, originally introduced in [16], to allow for mixed state inputs and outputs.
- We define the set of dcq-computable states as the set of states which can be approximated with arbitrary precision using a dcq-TM.
- We define the (conditional) Kolmogorov complexity K of quantum states in the dcq-TM model and show that it is machine independent up to a constant.
- Using the concept of *classical representation* of a quantum state to refer the array of real numbers describing its density matrix in the computational basis, we show that the defined set of dcq-computable states coincides with the set of states with computable classical representations. Moreover, we note that the algorithmic complexity of a quantum state and that of its classical representation are equal up to a constant.
- To contrast with the above result, we compare the properties of quantum states and their classical representations when used as a resource. In this context, we show that the representation contains more descriptive information when used as a resource for computation. We then further explore this difference by reducing the problem of computing an n-qubit state ρ to the problem of computing a *duplicate* of ρ given access to one copy of it, resulting in a relation between the information in a single copy of a quantum state versus the information in two copies of itself. We conclude that for the majority of quantum states, cloning them is essentially as hard as to create them. The result can be interpreted as an algorithmic information version of the no-cloning theorem.
- Finally, we explore the concept of algorithmic mutual information for dcq-computable states. We show that the analogous version of the classical chain rule is not satisfied for general bipartite states and interpret this discrepancy to be caused by the presence of correlations among the two subsystems. We propose an alternative generalization of the chain rule for quantum systems and show that it is indeed satisfied under our definition of K. Using this result, we present a correlation-accounting definition for algorithmic mutual information and show that it satisfies the symmetry of information property.

This paper is organized as follows: In Section 2, we briefly go through some important definitions on classical Kolmogorov complexity and computability of real numbers. In Section 3, we describe the dcq-TM machine model and present definitions for classical representations of quantum states, as well as classical simulators for quantum machines. In Section 4, we define the concept of computable quantum states and compare it to the set of states with computable classical representations. In Section 5, we define the concept of Kolmogorov complexity in the dcq-TM model and study some comparisons with the complexity of the associated classical representations, culminating in a dcq-TM version of no-cloning theorem. Later, in Section 6 we study potential definitions of algorithmic mutual information with the intent of quantifying quantum correlations, resulting in a generalization of the chain rule and an alternative definition for mutual information. Finally, in Section 7, we summarize and discuss the results, as well as point to further research directions on the topic.

2 Preliminaries

2.1 Kolmogorov complexity

In this subsection we briefly review some basic properties of the standard classical algorithmic complexity K in order to better motivate the definitions in the following sections. Classically, the Kolmogorov complexity of a string x is defined as the length of the shortest program that produces x when given to a classical universal Turing machine.[1] We refer the reader to the book of Li and Vitányi [14] for a complete study on this topic.

[1] In this work we focus on inputs to the Turing Machines which need not be prefix-free. In the literature, this is sometimes called "plain" Kolmogorov complexity.

Definition 2.1. *(Kolmogorov complexity)*
Let $x, y \in \{0,1\}^*$ be two strings and U a classical universal Turing machine. The Kolmogorov complexity of x given y is defined as:

$$\mathrm{K}(x \mid y) = \inf_p \{|p| : \mathsf{U}(p, y) = x\}. \tag{1}$$

The default value for y, the auxiliary input for the program p, is the empty string. In order to avoid overloaded notation, in those cases we typically drop this argument in the notation. Notice that Kolmogorov complexity is machine independent in the sense that we can fix a universal Turing machine U whose program size is at most a constant additive term worst than any other machine, and the running time is, at most, a logarithmic multiplicative factor slower than in any other machine (see Theorem 7.1.1 from [14]).

In information theory, one of the fundamental quantities is mutual information. There is an analogous version for Kolmogorov complexity that we will call *algorithmic mutual information*, and is given by

$$\mathrm{I}_{\mathrm{K}}^{(1)}(x, y) = \mathrm{K}(x) - \mathrm{K}(x \mid y). \tag{2}$$

In contrast with standard (Shannon) mutual information, this quantity is independent of any probability distribution and is concerned only with how much of the information in the string y can be used to describe x. In information theory, one useful result is the chain rule, which allows us to express the entropy of a joint distribution in terms of its conditional and marginal distributions. For Kolmogorov complexity we have,

Property 2.1. *(Chain rule 1)*
For all strings x and y in $\{0,1\}^*$,

$$\mathrm{K}(x, y) = \mathrm{K}(x) + \mathrm{K}(y \mid x) + O(\log(\mathrm{K}(x, y))). \tag{3}$$

Using Equation (3) in combination with Equation (2) leads to a result called *symmetry of information*, which states that

$$\mathrm{I}_{\mathrm{K}}^{(1)}(x, y) = \mathrm{I}_{\mathrm{K}}^{(1)}(y, x) + O(\log(\mathrm{K}(x, y))). \tag{4}$$

The chain rule also allows us to relate $I_{\mathrm{K}}^{(1)}$ with another quantity, which can also be understood as a measure of mutual descriptive information

$$\mathrm{I}_{\mathrm{K}}^{(2)}(x, y) = \mathrm{K}(x) + \mathrm{K}(y) - \mathrm{K}(x, y). \tag{5}$$

Analogous expressions for $\mathrm{I}_{\mathrm{K}}^{(1)}$ and $\mathrm{I}_{\mathrm{K}}^{(2)}$ exist in classical information theory for random variables X, Y by replacing the Kolmogorov complexity K by the Shannon entropy function H. In classical information theory those quantities are proven to be equivalent [29]. In the case of algorithmic information theory, they are related by

$$\mathrm{I}_{\mathrm{K}}^{(1)}(x, y) = \mathrm{I}_{\mathrm{K}}^{(2)}(x, y) + O(\log(\mathrm{K}(x, y))). \tag{6}$$

Let $n = |x| + |y|$, by noting that $\mathrm{K}(x, y) \leq \mathrm{K}(xy) + O(\log(n))$ we can slightly modify the chain rule to a form that will be later more suitable for generalizing to the quantum case [14]:

Property 2.2. *(Chain rule 2)*
For all strings x and y in $\{0,1\}^*$ with $n = |x| + |y|$,

$$\mathrm{K}(xy) = \mathrm{K}(x) + \mathrm{K}(y \mid x) + O(\log(n\,\mathrm{K}(xy))) \tag{7}$$

$$= \mathrm{K}(x) + \mathrm{K}(y \mid x) + O(\log(n)). \tag{8}$$

Even though Eq. (8) is the simplest form of the classical chain rule, we want to highlight Eq. (7) because it will allow us to better compare it with its quantum counterpart later on.

2.2 Computability of real numbers

In this section, we briefly review the notions associated with the computability and algorithmic complexity of real numbers. Throughout the paper, we shall need to encode tuples of strings into a single string. Therefore, we denote by $[x_1, \ldots, x_m]$ the encoding of the m-tuple $(x_1, \ldots, x_m)$ of binary strings into a single binary string. Fix a one-to-one encoding of natural numbers in strings and encoding of rational numbers by assigning the information of sign, numerator, and denominator to the different components of a tuple.

Definition 2.2. *A real number r is said to be computable if there exists a (classical) Turing machine T which on any input $k \in \mathbb{N}$ outputs $q_k \in \mathbb{Q}$, such that*

$$|r - q_k| \leq \frac{1}{k}. \tag{9}$$

In this case, we say that T computes r.

Computable real numbers can be encoded effectively in natural numbers. Given an enumeration of the set of Turing machines, we will consider that a computable real number is encoded by any index that represents a Turing machine that will output the approximation of it according to Def. 2.2. A tuple of real numbers $(r_1, \ldots, r_n) \in \mathbb{R}^n$ is computable if all its components r_i are computable. This encompasses complex numbers (ordered pairs of real numbers) and, notably, density matrices (arrays of complex numbers). We can proceed now to define the Kolmogorov complexity of a real number by identifying it with the length of a program that produces a sequence that converges to it according to Equation (9):

Definition 2.3. *Let r be a real number and U a universal classical Turing machine. The Kolmogorov complexity of r is given by*

$$\mathrm{K}(r) = \inf_p \{|p| : \mathsf{U}_p \text{ computes } r)\}, \tag{10}$$

where $\mathsf{U}_p(x) = \mathsf{U}(p, x)$.

The notion Kolmogorov complexity generalizes straightforwardly to arrays of real numbers and matrices as expected.

Remark. *Since the set of computable numbers contains $\mathbb{N}$, some ambiguity may arise when talking about the complexity of a natural number, as both Def. 2.1 and Def. 2.3 may apply (depending on the chosen encoding), but they in general do not coincide. In this document, unless stated otherwise, whenever a number x is understood to take values only in $\mathbb{N}$ (index in an enumeration, counter in a* for *loop, etc.) we assume that $\mathrm{K}(x)$ is given by Def. 2.1. Otherwise, we assume its complexity is given by Def. 2.3.*

3 Definition of the machine

3.1 Deterministic control quantum Turing machine (dcq-TM)

A dcq-TM is a completely deterministic machine in the sense that the flow of computation is entirely classically controlled, but it is also capable of having quantum states as inputs and outputs. The following description of a dcq-TM is based on the one defined in [16] with the added feature that the machine is also able to handle general mixed states, and not just pure states. The machine is understood as having at least two infinite tapes, one classical and one quantum, with their respective classical and a quantum heads[2]. The distinctive feature of the machine is that it has a deterministic control over the computation happening on the quantum tape – this control is only dependent on the set of internal states (Q) of the dcq-TM, which are classical, and the contents of the classical tape. Thus, the set of internal states and the classical input specify the dcq-TM's computational dynamics entirely, and the quantum tape is a

[2]Please note that our "quantum head" is in fact a classical head that operates over the quantum tape. In this sense, it evolves according to a completely classical-deterministic transition function, as opposed to the Deutsch and Bernstein-Vazirani models [5, 8] where the quantum head evolves unitarilly and can be found in superpositions of its classical internal states.

"work only" tape whose contents do not influence the computation. The quantum tape is effectively an infinite quantum memory, composed by quantum cells, each associated with a two dimensional (qubit) Hilbert space $\mathbb{C}^2$. The classical control of the dcq-TM commands the quantum head in a deterministic way. It chooses where the quantum head moves to and picks, from a pre-chosen universal set $\mathbb{U}$ (which without loss of generality is assumed to be finite), which unitary gates to apply and to which cells of the quantum tape to apply them to. It is important to stress that the unitary transformations in $\mathbb{U}$ must be described only with computable complex numbers, otherwise the machine would compute super-Turing distribution/functions. The machine is rigorously defined as follows,

Definition 3.1. *A deterministic-control Quantum Turing machine or dcq-TM is defined by the pair, (Q, δ), where Q and δ are, respectively, the finite set of control states containing at least the two distinct states q_s (starting state) and q_h (halting state) and the transition function of the computation. The transition function is,*

$$\delta : Q \times \mathbb{A} \to \mathbb{U} \times \mathbb{D} \times \mathbb{A} \times \mathbb{D} \times Q, \tag{11}$$

where $\mathbb{A} = \{0, 1, \square\}$ is the alphabet of the classical tape, $\mathbb{U}$ is a universal set of unitary operators that can be applied to the quantum tape, and $\mathbb{D} = \{\mathsf{L}, \mathsf{N}, \mathsf{R}\}$ is set of possible head displacements (left, none, right).

3.1.1 Anchor cell and input/output schemes

In order to have the quantum and classical inputs/outputs properly defined in the dcq-TM we introduce the concept of an *anchor cell*. The anchor cell serves the purpose of specifying a global reference point for each tape, in order to identify unambiguously the inputs and outputs of the machine. For a given tape, we specify the anchor cell (on that tape) to be the first cell before the input starts i.e if we index the input cells with positive numbers $i > 0$, the anchor cell will always be the cell with index $i = 0$. The input and the output schemes of the dcq-TM presented here are slightly different from the ones in the definition of [16], where there is no *anchor cell* defined, and the machine is not aware of the size of its quantum input/output. In the spirit of generalizing the model for general mixed states and more naturally defining inputs and outputs that may start or end with $|0\rangle$ states. We introduce this alternative definition from which it is straightforward to recover all the results already established regarding the dcq-TM.

Inputs – We say the machine received as input the pair $(x; \rho_{\langle 1,n\rangle})$ if its internal state is q_s, its classical and quantum heads are parked in their respective anchor cells, and the tapes' contents are as shown in Figure 1. The classical input x is the string written between the anchor and the first blank after the anchor. At the end of the classical input, we insert a blank cell to distinguish strings x and n, where the latter specifies the quantum input in the quantum tape. The quantum input $\rho_{\langle 1,n\rangle}$ is the physical quantum state present in the set of cells between the anchor and the cell specified by string n.

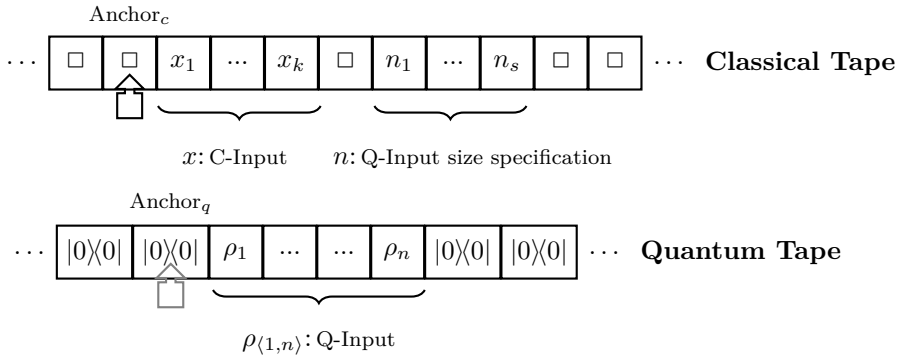

Figure 1: **Starting configuration of the dcq-Turing machine with a classical and quantum input.** $x = x_1 \cdot x_2 \cdots x_k$ is the classical input, while $\rho_{\langle 1,n\rangle}$ is the quantum input "written" on n cells of the quantum tape, where ρ_i is the local state of the $i-$th cell, represented by $\boldsymbol{\rho}_i = \mathrm{Tr}_{(1,\ldots,i-1,i+1,\ldots,n)}[\boldsymbol{\rho}_{\langle 1,n\rangle}]$. The string $x\square n$ is surrounded by blank symbols extending to infinity in both directions. The quantum input is surrounded by cells in the local zero state $|0\rangle\langle 0|$ extending to infinity in both directions.

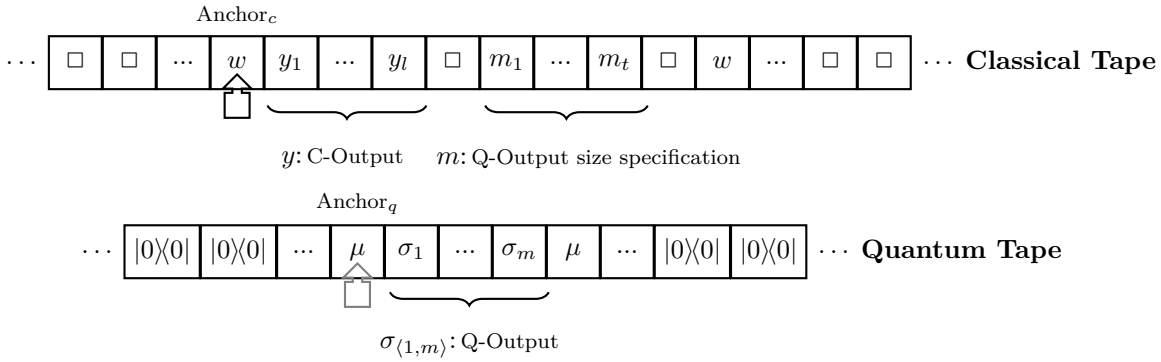

Figure 2: **Final state of the dcq-Turing machine with a classical and quantum outputs.** $y = y_1 \cdot y_2 \cdots y_r$ is the classical output, while $\sigma_{\langle 1,s\rangle}$ is the quantum output, where σ_i is the local state of the $i-$th cell, represented by $\boldsymbol{\sigma}_i = \mathrm{Tr}_{(1,\ldots,i-1,i+1,\ldots,n)}[\boldsymbol{\sigma}_{\langle 1,s\rangle}]$. The size (in number of qubits) of $\sigma_{\langle 1,s\rangle}$ is specified by the string s in the classical tape. The working cells, which are used during the computation but are not part of the output in the classical and quantum tape are denoted by w and μ, respectively.

Outputs – We say the machine produced as output, $(y;\sigma_{\langle 1,s\rangle})$, if the machine is in the halting state $\mathsf{q_h}$ and the classical and quantum tapes are in the configuration shown in Figure 2. The classical output y is defined as the string written in the set of cells that are between the anchor and first blank to the right of the anchor. The quantum output specification m is defined as the string between the first and the second blanks appearing after the anchor. Finally, the quantum output is defined as the physical quantum state present in the set of cells between the anchor and the cell specified by string m.

Although there are many ways to evaluate functions that have multiple classical inputs/outputs using the above configuration, for the purposes of the examples of this paper it is convenient to have a unified encoding to use whenever we refer to algorithms that use a single string x to encode a tuple $(x_1,\ldots,x_k)$ of strings. We use the following encoding

$$(x_1,\ldots,x_k) \equiv x_1 \cdot \ldots \cdot x_k \equiv \ell(x_1)0x_1\ell(x_2)0x_2\cdots\ell(x_{k-1})0x_{k-1}x_k, \tag{12}$$

where $\ell(x_i) = 1^{|x_i|}$ is a string of 1s of length $|x_i|$. Using this encoding, each input x_i adds $2|x_i|+1$ bits to the input string, except the last one, which only adds $|x_i|$; that is

$$|(x_1,\ldots,x_k)| = \sum_{i=1}^{k-1} 2|x_i| + |x_k| + (k-1). \tag{13}$$

Multiple quantum inputs/outputs can be defined by encoding the tuple of sizes (number of qubits) of the different systems in the same way described above.

3.1.2 Symbolic notation for dcq-TM computations

Thus, an input on the dcq-TM of the form $(x;\rho_{\langle 1,n\rangle})$, specifies the following representation of the tapes,

$$\begin{aligned} [\text{Classical tape}] &= [\square]_{(-\infty,0)}\ [x]\ [\square]\ [n]\ [\square]_{(|x|+|n|+1,\infty)} \\ \rho_{\text{ Quantum Tape}} &= |\mathbf{0}\rangle\langle\mathbf{0}|_{(-\infty,0)} \otimes \rho_{\langle 1,n\rangle} \otimes |\mathbf{0}\rangle\langle\mathbf{0}|_{(n+1,\infty)}, \end{aligned} \tag{14}$$

where $|\mathbf{0}\rangle\langle\mathbf{0}|_{(a,b)}$ denotes $|0\rangle\langle 0|_a \otimes |0\rangle\langle 0|_{a+1} \otimes \ldots \otimes |0\rangle\langle 0|_b$. If a specific dcq-TM machine, T, on input $(x;\rho_{\text{in}})$ produces output $(y;\rho_{\text{out}})$ we represent the computation symbolically as,

$$\mathsf{T}(x;\rho_{\text{in}}) = (y;\rho_{\text{out}}), \tag{15}$$

for the overall output, and

$$\mathsf{T}^{(C)}(x;\rho_{\text{in}}) = y \qquad \mathsf{T}^{(Q)}(x;\rho_{\text{in}}) = \rho_{\text{out}}, \tag{16}$$

for the classical and quantum outputs. To represent empty inputs/outputs we use the symbols □ for the classical tape, and ϵ for the quantum one. Note that ϵ is used to denote quantum input/outputs of size

zero. We call the reader's attention to the fact that ϵ is not a density operator acting on a Hilbert space, since a zero size input/output does not refer to any system in the quantum tape; we use it to refer to the cases where there may not be a quantum input/output.

The computability properties of dcq-TM may be dependent on the chosen set $\mathbb{U}$ of basic quantum operations, namely the states that can be prepared exactly by the machine. On this work, we will focus on dcq-TM based on the universal set $\mathbb{U} = \{\mathbb{1}, \mathsf{H}, \mathsf{S}, \frac{\pi}{8}, \mathsf{Sw}, \mathsf{c\text{-}Not}\}$ commonly used in the circuit model for quantum computation. This set is important because it is associated with the capabilities of practical quantum computers. It is also important to remark that the transition function of a dcq-TM depends only on the contents of the classical tape, this is the reason the model is denoted as *deterministic control*. In [16], the universality of this model was proven and, moreover, it was also shown that the dcq-TM model could simulate a special type of quantum circuit called a *seesaw circuit*. A *polynomial translatability theorem* was proven which states that any general quantum circuit can be simulated by one such seesaw circuit with a polynomially bounded overhead on the number of gates, showing that the dcq-TM can solve in a bounded polynomial-time the same problems that the regular quantum circuit model can, therefore, the BQP complexity class coincides with the analogous dcBQP class defined for the dcq-TM model. We also know that there is a universal dcq-TM, T_U, that can simulate every other dcq-TM on any arbitrary input. As such, without loss of generality,

$$\mathsf{T}_\mathsf{U}(x;\rho_{\text{in}}) = (\square;\rho_{\text{out}}) \Leftrightarrow \mathcal{C}_x^*(\rho_{\text{in}}) = \rho_{\text{out}}. \tag{17}$$

The previous expression represents symbolically the equivalence between the computation taken on a specific input in the dcq-TM model, and it is implementation of the *seesaw circuit* $\mathcal{C}_x^*$ that gives the same output. From now on, whenever we refer to a universal dcq-TM, we mean a universal dcq-TM machine fulfilling the s-m-n property.

Sometimes, we may want to refer to an instance of a universal dcq-TM U, simulating some other dcq-TM T. From the s-m-n property, we know that there is a string $t \in \{0,1\}^*$ such that, for any input $(x;\rho_{\text{in}})$ we have that $\mathsf{U}\big((t,x);\rho_{\text{in}}\big) = \mathsf{T}(x;\rho_{\text{in}})$. In these cases we say that the machine U *runs the program* t *on input* $(x;\rho_{in})$, denoted symbolically by:

$$\mathsf{U}_t(x;\rho_{\text{in}}) = \mathsf{U}\big((t,x);\rho_{\text{in}}\big) = \mathsf{T}(x;\rho_{\text{in}}). \tag{18}$$

3.2 Classical quantum state encoding and simulators

In this work we are interested in comparing the complexity aspects of two related, but fundamentally different types of objects. The first are the quantum states themselves, understood as density operators acting on a given Hilbert space, encoded in the physical states of the cells of the quantum tape, and are denoted by standard Greek letters ρ, σ, etc. The second are the *classical representations* of the quantum states, which are understood as arrays of complex numbers describing the respective density matrices in the computational basis, encoded by strings of symbols in the classical alphabet $\mathbb{A}$ according to an appropriate rule (as described in Section 2.2), and denoted by bolded Greek letters $\boldsymbol{\rho}, \boldsymbol{\sigma}$, etc. In the context of inputs and outputs of dcq-TMs, a statement of the form

$$\mathsf{T}(\boldsymbol{\rho}_{\text{in}};\epsilon) = (\boldsymbol{\rho}_{\text{out}};\epsilon), \tag{19}$$

is interpreted as follows: "Upon classical input of any string encoding the computational-basis matrix associated to the state ρ_{in} and no quantum input, the dcq-TM T outputs a string encoding the respective matrix for the state ρ_{out} without a quantum output". Finally, for simplicity, following the relation between the states and their matrix representations, for every function f of quantum states $\rho, \sigma, \dots$, we define $f(\boldsymbol{\rho}, \boldsymbol{\sigma}, \dots) \equiv f(\rho, \sigma, \dots)$.

Before moving on to discuss directly dcq-computable and (plain) dcq-computable quantum states, we establish a notation for the classical simulators of these machines. For any dcq-TM T, consider a "classical" machine $\tilde{\mathsf{T}}$ (a dcq-TM that does not interact with the quantum tape and whose quantum output is ϵ for any input), defined constructively from T, which appropriately approximates the matrix operations associated with the logic gates applied during the execution of T such that

$$\mathsf{T}(x;\rho_{\text{in}}) = (y;\rho_{\text{out}}) \Leftrightarrow \tilde{\mathsf{T}}\big((x,\boldsymbol{\rho}_{\text{in}});\epsilon\big) = \big((y,\boldsymbol{\rho}_{\text{out}});\epsilon\big), \tag{20}$$

whenever the respective string encodings $\boldsymbol{\rho}_{\text{in}}$ and $\boldsymbol{\rho}_{\text{out}}$ exist (that is, whenever the matrices $\boldsymbol{\rho}_{\text{in}}$ and $\boldsymbol{\rho}_{\text{out}}$ are computable).

4 Computability of quantum states

4.1 Directly computable states

Definition 4.1. *A quantum state ρ is directly dcq-computable whenever it can be outputted by a dcq-TM, provided the machine started the computation with no auxiliary input i.e.,*

$$\exists \mathsf{T} : \mathsf{T}(\square;\epsilon) = (\square;\rho). \tag{21}$$

The set of directly dcq-computable states is denoted by $\mathsf{DCOMP_q}$.

Denote by $\mathsf{T}(x;\sigma)\downarrow_t$ a particular (defined constructively from T and t) dcq-TM that simulates $\mathsf{T}(x;\sigma)$, except that if at the step t it has not halted, if forces it to halt, then outputs the output of the simulated machine (according to the rules specified in Section 3). $\mathsf{T}^{(C)}(x;\sigma)\downarrow_t$ and $\mathsf{T}^{(Q)}(x;\sigma)\downarrow_t$ are defined analogously. Consider now Algorithm 1 which, given a universal dcq-TM U, uses Cantor's zig-zag method to assign to each non-negative integer s a different directly dcq-computable state associated with running U for some combination of input and number of steps, which is different for every value of s. Because the output of a dcq-TM is well defined for any state of its tapes, each value of s is uniquely associated with a directly dcq-computable state through Algorithm 1. Additionally, because every directly dcq-computable state ρ is associated with U running *some* program for *some* number of steps, we know that there exists a value of s for which Algorithm 1 outputs ρ. This means that the set of states that can be returned by Algorithm 1 coincides set of directly dcq-computable states. For a fixed reference universal dcq-TM, the quantum output of Algorithm 1 defines a surjective function $\Pi_1(s)$ from the set of non-negative integers to $\mathsf{DCOMP_q}$, that is, an enumeration of it.

Parameters: Universal dcq-TM U ;
Input: Integer $s \geq 0$;
Set n equal to the smallest natural number such that $s < (n+1)(n+2)/2$;
if $n \leq 1$ **then**
  $m = s - n$;
else
  $m = s \mod (n(n+1)/2)$;
end
$\sigma = \mathsf{U}^{(Q)}(n-m;\epsilon)\downarrow_m$;
return $(\square;\sigma)$;

Algorithm 1: directly dcq-computable state generator algorithm.

4.2 Computable states

Definition 4.2. *A quantum state ρ is dcq-computable if there exists a dcq-TM T and an infinite sequence $\{\sigma_i\}_i$ of directly dcq-computable states such that:*

1. $\mathsf{T}(k;\epsilon) = (\square;\sigma_k)$*, i.e., upon classical input k and no quantum input, the machine T outputs the state σ_k.*
2. $D(\rho,\sigma_k) \leq 1/k$*, where $D(\cdot)$ denotes the trace distance of operators acting on the respective Hilbert space.*

We denote the set of all dcq-computable quantum states by $\mathsf{COMP_q}$.

Consider the set $\mathsf{COMP_c}$ of quantum states for which their density matrix representation, when written in the computational basis, has computable complex numbers as components. This set is relevant because it represents the set of quantum states that can be written in a classical computer. Let us state a pair of algorithms that will be useful in finding relationships between quantum states and their representation in the context of the dcq-TM model. Since our machine model is controlled by classical programs, we would expect that $\mathsf{COMP_q}$, the set of computable states as defined by Def. 4.2 and the set $\mathsf{COMP_c}$ coincide. This is indeed the case as shown below.

Theorem 4.1. $\mathsf{COMP_q} = \mathsf{COMP_c}$

Proof.
($\mathsf{COMP_q} \subseteq \mathsf{COMP_c}$):
Let $\rho \in \mathsf{COMP_q}$, and $\{\sigma_i\}_i$ a sequence which satisfies Def. 4.2. From the definition of $\mathsf{COMP_c}$, to show that $\rho \in \mathsf{COMP_c}$ we must show that $\boldsymbol{\rho}$ is computable. Note that the set $\mathbb{U}$ consists on matrices with computable components. It follows that the associated sequence of matrices $\{\boldsymbol{\sigma}_i\}_i$ is computable since, for any dcq-TM T such that $\mathsf{T}(k;\epsilon) = (\square;\sigma_k)$, the classical machine $\tilde{\mathsf{T}}$ simulates the evolution of the initial state though the quantum circuit specified by T and computes the components $\boldsymbol{\sigma}_n$. By recalling that the limit of any classically computable sequence is computable, it follows that $\lim_{n\to\infty} \boldsymbol{\sigma}_n = \boldsymbol{\rho}$ is computable and thus, $\rho \in \mathsf{COMP_c}$.

($\mathsf{COMP_c} \subseteq \mathsf{COMP_q}$):
Let $\boldsymbol{\rho}$ be a density matrix with computable components. From the definition of computability of real numbers, this means that there exists a set $\left\{f^k_{rs}\right\}_{r,s,k}$, of computable functions over $\mathbb{N}$ such that the sequence of matrices $\boldsymbol{\sigma}_k = (g_{rs}(k))$, where $g_{rs}(k) = f^0_{rs}(k) + i f^1_{rs}(k)$, satisfies

$$D(\boldsymbol{\sigma}_k, \boldsymbol{\rho}) \leq \frac{1}{k}. \tag{22}$$

Parameters: Universal dcq-TM U ;
Input: Integer k, $2^n \times 2^n$ density matrix $\boldsymbol{\sigma}$;
$s \leftarrow 0$;
while $\boldsymbol{\sigma}' = \tilde{\Pi}_1(s)$ *is not an* $2^n \times 2^n$ *matrix such that* $D(\boldsymbol{\sigma}', \boldsymbol{\sigma}) \leq 1/(2k)$ **do**
  $s \leftarrow s + 1$;
end
$\sigma' \leftarrow \Pi_1(s)$;
return $(s;\sigma')$

Algorithm 2: Given a density matrix $\boldsymbol{\sigma}$ and integer k, finds a directly dcq-computable state that is closer to σ than $1/(2k)$.

We want to use $\{\boldsymbol{\sigma}_k\}$ to construct a (computable) sequence $\{\sigma'_k\}$ of states associated with directly dcq-computable states that satisfy Def. 4.2. Consider the following construction for σ'_k: Fix a universal dcq-TM U and run Algorithm 2 with input $((k, \boldsymbol{\sigma}_{2k});\epsilon)$ and define σ'_k as its quantum output. We know that Algorithm 2 will always find such state because the set of gates that a dcq-TM has access is universal, and hence the set of directly dcq-computable states is dense in the set of quantum states. To prove that $\rho \in \mathsf{COMP_q}$ it only remains to show that the sequence $\{\sigma_i\}$ satisfies Def. 4.2(2). Indeed, we have that

$$D(\sigma'_k, \rho) \leq D(\sigma'_k, \sigma_{2k}) + D(\sigma_{2k}, \rho) \leq \frac{1}{2k} + \frac{1}{2k} = \frac{1}{k}. \tag{23}$$

□

5 Algorithmic complexity in the dcq-TM model

Having defined the features of the dcq-TM, we proceed to introduce two (non-equivalent) notions of the concept of Kolmogorov complexity. The first one is a generalization of the one proposed in [16] for general directly dcq-computable states:

Definition 5.1. *Let ρ be a directly dcq-computable quantum state and T be a dcq-TM. The basic algorithmic complexity of ρ is*

$$\mathrm{K}^d_{\mathsf{T}}(\rho) = \min_p \left\{ |p| : \mathsf{T}^{(Q)}(p;\epsilon) = \rho \right\}. \tag{24}$$

This definition is limited in that it is only applicable to directly dcq-computable states (denoted by the superscript d), which are dependent on the specific chosen universal set of gates $\mathbb{U}$ in Def. 3.1. We are interested in extending the concept of algorithmic complexity for dcq-computable states.

Definition 5.2. *Let ρ be a quantum state and T be a dcq-TM. The algorithmic complexity of ρ is given by*

$$\mathrm{K}_{\mathsf{T}}(\rho) = \inf\left\{|p| : \forall k \in \mathbb{N}\ D\left(\mathsf{T}^{(Q)}\left((p,k);\epsilon\right),\rho\right) \leq \frac{1}{k}\right\} \tag{25}$$

Definition 5.3. *Let x be a binary string, ρ, σ be two quantum states, and T be a dcq-TM. The conditional algorithmic complexity of ρ given the pair $(x;\sigma)$ is*

$$\mathrm{K}_{\mathsf{T}}(\rho \mid x,\sigma) = \inf\left\{|p| : \forall k \in \mathbb{N}\ D\Big(\mathsf{T}^{(Q)}\left((p,x,k);\sigma\right),\rho\Big) \leq \frac{1}{k}\right\}. \tag{26}$$

In other words, we define the complexity of a quantum state as the infimum length of the *classical description* of the sequence that converges to such a state. Note that, because Equation (25) considers the infimum of the set, states that are not dcq-computable are assigned infinite complexity. This definition is analogous to the standard classical complexity for real numbers (see Def. 2.3) and it is machine independent in the following sense:

Property 5.1. *Let T and T' be universal dcq-TM, then for any string x and quantum states ρ, σ it holds that*

$$\mathrm{K}_{\mathsf{T}}(\rho \mid x;\sigma) = \mathrm{K}_{\mathsf{T}'}(\rho \mid x;\sigma) + O(1) \tag{27}$$

Proof. Let s be a map that translates T'-programs to T-programs and $c \in \mathbb{N}$ be such that $|s(p)| \leq |p| + c$ for every $p \in \{0,1\}^*$. Then,

$$\begin{aligned}
\mathrm{K}_{\mathsf{T}'}(\rho \mid x;\sigma) + c &= \inf\left\{|p| : \forall k \in \mathbb{N}\ D\left(\mathsf{T}'^{(Q)}\left((p,x,k);\sigma\right),\rho\right) \leq \frac{1}{k}\right\} + c \\
&= \inf\left\{|p| : \forall k \in \mathbb{N}\ D\left(\mathsf{T}^{(Q)}\left((s(p),x,k);\sigma\right),\rho\right) \leq \frac{1}{k}\right\} + c \\
&\geq \inf\left\{|s(p)| : \forall k \in \mathbb{N}\ D\left(\mathsf{T}^{(Q)}\left((s(p),x,k);\sigma\right),\rho\right) \leq \frac{1}{k}\right\} \\
&\geq \inf\left\{|p| : \forall k \in \mathbb{N}\ D\left(\mathsf{T}^{(Q)}\left((p,x,k);\sigma\right),\rho\right) \leq \frac{1}{k}\right\} \\
&= \mathrm{K}_{\mathsf{T}}(\rho \mid x;\sigma).
\end{aligned} \tag{28}$$

By a symmetric argument, since T' is also assumed to enjoy the *s-m-n* property, there is $c' \in \mathbb{N}$ such that:

$$\mathrm{K}_{\mathsf{T}}(\rho \mid x;\sigma) + c' \geq \mathrm{K}_{\mathsf{T}'}(\rho \mid x;\sigma). \tag{29}$$

□

Remark. *From Property 5.1 we can talk about the machine independent algorithmic complexity* K *(without subscript) and consider complexities that are equal up to an additive constant to be equivalent.*

We can turn our attention now into exploring the relationship between the complexity of a quantum state and its respective classical representation. As the following theorem states, they turn out to be same.

Theorem 5.2. *For every dcq-TM T and quantum state ρ it holds that*

$$\mathrm{K}(\boldsymbol{\rho}) = \mathrm{K}(\rho). \tag{30}$$

Proof. The result follows directly from the algorithms in the proof of Theorem 4.1, with the constant term being the length of each algorithm's specification. □

An immediate implication of Theorem 5.2 is that, since the components of a density matrix can have arbitrarily high complexity even for single qubit quantum systems, there is no upper bound for the complexity of an n-qubit quantum state. This is in contrast to the complexity of n-bit strings, which is upper bounded by n.[3]

[3]This is also a significant way in which this definition differs from Vitányi's approach, which is also based on classical descriptions, and for which the complexity of an n-bit pure state is upper bounded by $2n$ (see Theorem 3 from [34])

Does Theorem 5.2 mean that the information value of the state of a quantum system exactly coincides with that of the numbers in its density matrix? The reason the two quantities in Equation (30) coincide is because we are quantifying the complexity of a quantum system using a classical string (the description of the program). Note that having access to the description of a program that computes a given quantum state is not the same as having a copy of the quantum state itself. To get a little better insight about how these two types of objects differ we need to consider their use in the role of a *resource*.

5.1 Conditional complexity and state cloning

Let us continue by evidencing a consequence of the no-cloning theorem. Since the machine receives a specification of the size of its quantum input, we know that for any state ρ it holds that $\mathrm{K}(\rho\,|\,\square;\rho) = 0$, as the program that outputs the quantum input has constant length. On the other hand $\mathrm{K}(\rho\otimes\rho\,|\,\square;\rho) \neq 0$, as there is not a single machine that can copy arbitrary quantum states. In the worst case scenario, such machine would construct the second copy of ρ from scratch, which means that the complexity of duplicating a physical state is upper bounded by the complexity of the state itself:

$$\mathrm{K}(\rho\otimes\rho\,|\,\square;\rho) \leq \mathrm{K}(\rho). \tag{31}$$

This is in contrast to the case in which one would want to copy a classical input, in which case it is always possible to output twice the input. If one had *a priori* access to the classical description of the state, it is possible to run Algorithm 2 repeatedly to compute several copies of ρ. Therefore, copying a quantum state given its associated density matrix $\boldsymbol{\rho}$ has constant complexity:

$$\mathrm{K}(\rho\otimes\rho\,|\,\boldsymbol{\rho};\epsilon) = 0. \tag{32}$$

This exercise suggests that the classical description of a quantum state contains "more" descriptive information that the physical state itself, and thus we can conclude that

Property 5.3. *For any dcq-computable quantum states ρ,σ it holds that*

$$\mathrm{K}(\rho\,|\,\boldsymbol{\sigma};\epsilon) \leq \mathrm{K}(\rho\,|\,\square;\sigma). \tag{33}$$

We now turn our attention to answering if, for arbitrary quantum states, the complexity of copying the state is the same as computing it. In other words, we want to know if the converse of Equation (31) holds. Intuitively, this would mean that any machine that computes the duplicate of a state ρ would necessarily have the descriptive information of that state somehow encoded within it. In order to tackle this question, let us first state a couple of lemmas, the proofs of which can be found in the Appendix. Recall that the fidelity between two quantum states ρ and σ is defined as

$$F(\rho,\sigma) = \left(\mathrm{Tr}\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2. \tag{34}$$

Lemma 5.4. *Let $T:\mathcal{D}(\mathcal{H})\to\mathcal{D}(\mathcal{H}^{\otimes 2})$ be a CPTP map, i.e., a completely positive and trace-preserving map. Define the* copying error $\mathtt{CE}_T$ *of the transformation T on the state $\rho\in\mathcal{D}(\mathcal{H})$ as*

$$\mathtt{CE}_T(\rho) = \sqrt{1-F(\rho^{\otimes 2},T(\rho))}. \tag{35}$$

For any $\rho_1,\rho_2\in\mathcal{D}(\mathcal{H})$ such that $\mathtt{CE}_T(\rho_1)+\mathtt{CE}_T(\rho_2)\leq\varepsilon\leq\frac{1}{4}$ it holds that

$$F(\rho_1,\rho_2)\leq\frac{1}{2}-\sqrt{\frac{1}{4}-\varepsilon}\quad\text{or}\quad F(\rho_1,\rho_2)\geq\frac{1}{2}+\sqrt{\frac{1}{4}-\varepsilon}. \tag{36}$$

Lemma 5.5. *Let $\mathcal{H}$ be an N-dimensional Hilbert space and $A=\{\rho_i\in\mathcal{D}(\mathcal{H})\}$ be a set of density operators such that for $i\neq j$:*

$$F(\rho_i,\rho_j)\leq\delta_0(N), \tag{37}$$

with

$$\delta_0(N) = \frac{1}{N^2}\left(\frac{1-\sqrt{1-\frac{1}{N}}}{1+2^{\left(\frac{3^N+1}{2}\right)}}\right)^4. \tag{38}$$

Then, A cannot have more than N elements.

Lemma 5.6. *For any CPTP map* $T : \mathcal{D}(\mathcal{H}) \to \mathcal{D}\left(\mathcal{H}^{\otimes 2}\right)$*:*

1. $D\left(T(\rho), \rho^{\otimes 2}\right) \leq 1/k \implies \mathtt{CE}_T(\rho) \leq \sqrt{2/k}$.
2. $D\left(T(\rho), \rho^{\otimes 2}\right) \leq 3D(\rho, \rho') + D\left(T(\rho'), \rho'^{\otimes 2}\right)$.

Theorem 5.7. *For any n-qubit quantum state ρ it holds that*

$$\mathrm{K}(\rho) \leq \mathrm{K}(\rho \otimes \rho \,|\, \square; \rho) + 2n + 2\log(n). \tag{39}$$

Proof. To prove the result, we construct a family of algorithms (See Algorithm. 3) with three parameters: a program index t, a qubit number n, and a list index m. Then show that, given an n-qubit state ρ, there exist $t, m \in \mathbb{N}$ satisfying:

$$|t| \leq K(\rho \otimes \rho | \square; \rho) + c; \qquad |m| \leq n, \tag{40}$$

where c is a constant independent of n, such that the program specified by Algorithm 3 with the parameters (t, n, m) computes ρ. We can see Algorithm 3 as a program that computes ρ from ϵ using a program that computes $\rho \otimes \rho$ from ρ as a subroutine.

Parameters: Universal dcq-TM U. Integers t, n, m;
Input: Integer k;

(List preparation phase)
$L \leftarrow (); s \leftarrow 0; \varepsilon_0 \leftarrow \frac{1}{4}\delta_0(2^n)(1 - \delta_0(2^n)); k_0 \leftarrow \left(\frac{4}{\varepsilon_0}\right)^2$;
while *The size of the list* $|L| \leq m$ **do**
  $\boldsymbol{\sigma} \leftarrow \tilde{\Pi}_1(s)$;
  if $\boldsymbol{\sigma}$ *is a* $2^n \times 2^n$ *matrix such that* $\mathtt{CE}_{T_{t,k_0}}(\boldsymbol{\sigma}) \leq \frac{2}{\sqrt{k_0}}$ & $F(\boldsymbol{\sigma}, \boldsymbol{\sigma}') \leq \delta_0$ *for all* $\boldsymbol{\sigma}' \in L$ **then**
    Add $\boldsymbol{\sigma}$ to L;
  end
  $s \leftarrow s + 1$;
end
$\boldsymbol{\rho}_0 \leftarrow L[m]$ (the m-th element of the list L);

(Output preparation phase)
$s \leftarrow 0; \varepsilon_{\text{out}} \leftarrow \min\left\{\delta_0(2^n)(1 - \delta_0(2^n)), \frac{1}{k^2}\left(1 - \frac{1}{k^2}\right)\right\}; k_{\text{out}} \leftarrow \left(\frac{4}{\varepsilon_{\text{out}}}\right)^2$;
while *no matrix* $\boldsymbol{\rho}_{\text{out}}$ *is found* **do**
  $\boldsymbol{\sigma} \leftarrow \tilde{\Pi}_1(s)$;
  if $\boldsymbol{\sigma}$ *is a* $2^n \times 2^n$ *matrix such that* $\mathtt{CE}_{T_{t,k_{\text{out}}}}(\boldsymbol{\sigma}) \leq \frac{2}{\sqrt{k_{\text{out}}}}$ & $F(\boldsymbol{\sigma}, \boldsymbol{\rho}_0) > \frac{16}{\sqrt{k_0}}$ **then**
    $\boldsymbol{\rho}_{\text{out}} \leftarrow \boldsymbol{\sigma}$;
  else
    $s \leftarrow s + 1$;
  end
end
return $(\square; \rho_{\text{out}} = \Pi_1(s))$;

Algorithm 3: Approximation of an n-qubit state given a subroutine that is known to be able to duplicate it and a specification index $m \leq \log(n)$.

Let U be a reference universal dcq-TM. For any $t, k \in \mathbb{N}$ define the family of CPTP maps $\left\{T_{t,k} : \mathbb{C}^{2^n} \to \mathbb{C}^{2^{2n}}\right\}$ as:

$$T_{t,k}(\sigma) = \mathsf{U}_t^Q(k; \sigma). \tag{41}$$

That is, $T_{t,k}$ represents the transformation (understood as going from the quantum input space to the quantum output space) applied by the machine U running the program t on classical input k.

Let us now gain a little intuition on the two phases of Algorithm 3. The list preparation phase goes through the set of directly dcq-computable states looking for states that are "close" to being copied by T_{t,k_0}. Whenever it finds one such state, it adds it to the list L if it is "far" from all the other states

already in the list. The phase ends when the m-th element is added to the list this way and assigns $\boldsymbol{\rho}_0$ to be equal to the last state added. On the other hand, the output preparation phase goes again through the set of directly dcq-computable states, now looking for a state that is both close to the state ρ_0 obtained from the first phase and "close enough" to being copied by $T_{t,k_{\text{out}}}$. To show that the proposed algorithm satisfies the desired properties, we break down the proof into two parts. We want to show that:

i) If U_t computes $\rho \otimes \rho$ from ρ and for large enough m, the list preparation phase will always add a matrix $\boldsymbol{\rho}_0$ for which
$$F(\boldsymbol{\rho}_0, \boldsymbol{\rho}) \geq 1 - \delta_0(2^n), \tag{42}$$
and such matrix will be found before the list exceeds 2^n elements

ii) On input k, if $\boldsymbol{\rho}_0$ satisfies Equation (42) then the output state $\rho_{\text{out}}(k)$ will satisfy
$$D(\rho_{\text{out}}(k), \rho) \leq \frac{1}{k}. \tag{43}$$

Once these properties are proven, the bound in Equation (39) follows by noting that the minimal size program t' that computes $\rho \otimes \rho$ from ρ can be specified by using $|t'| = K_{\mathsf{U}}(\rho \otimes \rho | \square; \rho) \leq K(\rho \otimes \rho | \square; \rho) + c$ bits, and the values of $m \leq 2^n$ and n can be specified with n and $\log(n)$ bits, respectively.

- Proof of statement i)

Note that whenever a matrix $\boldsymbol{\sigma}$ gets added to L it satisfies by Lemma 5.6(1) and the definition of dcq-computability
$$\mathtt{CE}_{T_{t,k_0}}(\boldsymbol{\sigma}) + \mathtt{CE}_{T_{t,k_0}}(\boldsymbol{\rho}) \leq \frac{2}{\sqrt{k_0}} + \sqrt{\frac{2}{k_0}} \leq \frac{4}{\sqrt{k_0}}, \tag{44}$$
invoking Lemma 5.4 we get that
$$F(\boldsymbol{\sigma}, \boldsymbol{\rho}) \leq \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_0}}} \quad \text{or} \quad F(\boldsymbol{\sigma}, \boldsymbol{\rho}) \geq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_0}}}, \tag{45}$$
and substituting k_0 in terms of $\delta_0(2^n)$
$$F(\boldsymbol{\sigma}, \boldsymbol{\rho}) \leq \frac{\delta_0}{4} \quad \text{or} \quad F(\boldsymbol{\sigma}, \boldsymbol{\rho}) \geq 1 - \frac{\delta_0}{4}. \tag{46}$$
Depending on the transformation T_{t,k_0}, the loop inside the **while** statement of the list preparation phase will add some amount of elements to the list before there are no more states that satisfy the stated conditions. Let $l_{\max}$ be the maximum amount of elements that can be added to L in this way. From Lemma 5.5 we know that the $l_{\max} \leq 2^n$ because the largest size of a set $\{\boldsymbol{\sigma}_i\}$ that satisfies $F(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) \leq \delta_0(\mathbb{C}^{2^n})$ for $i \neq j$ is equal to $\dim(\mathbb{C}^{2^n}) = 2^n$. Recall now that the set of n-qubit directly dcq-computable states is dense in the set of n-qubit quantum states and the fact that the function $\tilde{\Pi}_1(s)$ will eventually output every directly dcq-computable state. Let s' be the smallest integer such that
$$D(\boldsymbol{\rho}' = \tilde{\Pi}_1(s'), \boldsymbol{\rho}) \leq \frac{1}{3k_0}, \tag{47}$$
using Lemma 5.6(2) and the fact that U_t computes $\rho \otimes \rho$ from ρ
$$D\left(T_{t,k_0}(\boldsymbol{\rho}'), \boldsymbol{\rho}'^{\otimes 2}\right) \leq 3D(\boldsymbol{\rho}', \boldsymbol{\rho}) + D\left(T_{t,k_0}(\boldsymbol{\rho}), \boldsymbol{\rho}^{\otimes 2}\right) \tag{48}$$
$$\leq \frac{2}{k_0}, \tag{49}$$
and from Lemma 5.6(1) we get
$$\mathtt{CE}_{T_{t,k_0}}(\boldsymbol{\rho}') \leq \frac{2}{\sqrt{k_0}}. \tag{50}$$

Suppose the loop reaches s' without finding $\boldsymbol{\sigma}$ satisfying Equation (42). Then, as shown in Equation (50), at $s = s'$ the matrix $\boldsymbol{\rho}' = \tilde{\Pi}_1(s')$ will satisfy the first condition in the check to be added to L. Additionally, for every $\boldsymbol{\sigma}$ already in L at that point, it will hold that (see Lemma 1 from [27])

$$\sqrt{1 - F(\boldsymbol{\rho}', \boldsymbol{\sigma})} \geq |\underbrace{F(\boldsymbol{\rho}', \boldsymbol{\rho})}_{\geq 1 - \frac{\delta_0}{4}} - \underbrace{F(\boldsymbol{\rho}, \boldsymbol{\sigma})}_{\leq \frac{\delta_0}{4}}| \geq 1 - \frac{\delta_0}{2}, \tag{51}$$

and hence

$$F(\boldsymbol{\rho}', \boldsymbol{\sigma}) \leq \delta_0 \left(1 - \frac{\delta_0}{4}\right) \leq \delta_0, \tag{52}$$

which satisfies the second condition to be added to L. Therefore, for large enough m, the list L will always have an element that satisfies Equation (42). Particularly, there is a value of $m \leq l_{\max} \leq 2^n$ for which the cycle ends when such element is added and sets it to be $\boldsymbol{\rho}_0$.

- Proof of statement ii)

Suppose $\boldsymbol{\rho}_0$ satisfies Equation (42). We proceed to show that any density matrix $\boldsymbol{\sigma}$ that holds both of the conditions in the **if** statement of the output preparation phase must also satisfy

$$D(\boldsymbol{\sigma}, \boldsymbol{\rho}) \leq \frac{1}{k}. \tag{53}$$

Analogously to Equations (44) and (45), whenever a matrix $\boldsymbol{\sigma}$ is found that satisfies $\texttt{CE}_{T_{t,k_{\text{out}}}}(\boldsymbol{\sigma}) \leq 2/\sqrt{k_{\text{out}}}$ we have that

$$\texttt{CE}_{T_{t,k_{\text{out}}}}(\boldsymbol{\sigma}) + \texttt{CE}_{T_{t,k_{\text{out}}}}(\boldsymbol{\rho}) \leq \frac{2}{\sqrt{k_{\text{out}}}} + \sqrt{\frac{2}{k_{\text{out}}}} \leq \frac{4}{\sqrt{k_{\text{out}}}}, \tag{54}$$

and

$$F(\boldsymbol{\sigma}, \boldsymbol{\rho}) \leq \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_{\text{out}}}}} \quad \text{or} \quad F(\boldsymbol{\sigma}, \boldsymbol{\rho}) \geq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_{\text{out}}}}}. \tag{55}$$

In other words, such $\boldsymbol{\sigma}$ is either "very close" to $\boldsymbol{\rho}$ or "very far" from it. Consider the case where left part of Eq (55) is true, we have that (noting that $k_{\text{out}} \geq k_0$)

$$\begin{aligned}
\sqrt{1 - F(\boldsymbol{\rho}_0, \boldsymbol{\sigma})} &\geq | \underbrace{F(\boldsymbol{\rho}_0, \boldsymbol{\rho})}_{\geq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_0}}}} - \underbrace{F(\boldsymbol{\rho}, \boldsymbol{\sigma})}_{\leq \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_{\text{out}}}}}} | \\
&\geq \left| \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_0}}} + \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_{\text{out}}}}} \right| \\
&\geq \sqrt{1 - \frac{16}{\sqrt{k_0}}},
\end{aligned} \tag{56}$$

and therefore

$$F(\boldsymbol{\rho}_0, \boldsymbol{\sigma}) \leq \frac{16}{\sqrt{k_0}}, \tag{57}$$

which means that such $\boldsymbol{\sigma}$ cannot satisfy the second condition of the **if** statement. We conclude that for any $\boldsymbol{\sigma}$ that satisfies both conditions (and hence also for $\boldsymbol{\rho}_{\text{out}}$), it holds that

$$\begin{aligned}
F(\boldsymbol{\sigma}, \boldsymbol{\rho}) &\geq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{4}{\sqrt{k_{\text{out}}}}} \\
&\geq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{1}{k^2}\left(1 - \frac{1}{k^2}\right)} \\
&= 1 - \frac{1}{k^2}.
\end{aligned} \tag{58}$$

Finally, translating into trace distance we get

$$D(\boldsymbol{\sigma}, \boldsymbol{\rho}) \leq \sqrt{1 - F(\boldsymbol{\sigma}, \boldsymbol{\rho})} \leq \frac{1}{k} \tag{59}$$

□

One way to interpret Theorem 5.7 along with Equation (32) is that, for the task of describing $\rho \otimes \rho$, the advantage of having prior access to a single copy of ρ as compared to not having any prior resource state is at most $2n + 2\log(n)$ bits (up to an additive constant). This means that for complex quantum states, for which $\mathrm{K}(\rho) \gg n$, having access to a copy of ρ gives effectively *no advantage* for describing $\rho \otimes \rho$. This is in contrast with classical strings, where having a copy of the string gives all the information needed to compute its duplicate, and can be understood as an algorithmic information version of the no-cloning theorem.

6 On the chain rule and complexity of quantum correlations

One feature of quantum multipartite states is that, in general, they contain correlations between their parts. One way to quantify correlations is through the notion of mutual information. In this section we use the dcq-TM model to generalize the chain rule and extend the concept of algorithmic mutual information to quantum states. Let us first define the concept of prefix for quantum states.

Definition 6.1. *(Prefix of a quantum state)*
Let $\rho_{\langle 1,n\rangle} \in \mathcal{D}\left(\mathbb{C}^{2^n}\right)$. For $m \leq n$, we say that $\sigma \in \mathcal{D}(\mathbb{C}^{2^m})$ is a prefix of $\rho_{\langle 1,n\rangle}$ whenever

$$\sigma = \mathrm{Tr}_{(m+1,\ldots,n)}\left[\rho_{\langle 1,n\rangle}\right]. \tag{60}$$

We can make use of quantum prefixes to study the behaviour of K when considering systems split into two (or more) parts. Note that an n-qubit state has n possible prefixes, which are manifestly defined by the layout of the cells in the quantum tape. When we want to stress the number m of qubits of a prefix we will refer to it as the "m-prefix". To make the separation more explicit in some of the following definitions we will denote the larger n-qubit space as $\mathcal{H}_{AB}$ and the m-qubit prefix space as $\mathcal{H}_A$ so we can write a quantum state and its prefix as ρ_{AB} and ρ_A, respectively.

By analogy with the classical algorithmic information theory, it seems natural to begin our exploration of mutual information by considering the straightforward generalization of two (classically) related quantities:

Definition 6.2. *(Algorithmic mutual information 1)*
Let ρ_A and ρ_B be two quantum states. Define the mutual information of ρ_A respect to ρ_B by:

$$\mathrm{I}^{(1)}(\rho_A : \rho_B) = \mathrm{K}(\rho_A) - \mathrm{K}(\rho_A \mid \boldsymbol{\rho}_B). \tag{61}$$

Definition 6.3. *(Algorithmic mutual information 2)*
Let ρ_{AB} be a bipartite quantum state. Define the mutual information between the partial states ρ_A and ρ_B by:

$$\mathrm{I}^{(2)}(\rho_A : \rho_B) = \mathrm{K}(\rho_A) + \mathrm{K}(\rho_B) - \mathrm{K}(\rho_{AB}). \tag{62}$$

Definition 6.2 compares the algorithmic complexity of the state ρ_A assuming the machine starts with the all zero state in its classical and quantum tapes, versus its complexity assuming the machine has the description of the state ρ_B given as a classical input. The choice of giving the classical description instead the state itself will become clearer as we study its potential in the context of correlations. Evidently $\mathrm{K}(\rho_A|\boldsymbol{\rho}_B)$ is upper bounded by $\mathrm{K}(\rho_A)$, and hence $\mathrm{I}^{(1)}$ is non-negative. On the other hand, Def. 6.3 considers the difference between computing the states ρ_A and ρ_B separately as opposed to computing the (potentially correlated) single state ρ_{AB}. The main conceptual difference between $\mathrm{I}^{(1)}$ and $\mathrm{I}^{(2)}$ is whether to consider the states ρ_A and ρ_B to exist independently or as a part of a greater joint state. Under the classical definition of Kolmogorov complexity, the analogous expressions to (61) and (62) for mutual information turn out to be connected through the chain rule as shown by Equation (7). To explore what is the connection between these two quantities, we may ask ourselves how would a quantum

version of the chain rule look like? A straightforward generalization of Equation (7) would be to replace the string xy with a bipartite state ρ_{AB} leading to an expression of the form:

$$\mathrm{K}(\rho_{AB}) \underset{?}{=} \mathrm{K}(\rho_A) + \mathrm{K}(\rho_B \mid \boldsymbol{\rho}_A) + O\big(\log(n\,\mathrm{K}(\rho_{AB}))\big), \tag{63}$$

but as Theorem 6.1 states, the above equality does not hold for arbitrary joint states.

Theorem 6.1. *There exist no constant g such that*

$$\mathrm{K}(\rho_{AB}) \leq \mathrm{K}(\rho_A) + \mathrm{K}(\rho_B \mid \boldsymbol{\rho}_A) + g\,\log(n\,\mathrm{K}(\rho_{AB})) \tag{64}$$

for every natural number n and n-qubit quantum bipartite state ρ_{AB}.

Proof. Let $g \in \mathbb{R}$, $\rho_{AB} \in \mathcal{D}((\mathbb{C}^2)^{\otimes n=n_a+n_b})$ such that $\rho_A \in \mathcal{D}((\mathbb{C}^2)^{\otimes n_a})$ and $\rho_B \in \mathcal{D}((\mathbb{C}^2)^{\otimes n_b})$. Let $\rho_A = \sum_{i=1}^{2^{n_a}} \alpha_i |i\rangle\langle i|$ and $\rho_B = \sum_{j=1}^{2^{n_b}} \beta_i |j\rangle\langle j|$ be the spectral decomposition of the reduced systems ρ_A and ρ_B. Assume that the Inequality (64) holds for some ρ_{AB} for which ρ_A and ρ_B are proper mixed states and some constant g. We show that there exists then a $\tilde{\rho}_{AB}$ for which the inequality does not hold. Hence, there is no g that satisfies Equation (64) for all bipartite mixed states.

Note that since ρ_A and ρ_B are proper mixed states, there exist r, r' and s, s' such that $r \neq r'$, $s \neq s'$, and $\alpha_r, \alpha_{r'}, \beta s, \beta s'$ are all non-zero. Define the set of states

$$\tilde{\mathcal{C}} = \left\{ \tilde{\rho}^{\lambda}_{AB} \mid \tilde{\rho}^{\lambda}_{AB} = \lambda(\rho_A \otimes \rho_B) + (1-\lambda)\sigma_{AB}, \text{ such that } \lambda \in (0,1) \text{ is computable } \right\}, \tag{65}$$

where $\rho_A \otimes \rho_B = \sum_{i,j=1}^{2^{n_a}, 2^{n_b}} \alpha_i \beta_j |i,j\rangle\langle i,j|$, and

$$\sigma_{AB} = \rho_A \otimes \rho_B + \gamma(|r,s\rangle\langle r',s'| + |r',s'\rangle\langle r,s|), \tag{66}$$

where $\gamma = \sqrt{\alpha_r \alpha_{r'} \beta_s \beta_{s'}}$. Note that σ_{AB} and $\rho_A \otimes \rho_B$ are strictly different density operators and, since the partial traces on the second term of σ_{AB} both vanish, we have that for any $0 \leq \lambda \leq 1$

$$\begin{aligned}
\tilde{\rho}^{\lambda}_A &= \mathrm{Tr}_B \tilde{\rho}^{\lambda}_{AB} \\
&= \lambda \mathrm{Tr}_B[\rho_A \otimes \rho_B] + (1-\lambda)\mathrm{Tr}_B \sigma_{AB} \\
&= \lambda \mathrm{Tr}_B[\rho_A \otimes \rho_B] + (1-\lambda)\left(\mathrm{Tr}_B[\rho_A \otimes \rho_B] + \gamma \mathrm{Tr}_B\left[|r,s\rangle\langle r',s'| + |r',s'\rangle\langle r,s|\right]\right) \\
&= \rho_A,
\end{aligned} \tag{67}$$

and similarly $\tilde{\rho}^{\lambda}_B = \rho_B$. Notice also that, since ρ_A and ρ_B are assumed to be dcq-computable, it follows from Theorem 4.1 all of the $\tilde{\rho}^{\lambda}_{AB}$ are dcq-computable quantum states. Let us denote the value $g_0 = \mathrm{K}(\rho_A) + \mathrm{K}(\rho_B \mid \boldsymbol{\rho}_A)$. We can write the inequality (64) as:

$$\mathrm{K}(\rho_{AB}) - g\log(n\,\mathrm{K}(\rho_{AB})) - g_0 \leq 0. \tag{68}$$

Define now the set

$$\mathcal{C}_m = \{\rho'_{AB} \mid \rho'_{AB} \in \mathcal{D}((\mathbb{C}^2)^{\otimes n}), \mathrm{K}(\rho'_{AB}) \leq m\}. \tag{69}$$

Note that for all m, the set $\mathcal{C}_m$ is finite (of size at most 2^m), whereas the set $\tilde{\mathcal{C}}$ is infinite (its size is equal to the set of computable real numbers between 0 an 1), this means that for any m, there exists λ such that $\tilde{\rho}^{\lambda}_{AB} \notin \mathcal{C}_m$ ($K(\tilde{\rho}^{\lambda}_{AB}) \geq m$). In other words, given a bipartite state ρ_{AB} with mixed partial traces, there are an infinite number of states ρ'_{AB} that share its partial traces and have larger quantum Kolmogorov complexity under the dcq-TM model. Additionally, for any n, g, g_0 we have that

$$\lim_{m\to\infty} \big(m - g\log(nm) - g_0\big) = \infty. \tag{70}$$

Therefore, there is an m_0 such that $m' - g\log(nm') - g_0 > 0$ for any $m' \geq m_0$. The result follows by noting that, from the above argument, there exists λ such that $\mathrm{K}(\tilde{\rho}^{\lambda}_{AB}) \geq m_0$, and hence

$$\mathrm{K}(\tilde{\rho}^{\lambda}_{AB}) > \mathrm{K}(\tilde{\rho}^{\lambda}_A) + \mathrm{K}(\tilde{\rho}^{\lambda}_B \mid \tilde{\boldsymbol{\rho}}^{\lambda}_A) + g\,\log(n\,\mathrm{K}(\tilde{\rho}^{\lambda}_{AB})). \tag{71}$$

□

To understand better relation between Def. 6.2 and Def 6.3, we have to consider the role of correlations in the description of bipartite states. In the Shannon approach, the notion of information is based on the knowledge of system's state: the more we are uncertain of the state of the system, the more we increase our knowledge (i.e., information) upon learning that state. Thus, mutual information between two systems is manifestly linked to correlations between them. In the original Kolmogorov approach, the notion of information is based on our ability to compute a given string. Consequently, mutual information between two strings tells us how much knowing one helps in computing the other. When generalizing to the quantum domain, we see that part of this intuition holds for Def. 6.2 as well. In other words, algorithmic mutual information is quantifying the similarity between the descriptions of two states, complexity wise, and says noting about the correlations in their joint state. Indeed, the quantities on the right hand side of Equation (61) do not depend on the joint state ρ_{AB} at all. On the other hand, Def. 6.3 effectively compares the complexity of computing a (generally) correlated state with its individual parts. One problem with Def. 6.3 as a measure of complexity is that, following Theorem 6.1, the quantity is unbounded from below.

At this point we can ask ourselves: is there a version of the chain rule that holds for quantum states? Such expression would help us motivate an alternative definition for algorithmic mutual information that satisfies a form of symmetry, as well as give us insight into the complexity of correlations in quantum systems. Theorem 6.2 provides a relatively straightforward generalization of Equation (7) by introducing a direct dependence on the joint state, as well as requiring a classical description of the prefix (as opposed to a copy of the state).

Theorem 6.2. *Let n, m be integers such that $1 < m \leq n$, and ρ be an n-qubit quantum state. For any m-prefix σ of ρ it holds that*

$$\mathrm{K}(\rho) = \mathrm{K}(\sigma) + \mathrm{K}(\rho|\boldsymbol{\sigma}) + O\left(\log(n\,\mathrm{K}(\rho))\right). \tag{72}$$

Proof. Let U be a reference universal dcq-TM:
($\leq$):
We specify an algorithm to compute ρ given as input two parameters: the specifications t_1, t_2 of two programs that compute σ and $\rho|\boldsymbol{\sigma}$, respectively. The algorithm works as follows: on input k, simulates the action of $\tilde{\mathsf{U}}_{t_1}(2k;\varepsilon)$ to obtain $\boldsymbol{\sigma_k}$, then simulate $\mathsf{U}_{t_2}((2k,\boldsymbol{\sigma_k});\varepsilon)$ to obtain ρ'_k. Let $\rho_k = \mathsf{U}_{t_2}((2k,\boldsymbol{\sigma});\varepsilon)$, using the triangle inequality and the contractive property of the trace distance we get that

$$\begin{aligned} D(\rho,\rho'_k) &\leq D(\rho,\rho_k) + D(\rho_k,\rho'_k) \\ &\leq D(\rho,\rho_k) + D(\boldsymbol{\sigma},\boldsymbol{\sigma_k}) \\ &\leq \frac{1}{2k} + \frac{1}{2k} = \frac{1}{k}. \end{aligned} \tag{73}$$

So the algorithm correctly computes ρ. Specifying t_1, t_2 can be done using $\mathrm{K}(\sigma) + \mathrm{K}(\rho|\boldsymbol{\sigma})$ bits. Since this is a two-input program, a specification of the length of one of the inputs must be added, which is of size at least $\min\{\log(\mathrm{K}(\sigma)), \log(\mathrm{K}(\rho|\boldsymbol{\sigma}))\} \leq \log(\mathrm{K}(\rho)) + \log(m)$. We can conclude then that

$$\mathrm{K}(\rho) \leq \mathrm{K}(\sigma) + \mathrm{K}(\rho|\boldsymbol{\sigma}) + \log(m\,\mathrm{K}(\rho)) \leq \mathrm{K}(\sigma) + \mathrm{K}(\rho|\boldsymbol{\sigma}) + \log(n\,\mathrm{K}(\rho)) \tag{74}$$

($\geq$):
By *reductio ad absurdum*, assume that for every positive real number c there exist an integer $n > 1$, an n-qubit quantum state ρ, and a prefix σ of ρ such that

$$\mathrm{K}(\rho) < \mathrm{K}(\sigma) + \mathrm{K}(\rho|\boldsymbol{\sigma}) - c\log(n\,\mathrm{K}(\rho)). \tag{75}$$

Let (c, ρ, σ) be such that they satisfy Equation (75) with ρ, σ being n and $m \leq n$ qubit states, respectively. Furthermore, let t_ρ be such that U_{t_ρ} computes ρ and $|t_\rho| = \mathrm{K}(\rho) + O(1)$. In the following treatment, it will be useful to have a shorthand notation for the prefixes of the quantum output of programs. Let $\eta(t,x)$ denote the m-prefix of $\mathsf{U}^Q_t(x;\varepsilon)$. For any positive integer s consider now the following sets:

$$P_s = \left\{ t \in \{0, 2^{|t_\rho|}\} \middle| \text{ for all } 1 \leq x \leq s,\ \mathsf{U}_t(x;\varepsilon) \text{ halts with a quantum output of at least } m \text{ qubits} \right\} \tag{76}$$

$$P_s^\eta = \left\{ t \in P_s \middle| \text{ for all } 1 \leq x \leq s,\ D(\eta(t,x),\eta) \leq \frac{1}{x} \right\} \tag{77}$$

Recall that a set A is computably enumerable if there exists a bijective function $\mathcal{E}_A : \{1, \ldots, |A|\} \to A$ and a machine T_A such that $\mathsf{T}_A(i;\varepsilon) = (\mathcal{E}_A(i);\varepsilon)$ for every $i = \{1, \ldots, |A|\}$. To see that the set P_s^η is computably enumerable given $s, m, |t_\rho|$, and $\boldsymbol{\eta}$, consider a machine that classically simulates instances of all programs up to length $|t_\rho|$ running with all inputs $x \in \{1, \ldots, s\}$ in parallel, keeping track of their simulated outputs as they halt, and outputting the i-th program found to satisfy the conditions in Equation (77). Denote the enumeration of P_s^η defined by this machine as $\mathcal{E}_{s,m,|t_\rho|,\boldsymbol{\eta}}$.

We want to use the set P_s^σ to find an upper bound for $\mathrm{K}(\rho|\boldsymbol{\sigma})$. Because we know that $t_\rho \in P_s^\sigma$ for any s, fix $s = s_0$ and let j be the index assigned to t_ρ by $\mathcal{E}_{s_0,m,|t_\rho|,\boldsymbol{\sigma}}$. We can compute $\rho|\boldsymbol{\sigma}$ as follows: consider a machine that on input $(k;\boldsymbol{\sigma})$ outputs the values of $|t_\rho|$, m, and j, then finds the value of $t_\rho = \mathcal{E}_{s_0,m,|t_\rho|,\boldsymbol{\sigma}}(j)$ and outputs the simulated quantum output of $\mathsf{U}_{t_\rho}(k)$. By recalling that $|m| \leq \log(n)$ we get that

$$\mathrm{K}(\rho|\boldsymbol{\sigma}) \leq \log(|P_{s_0}^\sigma|) + 2\log(\mathrm{K}(\rho)) + 2\log(n) = \log(|P_{s_0}^\sigma|) + 2\log(n\,\mathrm{K}(\rho)). \tag{78}$$

Using Equation (75) we can obtain a bound for the size of $P_{s_0}^\sigma$

$$\log(|P_{s_0}^\sigma|) > \mathrm{K}(\rho) - \mathrm{K}(\sigma) + (c-2)\log(n\,\mathrm{K}(\rho)) = \ell \tag{79}$$

which is independent of the value of s_0. To arrive to a contradiction, we will proceed to show that the bound found in Equation (79) is "too big", in the sense that it allows for a very short description of σ. To do this, we will use the fact that we can compute σ by simulating a program that computes ρ and then outputting only the first m qubits of its quantum output. In general, for input k, we do not really need to run a program that computes ρ, instead we can run any program that outputs a state "close enough" to the output of $\mathsf{U}_{t_\rho}(ak)$ for some natural number $a \geq 2$. We want to construct a small set of programs that are candidates to approximate σ in this way for a given k and then specify one of them by giving its index in an enumeration of that set. First, consider a slightly relaxed version of the P_s^η family of sets

$${}_2P_s^\eta = \left\{ t \in P_s | \text{ for all } 1 \leq x \leq s,\ D(\eta(t,x),\eta) \leq \frac{2}{x} \right\}. \tag{80}$$

Clearly, $P_s^\eta \subseteq {}_2P_s^\eta$ for all parameters s, η, and they are computably enumerable in the same way the P_s^η are. Additionally, note that for any $t \in P_s^\sigma$ and $1 \leq x \leq s$ it holds that

$$D(\eta(t,x),\eta(t_\rho,s)) \leq D(\eta(t,x),\sigma) + D(\sigma,\eta(t_\rho,s)) \leq \frac{1}{x} + \frac{1}{s} \leq \frac{2}{x}, \tag{81}$$

and thus, for any s we have that $P_s^\sigma \subseteq {}_2P_s^{\eta(t_\rho,s)}$ and, following Equation (79), we get that

$$|{}_2P_s^{\eta(t_\rho,s)}| > 2^\ell. \tag{82}$$

Define now the sets

$$\Sigma_s = \left\{ t \in P_s |\ |{}_2P_s^{\eta(t,s)}| > 2^\ell \right\}. \tag{83}$$

The set Σ_s is computably enumerable as shown by Algorithm 5 (see Appendix). Because we know that $t_\rho \in \Sigma_s$ can be used to approximate σ (given the value of m), we can think of the $\Sigma_s \subseteq P_s$ as a subset of candidates for programs to compute σ. Unfortunately, it is not straightforward to find a bound for $|\Sigma_s|$. Instead, define Σ_s^* as the set of states that can be outputted by Algorithm 4, which constructs a list of programs in Σ_s that are all at least $\frac{4}{s}$-distant from each other. The set Σ_s^* is computably enumerable by construction and satisfies the following properties:

1. For any distinct $t, t' \in \Sigma_s^*$ it holds that the associated ${}_2P_s^{\eta(t,s)}$, ${}_2P_s^{\eta(t',s)}$ are disjoint.
2. There is at least one $\tau \in \Sigma_s^*$ such that $D(\eta(t_\rho,s),\eta(\tau,s)) \leq \frac{4}{s}$ and therefore

$$D(\sigma,\eta(\tau,s)) \leq D(\sigma,\eta(t_\rho,s)) + D(\eta(t_\rho,s),\eta(\tau,s)) \leq \frac{1}{s} + \frac{4}{s} = \frac{5}{s} \tag{84}$$

Parameters: Universal dcq-TM U, enumeration $\mathcal{E}^{\Sigma}_{s,m,|t_\rho|}$ of the set Σ_s given $s, m, |t_\rho|, \ell$;
Input: Integers $i, s, m, |t_\rho|, \ell$;
$L \leftarrow ()$, $j \leftarrow 1$;
while $|L| < i$ **do**
  $t \leftarrow \mathcal{E}^{\Sigma}_{s,m,|t_\rho|}(j)$;
  if $D(\eta(t,s), \eta(t',s)) > \frac{4}{s}$ *for each* $t' \in L$ **then**
    Add t to L;
  end
  $j \leftarrow j + 1$;
end
return $L[i]$

Algorithm 4: Enumeration of the set Σ^*_s.

This means that we can compute σ as follows. On input k, let j be the index of a program τ that satisfies Equation (84) for the set Σ^*_{5k}. Directly compute the values of $m, |t_\rho|$, and ℓ; then run Algorithm 4 on input $(j, 5k, m, |t_\rho|, \ell)$ to obtain τ, and finally output the m-prefix of $\mathsf{U}^Q_\tau(5k;\epsilon)$, which by Equation (84) is at most $\frac{1}{k}$-distant from σ. The size of the index $j \leq |\Sigma^*_{5k}|$ can be bounded by

$$|\Sigma^*_{5k}| < \frac{2^{\mathrm{K}(\rho)+O(1)}}{2^\ell}, \tag{85}$$

by noting that

$$2^\ell |\Sigma^*_{5k}| = \sum_{t \in \Sigma^*_{5k}} 2^\ell < \sum_{t \in \Sigma^*_{5k}} {}_2P^{\eta(t,5k)}_{5k} \leq |P_s| \leq 2^{\mathrm{K}(\rho)+O(1)}, \tag{86}$$

where we used the fact that the considered sets ${}_2P^{\eta(t,5k)}_{5k}$ are disjoint. Equation (85) in turn gives us a bound for the complexity of σ

$$\begin{aligned} \mathrm{K}(\sigma) &\leq \log(|\Sigma^*_{5k}|) + 2\log(n) + 2\log(\mathrm{K}(\rho)) + 2\log(\ell) \\ &< \mathrm{K}(\rho) - \ell + 2\log(n\,\mathrm{K}(\rho)) + 2\log(\ell), \end{aligned} \tag{87}$$

substituting ℓ from Equation (79) and cancelling terms

$$\begin{aligned} 0 &< 2\log\big(\mathrm{K}(\rho) - \mathrm{K}(\sigma) + (c-2)\log(n\,\mathrm{K}(\rho))\big) - (c-4)\log(n\,\mathrm{K}(\rho)) \\ &\leq 2\log\big(n\,\mathrm{K}(\rho) + c\log(n\,\mathrm{K}(\rho))\big) - (c-4)\log(n\,\mathrm{K}(\rho)) \\ &\leq 2\log\big((c+1)n\,\mathrm{K}(\rho)\big) - (c-4)\log(n\,\mathrm{K}(\rho)) \\ &\leq 2\log(c+1) - (c-6)\log(n\,\mathrm{K}(\rho)) \\ &\leq 2\log(c+1) - (c-6) \qquad (\text{for } c \geq 6 \text{ and } n > 1). \end{aligned} \tag{88}$$

We arrive to the contradiction by noting that the last expression on Equation (88) is negative for large enough c. □

By applying Theorem 6.2 to a bipartite state ρ_{AB} we get our new version of the chain rule

$$\begin{aligned} \mathrm{K}(\rho_{AB}) &= \mathrm{K}(\rho_A) + \mathrm{K}(\rho_{AB}|\boldsymbol{\rho}_A) + O\left(\log(n\,\mathrm{K}(\rho_{AB}))\right) \\ &= \mathrm{K}(\rho_B) + \mathrm{K}(\rho_{AB}|\boldsymbol{\rho}_B) + O\left(\log(n\,\mathrm{K}(\rho_{AB}))\right), \end{aligned} \tag{89}$$

where the second equality comes from the fact that the state ρ_{AB} and the state obtained by swapping the order of the subsystems A and B are related by a $\log(n)$ overhead, which gets absorbed by the last term in Equation (72). Notably, Equation (89) recovers the form of the classical chain rule Equation (7) when the two subsystems are uncorrelated, that is

$$\mathrm{K}(\rho_A \otimes \rho_B) = \mathrm{K}(\rho_A) + \mathrm{K}(\rho_B|\boldsymbol{\rho}_A) + O\left(\log(n\,\mathrm{K}(\rho_A \otimes \rho_B))\right). \tag{90}$$

Equipped by this result, we can define a correlation-accounting version of mutual information by quantifying by "how much" a state deviates from the classical chain rule, we call this quantity *complexity of correlations*:

Definition 6.4. *(Complexity of quantum correlations)*
Let ρ_{AB} be a bipartite quantum state. Define the complexity of correlations in the subsystem A respect to B by:

$$\begin{aligned} C(A:B) &= \mathrm{K}(\rho_{AB}|\boldsymbol{\rho}_B) - \mathrm{K}(\rho_A|\boldsymbol{\rho}_B) \\ &= I^{(1)}(A:B) - I^{(Q)}(A:B), \end{aligned} \tag{91}$$

where

$$I^{(Q)}(A:B) = \mathrm{K}(\rho_A) - \mathrm{K}(\rho_{AB}|\boldsymbol{\rho}_B). \tag{92}$$

Because $\boldsymbol{\rho}_B$ has the information of the dimension of the space $\mathcal{H}_B$, it is straightforward to see that $C(A:B)$ is non-negative (up to a constant). Note that the quantity $I^{(Q)}(A:B)$ should not be interpreted as an information measure because it can be negative. In fact, it is generically negative, reaching its maximum zero value for product states $\rho_A \otimes \rho_B$ (we denote the quantity by $I^{(Q)}$ to highlight its analogous formal mathematical expression to $I^{(1)}$). From Equations (89) and (90) we know that C also satisfies symmetry of information: for an n-qubit state ρ_{AB} it holds that

$$\begin{aligned} C(A:B) - C(B:A) &= \underbrace{\big(I^{(1)}(A:B) - I^{(1)}(B:A)\big)}_{\underbrace{\in O(\log(n\,\mathrm{K}(\rho_A\otimes\rho_B)))}_{\in O(\log(n\,\mathrm{K}(\rho_{AB})))}} - \underbrace{\big(I^{(Q)}(A:B) - I^{(Q)}(B:A)\big)}_{\in O(\log(n\,\mathrm{K}(\rho_{AB})))} \\ &= O\big(\log(n\,\mathrm{K}(\rho_{AB}))\big). \end{aligned} \tag{93}$$

7 Discussion

In this work, we have laid the foundations of a quantum algorithmic complexity theory based on the dcq-TM model. We started by expanding the definition of the machine to allow it to work with mixed states as inputs/outputs and defined the set of computable quantum states. The choice of our machine model is motivated by two of its main properties: a) the set of machines is naturally discrete and shown to be Turing-complete, and b) there is a single, well defined, halting state for every machine. We mention this properties in contrast to quantum-controlled type of machine models whose description may be parameterized by the set of quantum states associated to a given Hilbert space. If the set of machines is described by a continuous set of states, it can lead to super-Turing capabilities.[4] This issue can be solved by allowing only some discrete subset of states (chosen adequately to output approximations of quantum states within a desired accuracy), as described in [6]. Another, more difficult to solve issue, is that machines running quantum programs do not always have a single, well defined, halting state. Instead, due to the unitary evolution principle of quantum mechanics, they are generally in a superposition of halting and working states, which makes defining computability difficult. This issue has been openly discussed (see, for instance [5, 15, 19, 25]). We make the observation that, in practical scenarios, for adequately modeling real-life quantum computers, we see ourselves as classical users interacting with the quantum machine, which is in line with the dcq-TM model.

Another advantage of the dcq-TM model is that, by having explicitly separated classical and quantum tapes, it lends naturally to comparing two of the approaches mentioned in [35]. That is, it lets us work with quantum states by inputting them as the physical state of the cells of the quantum tape, or alternatively, we can input their classical representation by encoding the corresponding density matrix as a string in the classical tape. We defined the algorithmic complexity of a state ρ as the size of the description of the shortest computable sequence of "outputtable" states that approximates ρ, and showed that it is machine independent. We also showed that any program that computes ρ can be transformed into one that computes $\boldsymbol{\rho}$, and vice-versa; which means that the resulting complexities $\mathrm{K}(\rho)$ and $\mathrm{K}(\boldsymbol{\rho})$ are equivalent. Nevertheless, we found that the conditional complexities $\mathrm{K}(\sigma|\rho)$ and $\mathrm{K}(\sigma|\boldsymbol{\rho})$

[4] Models of computation that allow for performing fully quantum programs have the problem of outputting uncomputable quantum states. For instance, consider an uncomputable real number $p \in (0,1)$. According to the Physical Church-Turing postulate, there is no physical system able to sample the Bernoulli distribution with parameter p. Otherwise, we could approximate arbitrarily p, by the law of large numbers and sampling. However, fully quantum programs would be able to create this distribution by preparing the state $\sqrt{1-p}|0\rangle + \sqrt{p}|1\rangle$ and measuring it on the computational basis.

are, in general, not the same. Instead, the classical representations contain more descriptive information than the physical system when used as a resource. This led us to study the behaviour of complexity for quantum state copying. By combining Equations (31) and (39) we can obtain the relation

$$\mathrm{K}(\rho) = \mathrm{K}(\rho \otimes \rho \mid \rho) + O(n), \tag{94}$$

which can be seen as a statement of how much of an advantage having a copy of a state gives when computing two copies. For any qubit number n and value k of complexity, there is a finite number of quantum states for which $\mathrm{K}(\rho) \leq k$, and an infinite number of states for which $\mathrm{K}(\rho) > k$. It follows that for any $n \in \mathbb{N}$ and integer function $k(n) \in O(n)$, there is an infinite number of n-qubit states ρ for which $\mathrm{K}(\rho) \gg k(n)$. Therefore, for *almost all* quantum states, having access to a copy of the state gives no significant advantage for computing its duplicate.

Finally, we considered notions of quantum algorithmic mutual information, and found that two classically equivalent definitions are no longer equivalent for the quantum case. We noted that the property that connects these two quantities in the classical case, namely the chain rule, does not hold for general bipartite quantum states. We interpreted this discrepancy to be caused by the presence of quantum correlations. To compute a quantum multipartite state it is not enough to be able to compute each of its parts individually, but one must also describe the inherent correlations present in the joint state. Taking this into account, we developed a generalization of the chain rule for quantum states in the dcq-TM model. This new property let us to propose a candidate for a measure of the algorithmic complexity of correlations in quantum systems, which is symmetric up to a logarithmic term in the complexity of the joint state times the number of qubits.

Potential future work directions include the study of the relationship between quantum Kolmogorov complexity and Von Neumann entropy, one way to approach this is through generalization of Brudno's Theorem, which relates the complexity rate with the entropy rate for stationary, ergodic sources. Such analysis has already been done in [3] for Deutsch's machine model [8], which is quantum-controlled. Additionally, further properties of complexity of correlations can be explored and a potential theory of algorithmic complexity of multipartite correlations can be developed.

Acknowledgements

The research on this paper was funded under the FCT project QuantumPrime reference: PTDC/EEI-TEL/8017/2020 (50%) the QuantaGENOMICS project, through the EU H2020 QuantERA II Programme, Grant Agreement No 101017733, CERN/FIS-PAR/0023/2019 (50%). The authors acknowledge Fundação para a Ciência e Tecnologia, Instituto de Telecomunicações Research Unit, ref. UIDB/50008/2020, UIDP/50008/2020 and PEst-OE/EEI/LA0008/2013 and LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020. NP acknowledges the FCT Estímulo ao Emprego Científico grant no. CEECIND/04594/2017/CP1393/CT000. ML acknowledges the PhD scholarship PD/BD/114334/2016.

References

[1] L. Antunes, A. Matos, A. Pinto, A. Souto, and A. Teixeira. One-way functions using algorithmic and classical information theories. *Theory of Computing Systems*, 52(1):162–178, Jan 2013. ISSN 1433-0490. DOI: 10.1007/s00224-012-9418-z.

[2] D. Azevedo, A. M. Rodrigues, H. Canhão, A. M. Carvalho, and A. Souto. Zgli: A pipeline for clustering by compression with application to patient stratification in spondyloarthritis. *Sensors*, 23(3), 2023. ISSN 1424-8220. DOI: 10.3390/s23031219.

[3] F. Benatti, T. Krüger, M. Müller, R. Siegmund-Schultze, and A. Szkoła. Entropy and quantum Kolmogorov complexity: A quantum Brudno's theorem. *Commun. Math. Phys.*, 265(1):437–461, 2006. DOI: 10.1007/s00220-006-0027-z.

[4] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, 1984. DOI: 10.1016/j.tcs.2014.05.025.

[5] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5): 1411–1473, 1997. DOI: 10.1137/S0097539796300921.

[6] A. Berthiaume, W. Dam, and S. Laplante. Quantum Kolmogorov complexity. *Journal of Computer and System Sciences*, 63(2):201–221, 2001. DOI: 10.1006/jcss.2001.1765.

[7] G. Chaitin. On the length of programs for computing finite binary sequences. *J. ACM*, 13(4), 1966. DOI: 10.1145/321356.321363.

[8] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Royal Society of London Proceedings Series A*, 400(1818):97–117, 1985. DOI: 10.1098/rspa.1985.0070.

[9] P. Gács. Quantum algorithmic entropy. *Journal of Physics A: Mathematical and General*, 34(35): 6859, 2001. DOI: 10.1088/0305-4470/34/35/312.

[10] Peter Grünwald and Paul Vitányi. *Algorithmic Information Theory*, pages 289–325. E, January 2008. DOI: 11245/1.297540.

[11] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of modern physics*, 81(2):865, 2009. DOI: 10.1103/RevModPhys.81.865.

[12] A. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1), 1965. DOI: 10.1080/00207166808803030.

[13] T. Lee and A. Romashchenko. Resource bounded symmetry of information revisited. *Theoretical Computer Science*, 345(2):386–405, 2005. ISSN 0304-3975. DOI: 10.1016/j.tcs.2005.07.017. Mathematical Foundations of Computer Science 2004.

[14] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition*. Texts in Computer Science. Springer, 2019. ISBN 978-3-030-11297-4. DOI: 10.1007/978-3-030-11298-1.

[15] Noah Linden and Sandu Popescu. The halting problem for quantum computers. *arXiv preprint quant-ph/9806054*, 1998. DOI: 10.48550/arXiv.quant-ph/9806054.

[16] P. Mateus, A. Sernadas, and A. Souto. Universality of quantum Turing machines with deterministic control. *Journal of Logic and Computation*, 27(1):1–19, 2017. DOI: 10.1093/logcom/exv008.

[17] T. Miyadera. Quantum Kolmogorov complexity and information-disturbance theorem. *Entropy*, 13 (4):778–789, 2011. ISSN 1099-4300. DOI: 10.3390/e13040778.

[18] T. Miyadera and H. Imai. Quantum Kolmogorov complexity and quantum key distribution. *Phys. Rev. A*, 79:012324, Jan 2009. DOI: 10.1103/PhysRevA.79.012324.

[19] Takayuki Miyadera and Masanori Ohya. On halting process of quantum turing machine. *Open Systems & Information Dynamics*, 12(3):261–264, 2005. DOI: 10.1007/s11080-005-0923-2.

[20] Kavan Modi, Aharon Brodutch, Hugo Cable, Tomasz Paterek, and Vlatko Vedral. The classical-quantum boundary for correlations: Discord and related measures. *Reviews of Modern Physics*, 84 (4):1655, 2012. DOI: 10.1103/RevModPhys.84.1655.

[21] C. Mora and H. Briegel. Algorithmic complexity and entanglement of quantum states. *Physical Review Letters*, 95:200503, 2005. DOI: 10.1103/PhysRevLett.95.200503.

[22] C. Mora, H. Briegel, and B. Kraus. Quantum Kolmogorov complexity and its applications. *International Journal of Quantum Information*, 2007. DOI: 10.1142/S0219749907003171.

[23] M Muller. *Quantum Kolmogorov complexity and the quantum Turing machine*. Ph.D. Thesis, Technical University of Berlin, 2007. DOI: 10.48550/arXiv.0712.4377.

[24] M. Müller. Strongly universal quantum Turing machines and invariance of Kolmogorov complexity. *IEEE Transactions on Information Theory*, 54(2):763–780, 2008. ISSN 0018-9448. DOI: 10.1109/TIT.2007.913263.

[25] John M Myers. Can a universal quantum computer be fully quantum? *Physical Review Letters*, 78 (9):1823, 1997. DOI: 10.1103/PhysRevLett.78.1823.

[26] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[27] A Rastegin. A lower bound on the relative error of mixed-state cloning and related operations. *Journal of Optics B: Quantum and Semiclassical Optics*, 5(6):S647, 2003. DOI: 10.1088/1464-4266/5/6/017.

[28] A. Sarkar, Z. Al-Ars, and K. Bertels. Estimating algorithmic information using quantum computing for genomics applications. *Applied Sciences*, 11(6), 2021. ISSN 2076-3417. DOI: 10.3390/app11062696.

[29] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 7 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[30] R. Solomonoff. A formal theory of inductive inference, part i. *Information and Control*, 7(1), 1964. DOI: 10.1016/S0019-9958(64)90223-2.
[31] A. Souto, L. Antunes, P. Mateus, and A. Teixeira. Witness hiding without extractors or simulators. In F. Manea, R. Miller, and D. Nowotka, editors, *Sailing Routes in the World of Computation*, pages 397–409, Cham, 2018. Springer International Publishing. DOI: 10.1007/978-3-319-94418-0˙40.
[32] K. Svozil. Quantum algorithmic information theory. *Journal of Universal Computer Science*, 2(5): 311–346, may 1996. DOI: 10.3217/jucs-002-05-0311.
[33] Andreia Teixeira, Armando Matos, André Souto, and Luís Antunes. Entropy measures vs. Kolmogorov complexity. *Entropy*, 13(3):595–611, 2011. ISSN 1099-4300. DOI: 10.3390/e13030595.
[34] P. Vitányi. Quantum Kolmogorov complexity based on classical descriptions. *IEEE Transactions on Information Theory*, 47(6):2464–2479, 2001. DOI: 10.1109/18.945258.
[35] Paul Vitanyi. Three approaches to the quantitative definition of information in an individual pure quantum state. In *Proceedings 15th Annual IEEE Conference on Computational Complexity*, pages 263–270. IEEE, 2000. DOI: 10.1109/CCC.2000.856757.
[36] A K Zvonkin and L A Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83, dec 1970. DOI: 10.1070/RM1970v025n06ABEH001269.

8 Appendix

8.1 Proof of Lemma 5.4

Definition 8.1. *The Bures angle $\Delta(\rho_1, \rho_2)$ between two quantum states ρ_1, ρ_2 is defined as:*

$$\Delta(\rho_1, \rho_2) = \cos^{-1}(\sqrt{F(\rho_1, \rho_2)}) \tag{95}$$

The Bures angle is a metric in the space of density operators acting on a given Hilbert space. Moreover, it is contractive under quantum operations [26], that is, for any CPTP transformation T it holds that

$$\Delta(T(\rho_1), T(\rho_2)) \leq \Delta(\rho_1, \rho_2). \tag{96}$$

Lemma 8.1. *Let $T : \mathcal{D}(\mathcal{H}) \to \mathcal{D}(\mathcal{H}^{\otimes 2})$ be a CPTP map. Define the* copying error $\texttt{CE}_T$ *of the transformation T on the state $\rho \in \mathcal{D}(\mathcal{H})$ as*

$$\texttt{CE}_T(\rho) = \sqrt{1 - F(\rho^{\otimes 2}, T(\rho))}. \tag{97}$$

For any $\rho_1, \rho_2 \in \mathcal{D}(\mathcal{H})$ such that $\texttt{CE}_T(\rho_1) + \texttt{CE}_T(\rho_2) \leq \varepsilon \leq \frac{1}{4}$ it holds that

$$F(\rho_1, \rho_2) \leq \frac{1}{2} - \sqrt{\frac{1}{4} - \varepsilon} \quad \text{or} \quad F(\rho_1, \rho_2) \geq \frac{1}{2} + \sqrt{\frac{1}{4} - \varepsilon}. \tag{98}$$

Proof. From the triangle inequality of the Bures angle we know that

$$\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2}) \leq \Delta(\rho_1^{\otimes 2}, T(\rho_1)) + \Delta(T(\rho_1), T(\rho_2)) + \Delta(T(\rho_2), \rho_2^{\otimes 2}). \tag{99}$$

Rearranging terms and taking the sin on both sides we get

$$\sin\left(\Delta(\rho_1^{\otimes 2}, T(\rho_1)) + \Delta(T(\rho_2), \rho_2^{\otimes 2})\right) \geq \sin\left(\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2}) - \Delta(T(\rho_1), T(\rho_2))\right). \tag{100}$$

We can bound the left hand side of Equation (100) by recalling that

$$\sin\left(\Delta(\rho_1^{\otimes 2}, T(\rho_1)) + \Delta(T(\rho_2), \rho_2^{\otimes 2})\right) \leq \underbrace{\sin\left(\Delta(\rho_1^{\otimes 2}, T(\rho_1))\right)}_{\texttt{CE}_T(\rho_1)} + \underbrace{\sin\left(\Delta(T(\rho_2), \rho_2^{\otimes 2})\right)}_{\texttt{CE}_T(\rho_2)}. \tag{101}$$

Additionally, noting that

$$\begin{aligned}\sin\left(\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2}) - \Delta(T(\rho_1), T(\rho_2))\right) = &\sin\left(\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2})\right)\cos\left(\Delta(T(\rho_1), T(\rho_2))\right) \\ &- \cos\left(\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2})\right)\sin\left(\Delta(T(\rho_2), \rho_2^{\otimes 2})\right),\end{aligned} \tag{102}$$

and using the following identities/inequalities (derived from the contractive property of the Bures angle and the squaring property of fidelity) we can bound the right hand side of Equation (100) and write it in terms of fidelity:

$$\begin{aligned}
\sin\left(\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2})\right) &= \sqrt{1 - F(\rho_1^{\otimes 2}, \rho_2^{\otimes 2})} = \sqrt{1 - F(\rho_1, \rho_2)^2} \\
\cos\left(\Delta(T(\rho_1), T(\rho_2))\right) &\geq \cos\left(\Delta(\rho_1, \rho_2)\right) = \sqrt{F(\rho_1, \rho_2)} \\
\cos\left(\Delta(\rho_1^{\otimes 2}, \rho_2^{\otimes 2})\right) &= \sqrt{F(\rho_1^{\otimes 2}, \rho_2^{\otimes 2})} = F(\rho_1, \rho_2) \\
\sin\left(\Delta(T(\rho_1), T(\rho_2))\right) &\leq \sin\left(\Delta(\rho_1, \rho_2)\right) = \sqrt{1 - F(\rho_1, \rho_2)}.
\end{aligned} \tag{103}$$

Substituting the inequalities (101) and (103) in the left and right hand sides of Equation (100), respectively, we get:

$$\begin{aligned}
\mathtt{CE}_T(\rho_1) + \mathtt{CE}_T(\rho_2) &\geq \sqrt{(F(\rho_1, \rho_2))(1 - F(\rho_1, \rho_2)^2)} + F(\rho_1, \rho_2)\sqrt{1 - F(\rho_1, \rho_2)} \\
&\geq F(\rho_1, \rho_2)(1 - F(\rho_1, \rho_2)),
\end{aligned} \tag{104}$$

where the last step holds because $F(\rho_1, \rho_2) \in [0, 1]$. Finally, assume that $\mathtt{CE}_T(\rho_1) + \mathtt{CE}_T(\rho_2) \leq \varepsilon \leq \frac{1}{4}$, then the solutions of the quadratic inequality in the interval $[0, 1]$ satisfy

$$\varepsilon \geq F(\rho_1, \rho_2)(1 - F(\rho_1, \rho_2)) \implies F(\rho_1, \rho_2) \leq \frac{1}{2} - \sqrt{\frac{1}{4} - \varepsilon} \quad \text{or} \quad F(\rho_1, \rho_2) \geq \frac{1}{2} + \sqrt{\frac{1}{4} - \varepsilon}. \tag{105}$$

□

8.2 Proof of Lemma 5.5

Lemma 5.5 is a generalization of the basic property which states that for a finite dimensional Hilbert space $\mathcal{H}$, any set $\{\rho_i\}$ of orthogonal states can have at most $\dim(\mathcal{H})$ elements. The lemma states that this property holds even if the states are not quite orthogonal, but instead have pair-wise fidelities upper bounded by $\delta_0(\mathcal{H})$. To prove this result we will start by proving a restricted version of the main lemma, which is then used to prove the more general statement. First, we restrict the sets to consist only of pure states, then we build upon that version removing the requirement of the states to be pure.

Theorem 8.2. *(Gram-Schmidt orthogonalization) Let $\mathcal{H}$ be a Hilbert space and $\{|\psi_i\rangle \in \mathcal{H}\}_{i=1}^k$ a normalized, linearly independent set. For $i = 1, 2, \ldots, k$ define*

$$|\tilde{e}_i\rangle = |\psi_i\rangle - \sum_{j<i} \langle \tilde{e}_j | \psi_i \rangle |\tilde{e}_j\rangle, \tag{106}$$

and

$$|e_i\rangle = \frac{|\tilde{e}_i\rangle}{||\tilde{e}_i\rangle|}. \tag{107}$$

The set $\{|e_i\rangle\}_{i=1}^k$ is an orthonormal set.

The following properties refer directly to the sets defined in Theorem 8.2:

Lemma 8.3. *Let $\mathcal{H}$ be an N-dimensional Hilbert space, $0 \leq \delta \leq 1$, and $\{|\psi_i\rangle \in \mathcal{H}\}$ a normalized, linearly independent set such that for $i \neq j$*

$$|\langle \psi_i | \psi_j \rangle|^2 \leq \delta. \tag{108}$$

The non-normalized Gram-Schmidt vectors $\{|\tilde{e}_i\rangle\}_{i=1}^k$ satisfy for $j < i$

$$|\langle \psi_i | \tilde{e}_j \rangle| \leq a_{j-1}\sqrt{\delta}, \tag{109}$$

where

$$a_j = \begin{cases} 1 & \text{if } j = 0 \\ a_{j-1}^2 + a_{j-1} & \text{if } j \geq 1 \end{cases} \tag{110}$$

Proof. We proceed by induction over the index j. Note that for $j = 1$ and any $i > 1$ we have that

$$|\langle\psi_i|\tilde{e}_j\rangle| = |\langle\psi_i|\psi_1\rangle| \leq \sqrt{\delta} = a_0\sqrt{\delta}. \tag{111}$$

Assume now that for some i and $j < i - 1$ it holds that for all $1 < i' \leq i, 1 \leq j' \leq j$ and $j' < i'$

$$|\langle\psi_{i'}|\tilde{e}_{j'}\rangle| \leq a_{j'-1}\sqrt{\delta}, \tag{112}$$

then

$$\begin{aligned}
|\langle\psi_i|\tilde{e}_{j+1}\rangle| &= \left|\langle\psi_i|\psi_{j+1}\rangle - \sum_{k\leq j}\langle\tilde{e}_k|\psi_{j+1}\rangle\langle\psi_i|\tilde{e}_k\rangle\right| \\
&\leq \underbrace{\left|\langle\psi_i|\psi_{j+1}\rangle\right|}_{\leq\sqrt{\delta}} + \sum_{k=1}^{j}\underbrace{\left|\langle\tilde{e}_k|\psi_{j+1}\rangle\right|\left|\langle\psi_i|\tilde{e}_k\rangle\right|}_{\leq a_{k-1}^2\delta \;\leq\; a_{k-1}^2\sqrt{\delta}} \\
&\leq \left(1 + \sum_{k=1}^{j} a_{k-1}^2\right)\sqrt{\delta} \\
&= a_j\sqrt{\delta}
\end{aligned} \tag{113}$$

□

Corolary 8.4. *Under the assumptions of Lemma 8.3, with* $\delta \leq 2^{-3^N}$

$$|\langle\psi_i|e_i\rangle| \geq \left|1 - 2^{3^N}\delta\right| \tag{114}$$

Proof. From Equation (106) we know that

$$|\psi_i\rangle = |\tilde{e}_i\rangle + \sum_{j<i}\langle\tilde{e}_j|\psi_i\rangle|\tilde{e}_j\rangle. \tag{115}$$

Recalling that the two terms on the right side are mutually orthogonal and that $||\psi_i\rangle| = 1$, it follows that $||\tilde{e}_i\rangle| \leq 1$ and therefore

$$|\langle\psi_i|e_j\rangle| \geq |\langle\psi_i|\tilde{e}_j\rangle|. \tag{116}$$

On the other hand, we can bound the sequence defined in Equation (114) by the doubly exponential sequence

$$a_{j+1} = a_j^2 + a_j \leq 2a_j^2 \leq 2^{3^j}, \tag{117}$$

and hence, by Lemma 8.3

$$\begin{aligned}
|\langle\psi_i|e_i\rangle| \geq |\langle\psi_i|\tilde{e}_i\rangle| &\geq \left|1 - \sum_{j<i}|\langle\tilde{e}_j|\psi_i\rangle|^2\right| \\
&\geq \left|1 - (a_i - 1)\delta\right| \\
&\geq \left|1 - (a_n)\delta\right| \\
&\geq 1 - 2^{3^N}\delta
\end{aligned} \tag{118}$$

□

Lemma 8.5. *(Pure states version) Let* $\mathcal{H}$ *be an* N*-dimensional Hilbert space and* $A = \{|\psi_i\rangle \in \mathcal{H}\}$ *a normalized set such that for* $i \neq j$

$$|\langle\psi_i|\psi_j\rangle|^2 \leq \delta_0^{\text{pure}}(N). \tag{119}$$

with

$$\delta_0^{\text{pure}}(N) = \left(\frac{1 - \sqrt{1 - \frac{1}{N}}}{1 + 2^{\left(\frac{3^N+1}{2}\right)}}\right)^2. \tag{120}$$

Then A *cannot have more than* N *elements.*

Proof. We intend to show that for $\delta_0^{\text{pure}}(N)$ defined as in Equation (120), the set A must be linearly independent. Let $A' = \{|\phi_i\rangle\}_{i=1}^M \subseteq A$ be a maximal linearly independent subset of A and $A'' = A \setminus A'$. We proceed to prove that any other state belonging to the Hilbert space $\mathcal{H}' = \text{Span}(A) = \text{Span}(A')$ cannot satisfy Equation (119), and thus A'' must be the empty set.

Let $\{|e_i\rangle\}_{i=1}^M$ be the set obtained from A' through the Gram-Schmidt process. Because the $|e_i\rangle$ form an orthonormal basis for $\mathcal{H}'$, it holds that for any $|\tilde{\phi}\rangle \in A'' \subseteq \mathcal{H}'$ there exists k such that

$$|\langle\tilde{\phi}|e_k\rangle|^2 \geq \frac{1}{M} \geq \frac{1}{N}. \tag{121}$$

Furthermore, from the triangle inequality of the trace distance

$$D(|\tilde{\phi}\rangle, |e_k\rangle) + D(|e_k\rangle, |\phi_k\rangle) \geq D(|\tilde{\phi}\rangle, |\phi_k\rangle), \tag{122}$$

writing the trace distances in terms of inner products we obtain

$$\sqrt{1 - \left|\langle\tilde{\phi}|e_k\rangle\right|^2} + \sqrt{1 - \left|\langle e_k|\phi_k\rangle\right|^2} \geq \sqrt{1 - \left|\langle\tilde{\phi}|\phi_k\rangle\right|^2}. \tag{123}$$

We can now use the inequalities (114) and (121) to get

$$\sqrt{1 - \frac{1}{N}} + \sqrt{1 - \left(1 - 2^{3^N}\delta_0^{\text{pure}}\right)^2} \geq \sqrt{1 - \left|\langle\tilde{\phi}|\phi_k\rangle\right|^2}, \tag{124}$$

which we can solve for $\left|\langle\tilde{\phi}|\phi_k\rangle\right|^2$ as follows

$$\left|\langle\tilde{\phi}|\phi_k\rangle\right|^2 \geq 1 - \left(\sqrt{1 - \frac{1}{N}} + \sqrt{1 - \left(1 - 2^{3^N}\delta_0^{\text{pure}}\right)^2}\right)^2. \tag{125}$$

All that remains now is to show that the right side of Equation (125) is strictly greater than δ_0^{pure}. Indeed, setting $a = 1 - \sqrt{1 - \frac{1}{N}}$ and $b = 2^{3^N}$ we can rewrite Equation (120) as

$$\sqrt{\delta_0^{\text{pure}}} = a - \sqrt{2b\delta_0^{\text{pure}}}, \tag{126}$$

and we can now use the fact that for all $N \geq 1$ it holds that $0 < \delta_0^{\text{pure}}(N) \leq 2^{-3^N} < 1$ to conclude that

$$\begin{aligned}
\delta_0^{\text{pure}} < \sqrt{\delta_0^{\text{pure}}} &= a - \sqrt{2b\delta_0^{\text{pure}}} \\
&< a - \sqrt{b\delta_0^{\text{pure}}(2 - b\delta_0^{\text{pure}})} \\
&= 1 - \underbrace{\left((1-a) + \sqrt{b\delta_0^{\text{pure}}(2 - b\delta_0^{\text{pure}})}\right)}_{\leq 1} \\
&\leq 1 - \left((1-a) + \sqrt{b\delta_0^{\text{pure}}(2 - b\delta_0^{\text{pure}})}\right)^2 \\
&= 1 - \left(\sqrt{1 - \frac{1}{N}} + \sqrt{1 - \left(1 - 2^{3^N}\delta_0^{\text{pure}}\right)^2}\right)^2 \\
&\leq \left|\langle\tilde{\phi}|\phi_k\rangle\right|^2.
\end{aligned} \tag{127}$$

This means that no state $|\tilde{\phi}\rangle \in \mathcal{H}'$ can satisfy both $|\tilde{\phi}\rangle \notin A'$ and $|\tilde{\phi}\rangle \in A$. Therefore, $A = A'$ is a linearly independent set and as such may have at most N elements. □

Lemma 8.6. *(General statement) Let $\mathcal{H}$ be an N-dimensional Hilbert space and $A = \{\rho_i \in \mathcal{D}(\mathcal{H})\}$ be a set of density operators such that for $i \neq j$:*

$$F(\rho_i, \rho_j) \leq \delta_0(N), \tag{128}$$

with

$$\delta_0(N) = \left(\frac{\delta_0^{\text{pure}}(N)}{N}\right)^2. \tag{129}$$

Then A cannot have more than N elements.

Proof. To prove the result we will proceed as follows: for each $\rho_i \in A$ we will define an associated pure state such that if A satisfies Equation (128) then the associated set I of pure states satisfies Equation (119), which in turn by lemma 8.5 implies that $|I| = |A| \leq N$.

Let $M = |A|$. Following the spectral decomposition theorem; for $i = 1, \ldots, M$, let $\{|\psi_i^k\rangle\}_{k=1}^N$ be a set of orthogonal states and $\{\lambda_i^k\}_{k=1}^N$ a set of non-negative real numbers such that

$$\rho_i = \sum_{k=1}^{N} \lambda_i^k |\psi_i^k\rangle\langle\psi_i^k|. \tag{130}$$

Furthermore, it is easy to check that for $i, j = 1, \ldots, M$ the following is an orthonormal set:

$$\left\{|\psi_j^s(i)\rangle = \sum_{k=1}^{N} \langle\psi_j^s|\psi_i^k\rangle|\psi_i^k\rangle\right\}_{s=1}^{N}. \tag{131}$$

We intend now to use the Uhlmann's theorem to find a bound for inner products $\langle\psi_j^s|\psi_i^k\rangle$ in terms of $\delta_0(N)$. Recall that any state of the form

$$\sum_{k=1}^{N} \sqrt{\lambda_i^k}|\psi_i^k\rangle_{\mathcal{H}}|\phi_i^k\rangle_{\mathcal{H}'} \tag{132}$$

is a purification of ρ_i in $\mathcal{H} \otimes \mathcal{H}'$ for any orthogonal basis $\{|\phi_i^k\rangle \in \mathcal{H}'\}$, where $\mathcal{H}'$ is an ancilliary Hilbert space isomorphic to $\mathcal{H}$. Now, from Ulhmann's theorem:

$$\begin{aligned}
\delta_0(N) \geq F(\rho_i, \rho_j) &= \max_{\Psi_i, \Psi_j} |\langle\Psi_i|\Psi_j\rangle|^2 \\
&\geq \Big| \sum_k \sqrt{\lambda_i^k}_{\mathcal{H}}\langle\psi_i^k|_{\mathcal{H}'}\langle\psi_i^k| \sum_s \sqrt{\lambda_j^s}|\psi_j^s\rangle_{\mathcal{H}}|\psi_j^s(i)\rangle_{\mathcal{H}'} \Big|^2 \\
&= \Big| \sum_{k,s} \sqrt{\lambda_i^k \lambda_j^s}\langle\psi_i^k|\psi_j^s\rangle \underbrace{\langle\psi_i^k|\psi_j^s(i)\rangle}_{\langle\psi_j^s|\psi_i^k\rangle} \Big|^2 \\
&= \Big| \sum_{k,s} \sqrt{\lambda_i^k \lambda_j^s}|\langle\psi_i^k|\psi_j^s\rangle|^2 \Big|^2,
\end{aligned} \tag{133}$$

where the maximum is taken over all purifications Ψ_i, Ψ_j of ρ_i, ρ_j, respectively. Since all terms in the last summation are positive numbers we can conclude that for any $k, s = 1, \ldots, N$:

$$\sqrt{\delta_0(N)} \geq \sqrt{\lambda_i^k \lambda_j^s}\big|\langle\psi_i^k|\psi_j^s\rangle\big|^2. \tag{134}$$

Define now the sets

$$I_i = \left\{|\psi_i^k\rangle \mid \lambda_i^k \geq \frac{1}{N}\right\}; \quad I = \bigcup_{i=1}^{M} I_i, \tag{135}$$

notice that the set I must have *at least* M elements because every density operator acting on an N-dimensional space must have at least one eigenvalue greater or equal to $\frac{1}{N}$. Substituting $|\psi_i^k\rangle \in I_i$ and $|\psi_j^s\rangle \in I_j$ for $i \neq j$ in Equation (134) we obtain

$$\frac{\delta_0^{\text{pure}}(N)}{N} = \sqrt{\delta_0(N)} \geq \sqrt{\lambda_i^k \lambda_j^s}\big|\langle\psi_i^k|\psi_j^s\rangle\big|^2 \geq \frac{1}{N}\big|\langle\psi_i^k|\psi_j^s\rangle\big|^2, \tag{136}$$

and thus

$$\delta_0^{\text{pure}}(N) \geq \big|\langle\psi_i^k|\psi_j^s\rangle\big|^2. \tag{137}$$

This means that the set I satisfies the conditions of lemma 8.5 and hence may have *at most* N elements. We can conclude then

$$M \leq |I| \leq N. \tag{138}$$

□

8.3 Enumeration of the Σ_s

In the proof of Theorem 6.2 the sets

$$\Sigma_s = \{t \in P_s |\ |{}_2P_s^{\eta(t,s)}| > 2^\ell\}, \tag{139}$$

were defined and claimed to be computably enumerable. Algorithm 5 provides a way to enumerate them by calling a computable enumeration $\mathcal{E}_s$ of the set P_s, and subroutine Υ, which on input (s, t, i):

- Outputs the i-th element of some fixed computable enumeration $\Upsilon_{s,t}$ of the set ${}_2P_s^{\eta(t,s)}$ if $i \leq |{}_2P_s^{\eta(t,s)}|$.
- Does not halt if $i > |{}_2P_s^{\eta(t,s)}|$.

Note that a subroutine with the above properties can always be constructed from a program that enumerates a set, since such program can be simulated to produce a certificate of correctness for its output whenever the index is less or equal to the size of the set (by printing the transcript of all its computation), so whenever it halts with an invalid certificate, the simulator can be made to enter an infinite cycle.

Parameters: Subroutine Υ, which in input (s, t, i) outputs the i-th element of some fixed computable enumeration $\Upsilon_{s,t}$ of the set ${}_2P_s^{\eta(t,s)}$. Computable enumeration $\mathcal{E}_s$ of the set P_s. Positive integers s, ℓ ;
Input: Integer $i \geq 1$;
$L \leftarrow (), j \leftarrow 1$;
while $|L| < i$ **do**
  Set n equal to the smallest natural number such that $j < \frac{(n+1)(n+2)}{2}$;
  if $n = 1$ **then**
    $m = s - n$;
  else
    $m = s \mod \frac{n(n+1)}{2}$;
  end
  $t' \leftarrow \mathcal{E}_s(n - m)$;
  Simulate the first m steps of $\Upsilon(s, t', 2^l + 1)$;
  if *The simulated machine reached its halting state and* t' *is not in* L **then**
    Add t' to L
  end
  $j \leftarrow j + 1$;
end
return $L[i]$

Algorithm 5: Enumeration of the set Σ_s given computable enumerations $\mathcal{E}_s, \Upsilon_{s,t}$ of the respective sets P_s and ${}_2P_s^{\eta(t,s)}$, and the values of s and ℓ.