Categorical Tensor Network States

Jacob D. Biamonte,[1,2,3,a] Stephen R. Clark[2,4] and Dieter Jaksch[5,4,2]
[1] *Oxford University Computing Laboratory, Parks Road Oxford, OX1 3QD, United Kingdom*
[2] *Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore*
[3] *ISI Foundation, Torino Italy*
[4] *Keble College, Parks Road, University of Oxford, Oxford OX1 3PG, United Kingdom*
[5] *Clarendon Laboratory, Department of Physics, University of Oxford, Oxford OX1 3PU, United Kingdom*

We examine the use of string diagrams and the mathematics of category theory in the description of quantum states by tensor networks. This approach lead to a unification of several ideas, as well as several results and methods that have not previously appeared in either side of the literature. Our approach enabled the development of a tensor network framework allowing a solution to the quantum decomposition problem which has several appealing features. Specifically, given an n-body quantum state $|\psi\rangle$, we present a new and general method to factor $|\psi\rangle$ into a tensor network of clearly defined building blocks. We use the solution to expose a previously unknown and large class of quantum states which we prove can be sampled efficiently and exactly. This general framework of categorical tensor network states, where a combination of generic and algebraically defined tensors appear, enhances the theory of tensor network states.

PACS numbers: 03.65.Fd, 03.65.Ca, 03.65.Aa

[a] `jacob.biamonte@qubit.org`

I. INTRODUCTION

Tensor network states have recently emerged from Quantum Information Science as a general method to simulate quantum systems using classical computers. By utilizing quantum information concepts such as entanglement and condensed matter concepts like renormalization, several novel algorithms, based on tensor network states (TNS), have been developed which have overcome many pre-existing limitations. These and other related methods have been used to perform highly accurate calculations on a broad class of strongly-correlated systems and have attracted significant interest from several research communities concerned with computer simulations of physical systems.

In this work we develop a tool set and corresponding framework to enhance the range of methods currently used to address problems in many-body physics. In this categorical network model of quantum states, each of the internal components that form the building blocks of the network are defined in terms of their mathematical properties, and these properties are given in terms of equations which have a graphical interpretation. In this way, diagrammatic methods from modern algebra and category theory [1] can be combined with graphical methods currently used in tensor network descriptions of many-body physics. Moreover our results indicate that it may be be advantageous to use “categorical components” within tensor networks, whose algebraic properties can permit a broader means of rewiring networks, and potentially reveal new types of contractible tensor networks. Our results include defining a new graphical calculus on tensors, and exposing their key properties in a “tensor tool box”. We use these tensors to present one solution to the problem of factoring any given quantum state into a tensor network. The graphical properties of this solution then enabled us to expose a wide class of tensor network states that can be sampled in the computational basis efficiently and exactly. Here we list several novel contributions of the present paper.

(i) We introduce a universal class of tensors to the quantum theory and tensor networks literature. We cast several properties and simplification rules applicable for classical networks of this type, into the language of tensor networks. (This fixed collection of tensors and the corresponding Boolean algebra framework we introduce provides the potential for a new tool to construct networks of relevance for problems in quantum information science and condensed matter physics. Recent work has used the collection of tensors we introduce here (arXiv version) together with their rewrite rules to solve models in lattice gauge theory [2] and to study correlator product states [3]).

(ii) We define a new class of quantum states, quantum Boolean states, expressed as: $|\psi\rangle = \sum f(x)|x\rangle$ where f is a switching function and the sum is over all n-long bit strings. A quantum state is called Boolean iff it can be written in a local basis with amplitude coefficients taking only binary values 0 or 1. Examples of states in this class include states such as W and GHZ -states. We develop tools which aid in the study of this class of states.

(iii) We prove that a tensor network representing a Boolean quantum state is determined from the classical network description of the corresponding function. The quantum tensor networks are found by the following method: we let each classical gate act on a linear space and replace the composition of functions, with the contraction of tensors. This technique is detailed in the present work and has very recently been used in [2].

(iv) We present a proof which shows a new universal constructive decomposition of any quantum state, into a network built in terms of tensors in the tool box we introduce.

(v) We present a new and large class of quantum states which we prove to be efficient to sample. A subclass of this class of states includes the widely studied class of correlator product states. This connection (arXiv version) was recently studied in [3]. Exploring this class in further detail is left to future studies.

(vi) Although Boolean algebra has long been used in both classical and quantum computer science, we seem to be the first to tailor its use to the problem of describing quantum states by tensor networks. This has resulted in new proof techniques and also enables one to use methods from the well developed graphical language of classical circuits inside the domain of tensor networks.

To explain the main motivation which prompted us to study tensor network states, let us recall the success of established numerical simulation methods, such as the density matrix renormalization group (DMRG) [4–6] which is based on an elegant class of tensor networks called Matrix Product States (MPS) [7]. For more than 15 years DMRG has been a key method for studying the stationary properties of strongly-correlated 1D quantum systems in regimes far beyond those which can be described with perturbative or mean-field techniques. Exploiting the tensor network structure of MPS has lead to explicit algorithms, such as the Time-Evolving Block Decimation (TEBD) method [8–13], for computing the real-time dynamics of 1D quantum systems. Accurate calculations of out-of-equilibrium properties has proven extremely useful for describing various condensed matter systems [14–16], as well as transport phenomena in ultra-cold atoms in optical lattices [17–19]. Additionally the TEBD method has recently been successfully adapted to the simulation of stochastic classical systems [20], as well as for simulating operators in the Heisenberg picture [6]. Despite these successes, limitations remain in the size, dimensionality, and classes of Hamiltonians that can be simulated with MPS based methods. To overcome these restrictions, several new algorithms have been proposed which are based on different types of tensor network states. Specifically: Projected Entangled Pair States (PEPS) [21–23] which directly generalize the MPS structure to higher dimensions, and the Multi-scale Entanglement Renormalization Ansatz (MERA) [24–27]) which instead utilizes an intuitive hierarchical structure.

Category theory is often used as a unifying language for mathematics [1] and in more recent times has been used to formulate physical theories [28]. One of the strong points of categorical modeling is that it comes equipped with many types of intuitive graphical calculi. (We mention the coherence results [1, 29, 30] and as a matter of convenience, make use of †-compactness. The graphical calculus of categories formally extends to a rigorous tool. See for instance, Selinger's survey of graphical languages for monoidal categories outlining the categories describing quantum theory [30].) The graphical theory behind the types of diagrams we consider here dates to the work of Lafont [31, 32] who built directly on the earlier work by Penrose [33].

A motivating reason for connecting category theory to tensor networks is that, increasingly, both existing and newly developed tensor network algorithms are most easily expressed in terms of informal graphical depictions. This graphical approach can now be complemented and enhanced by exploiting the long existing language of category theory [1, 28, 34]. This immediately enables the application of many established techniques allowing for both a "zoomed out" description exposing known high-level structures, but also enables new "zoomed in" descriptions, exposing "hidden" algebraic structures that have not previously been considered.

We will illustrate our categorical approach to tensor networks by focusing on tensor networks constructed from familiar components, namely Boolean logic gates (and multi-valued logic gates in the case of qudits), applied to this unfamiliar context. To accomplish this goal, we build on ideas across several fields. This includes extending the work by Lafont [35] which was aimed at providing an algebraic theory for classical circuits. (Lafont's work is related to the more recent work on proof theory by Guiraud [36], and is a different direction from other work on applying category theory to classical networks appearing in [37].) The use of symmetric monoidal categories tightens this approach and removes some redundancy in Lafont's graphical lemmas. The application of these results to tensor networks introduces several novel features. The first feature is that once conventional logic circuits are formulated as tensor networks they can be distorted into atemporal configurations since the indices (or legs) of tensors can be bent around arbitrarily. This permits a very compact tensor network representation of a large interesting class of Boolean states such as GHZ-states, W-states and symmetric states [38], using exclusively Boolean gate tensors. A second feature is that once expressed as tensors the corresponding classical logic circuits act on complex valued inputs and outputs, as opposed to just binary values. By permitting arbitrary single-qubit states (general rank-1 tensors) at the output of tensor networks, which are otherwise composed of only switching functions, we arrive at a broader class of generalized Boolean states. We prove in Theorem 23 that this class of states provides an explicit construction method for factoring any given quantum state into a tensor network. As expected, the cost of this exhaustiveness is that the resulting network is, in general, neither efficient in description or in contraction. However, by limiting both the gate count and number of the switching functions comprising the tensor networks to be polynomial in the system size, we obtain a class of states, which we call Generalized Polynomial Boolean States (GPBS see Definition 28), that can be sampled in the computational basis efficiently and exactly (Theorem 29).

a. Manuscript Structure. Next in Section II we quickly review the key concepts introduced in this paper before going into detail in the remaining sections. We continue in Section III by defining the network building blocks: this includes defining some new rank-3 tensors such as the quantum AND-state in Equation (9). We then consider how these components interact in Section IV. This is done in terms of algebraic structures, such as Bialgebras (Section IV B) and Hopf-algebras (Section IV B 1) which are well known to have a purely diagrammatic interpretation. With these definitions in place, in Section V we apply this framework to tensor network theory. As an illustrative example, we consider the W- and GHZ-states using our formalism. Specifically, we consider a particular categorical tensor network for many-body W-states in Section V. A proof of our decomposition theorem for quantum states is given in Section VI. In conclusion, we mention some future directions for work in Section VII. We have included Appendix A on algebras defined on quantum states and Appendix B on the Boolean XOR-algebra.

b. Background Reading. The results appearing in this work were found by tailoring several powerful techniques from modern mathematics: category theory, algebra and co-algebra and applicable results from classical network theory and graphical calculus. Tensor network states are covered in the reviews [21, 25, 39, 40]. For general background on category theory see [1]. For background on Boolean algebra, discrete set functions and circuit theory see [41] and see [42] for background on pseudo Boolean functions and for multi-valued logic see [43]. For background on quantum circuits and quantum computing concepts see [44, 45] and for background on the theory of entanglement see [46]. For the current capabilities of the existing graphical language of tensor network states see e.g. [47, 48] and for work on using ideas related to tensor networks for state preparation of physical systems see [48–50].

II. RESULTS OVERVIEW

In the present Section, we informally review our main results. The idea of translating any given quantum state or operator into a representation in terms of a connected network of algebraically defined components is explained next in Section II A with the corresponding algebraic definitions of these network components over viewed in Section II B. Boolean quantum logic tensors are then introduced in Section II C. We summarize our main results in Section II D.

A. Tensor network representations of quantum states

A qudit is a d-level generalization of a qubit. In physics a quantum state of n-qudits has an exact representation as a rank-n tensor with each of the open legs corresponding to a physical degree of freedom, such as a spin with $(d-1)/2$ energy levels. Such a representation, shown in Figure 1(a) is manifestly inefficient since it will have a number of complex components which grows exponentially with n. The purpose of tensor network states is to decompose this type of structureless rank-n tensor into a network of tensors whose rank is bounded. There are now a number of ways to describe strongly-correlated quantum lattice systems as tensor-networks. As mentioned in the introduction, these include MPS [16, 51, 52], PEPS [39, 53] and MERA [25, 54]. For MPS and PEPS, shown in Figures 1(b) and (c), the resulting network of tensors follows the geometry of the underlying physical system, e.g., a 1D chain and 2D grid, respectively. Alternatively a Tensor Tree Network (TTN) [55, 56] can be employed which has a hierarchical structure where only the bottom layer has open physical legs, as shown in Figure 1(d) for a 1D system and Figure 1(e) for a 2D one. (Each tensor in these networks is otherwise unconstrained, although enforcing some constraints, such as orthogonality, has numerical advantages.) For MERA the network is similar to a TTN, as seen in Figure 1(f) for 1D, but is instead comprised of alternating layers of rank-4 unitary and rank-3 isometric tensors. The central problem faced by all types of tensor networks is that the resulting tensor network for the quantity $\langle\psi|(\mathcal{O}|\psi\rangle)$, where $\mathcal{O}$ is some product operator, needs to be efficiently contractible if any physical results, e.g., expectation values, correlations or probabilities, are to be computed. For MPS and TTN efficient exact contractibility follows from the 1D chain or tree-like geometry, while for MERA it follows from its peculiar causal cone structure resulting from the constraints imposed on the tensors [25]. For PEPS, however, exact contraction is not efficient in general, but can be rendered efficient if approximations are made [39, 53].

In our approach we define a categorical tensor network state (CTNS) generally as any TNS which contains some algebraically constrained tensors along with possible generic ones. Indeed, when recast, certain widely used classes of TNS can be readily exposed as examples of CTNS. Specifically, variants of PEPS have been proposed called string-bond states [57]. Although these string-bond states, like PEPS in general, are not efficiently contractible, they are efficient to sample. By this we mean that for these special cases of PEPS any given amplitude of the resulting state (for a fixed computational basis state) can be extracted exactly and efficiently, in contrast to generic PEPS. This permits variational quantum Monte-Carlo calculations to be performed on string-bond states where the energy of the state is stochastically minimized [57]. This remarkable property follows directly from the use of a tensor, called the COPY-dot, which will form one of several tensors in the fixed toolbox considered in great detail later. As its name suggests, the COPY-dot duplicates inputs states in the

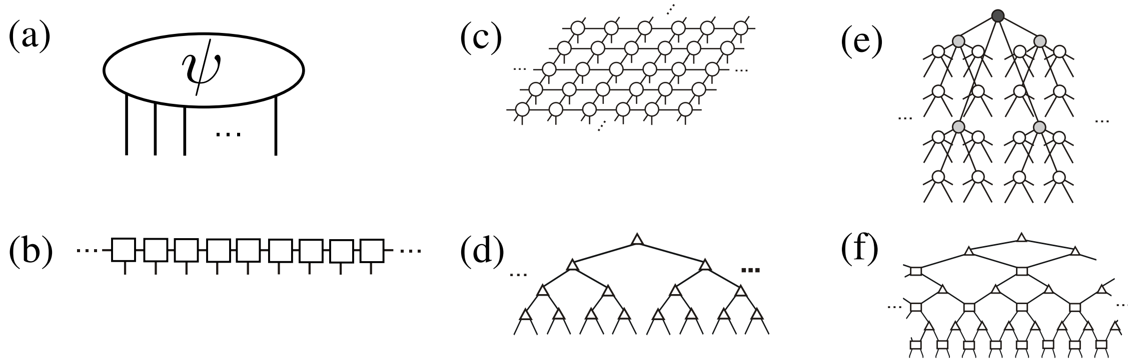


FIG. 1. (a) A generic quantum state $|\psi\rangle$ for n degrees of freedom represented as a tensor with n open legs. (b) A comb-like MPS tensor network for a 1D chain system [51, 52]. (b) A grid-like PEPS tensor network for a 2D lattice system [39, 53]. (d) A TTN for a 1D chain system where only the bottom layer of tensors possess open physical legs [55, 56]. (e) A TTN for a 2D lattice system. (f) A hierarchically structured MERA network for a 1D chain system possessing unitaries (rank-4 tensors) and isometries (rank-3 tensors) [25, 54]. This tensor network can also be generalized to a 2D lattice (not shown).

computational basis, and thus with these inputs breaks up into disconnected components, as depicted in Figure 2(a). By using the COPY-dot as the "glue" for connecting up a TNS, the ability to sample the state efficiently is guaranteed so long as the individual parts connected are themselves contractible. The generality and applicability of this trick can be seen by examining the structure of string-bond states, as well as other types of similar states like entangled-plaquette-states [58] and correlator-product states [59], shown in Figure 2(c)-(e). A long-term aim of this work is that by presenting our toolbox of tensors, entirely new classes of CTNS with similarly desirable contractibility properties can be devised. Indeed we have a useful result in this direction (Theorem 29) by introducing a new class of states, which can also be sampled exactly and efficiently.

On an interesting historical note, to the best of our knowledge, a graphical interpretation of tensors was first pointed out in [33]. A graphical language for describing the manipulations and steps of tensor network based algorithms has become widely used. By introducing new tensors, and by considering their graphical properties, an aim of our work is to extend the existing methods of the diagrammatic methods used. The graphical properties of tensors are defined succinctly via so-called *string diagrams*. As an exemplary illustration, we will consider in great detail CTNS which are composed entirely from a tensor toolbox built from classical Boolean logic gates. By invoking known theorems asserting the universality of multi-valued logic [43] (also called d-state switching), our methods can be readily applied to tensors of any finite dimension. Our approach provides not only an example of the use of well known gates in an unfamiliar context but also illustrates the potential power of having "algebraic components" within a tensor network. The next two sections highlight some of the properties of this toolbox of tensors and reviews our main result showing a new quantum state decomposition using a subset of them. The full details of this work then follow from Section III onwards.

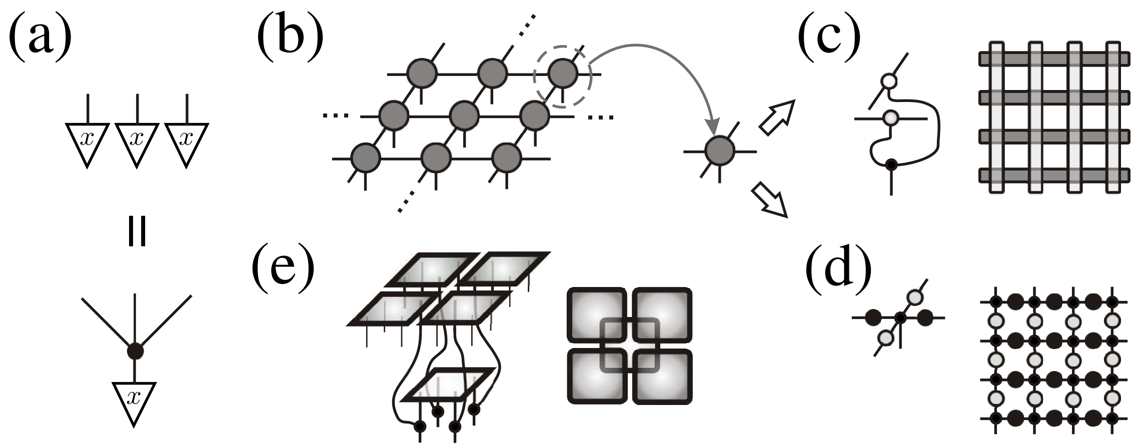


FIG. 2. (a) One of the simplest tensors, called the diagonal in category theory, the COPY-gate or the COPY-dot in circuits, copies computational basis states $|x\rangle$ where $x = 0, 1$ for qubits and $x = 0, 1, ..., d-1$ for qudits. The tensor subsequently breaks up into disconnected states. (b) A generic PEPS in which we expose a single generic rank-5 tensor. This tensor network can neither be contracted nor sampled exactly and efficiently. However, if the tensor has internal structure exploiting the COPY-dot then efficient sampling becomes possible. (c) The tensor breaks up into a vertical and a horizontal rank-3 tensor joined by the COPY-dot. Upon sampling computational basis states the resulting contraction reduces to many isolated MPS, each of which are exactly contractible, for each row and column of the lattice. This type of state is known as a string-bond state and can be readily generalized [57]. (d) An even simpler case is to break the tensor up into four rank-2 tensors joined by a COPY-dot forming a co-called correlator-product state [59]. (e) Finally, outside the PEPS class, there are entangled plaquette states [58] which join up overlapping tensors (in this case rank-4 ones describing a 2×2 plaquette) for each plaquette. Efficient sampling is again possible due to the COPY-dot.

B. Network components fully defined by diagrammatic laws

We will now review the set of tensors that form our universal building blocks. To get an idea of how the tensor calculus will work, consider Figure 3, which forms a presentation of the linear fragment of the Boolean calculus [35]): that is, the calculus of Boolean algebra we represent on quantum states, restricted to the building blocks that can be used to generate linear Boolean functions.

To recover the full Boolean-calculus, we must consider a non-linear Boolean gate: we use the AND-gate. Figure 3 together with Figure 4 form a full presentation of the calculus [35]. The origin and consequences of these relations will be considered in full detail in Section III. The presentations in Figure 3 together with Figure 4 represent a complete set of defining equations [60]. The results we report and the introduction of this new picture calculus into physics has already attracted significant interest and provided a new research direction in categorical quantum mechanics [61, 62].

Proceeding axiomatically we need to add a bit more to the presentation of the Boolean calculus to represent operators and quantum states. This is because e.g. all the diagrams in Figure 3 and 4 are read from the top of the page to the bottom. Our network model of quantum states requires that we are able to turn maps upside down, e.g. transposition. This additional flexibility comes from an added ability to bend wires. We can hence define

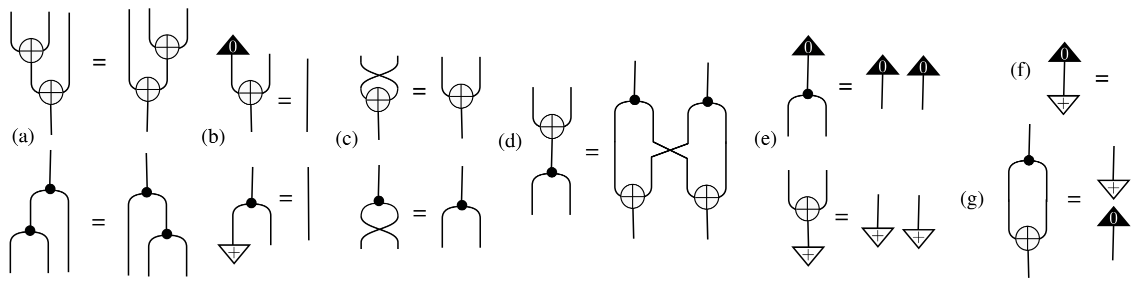


FIG. 3. Read top to bottom. A presentation of the linear fragment of the Boolean calculus. The plus ($\oplus$) dots are XOR and the black ($\bullet$) dots represent COPY. The details of (a)-(g) will be given in Sections III and IV. For instance, (d) represents the bialgebra law and (g) the Hopf-law (in the case of qubits $x \oplus x = 0$, in higher dimensions the units $\langle +|$ becomes $\langle 0| + \langle 1| + \cdots + \langle d-1|$).

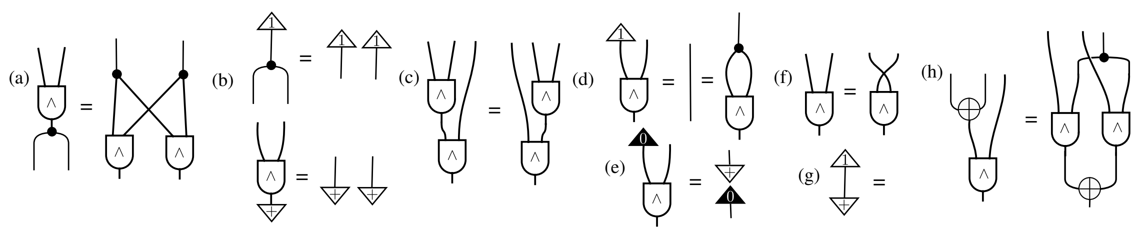
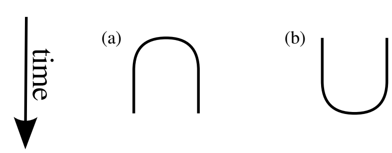


FIG. 4. Diagrams read top to bottom. A presentation of the Boolean-calculus with Figure 3. The details of (a)-(g) will be given in Sections III and IV. For instance, (h) represents distributivity of AND($\wedge$) over XOR ($\oplus$), and (d) shows that $x \wedge x = x$.

transposition graphically (see Figure 18 (d)).

The way forward is to add what category theory refers to as *compact structures* (see Section IV C for details). These compact structures are given diagrammatically as



and as will be explored in Section IV C these two structures allow us to formally bend wires and to define the transpose of a linear map/state, and provide a formal way to reshape a matrix. We understand (a) above as a cup, given as the generalized Bell-state $\sum_{i=0}^{d-1} |ii\rangle$ and (b) above as the so-called cap, Bell-costate $\sum_{i=0}^{d-1} \langle ii|$ or *effect*. (Normalization factors omitted: without loss of generality, we will often omit global scale factors (tensor networks with no open legs). This is done for ease of presentation. We note that for Hilbert space $\mathcal{H}$ there is a natural isomorphism $\mathbb{C} \otimes \mathcal{H} \cong \mathcal{H} \cong \mathcal{H} \otimes \mathbb{C}$.)

Compact structures provide a formal way to bend wires — indeed, we can now connect a diagram represented with an operator with spectral decomposition $\sum_i \beta_i |i\rangle\langle i|$ bend all the open wires (or legs) towards the same direction and it then can be thought of as representing a state ($\sum_i \beta_i |i\rangle|\bar{i}\rangle$ where overbar is complex conjugation), bend them the other way and it then can be thought of as representing a measurement outcome ($\sum_i \beta_i \langle \bar{i}|\langle i|$), that is an effect. (The isomorphism $\sum_i \beta_i \langle \bar{i}|\langle i| \cong \sum_i \beta_i |i\rangle\langle i| \cong \sum_i \beta_i |i\rangle|\bar{i}\rangle$ for a real valued basis becomes $\sum_i \beta_i \langle i|\langle i| \cong \sum_i \beta_i |i\rangle\langle i| \cong \sum_i \beta_i |i\rangle|i\rangle$ which amounts to flipping a bra to a ket and

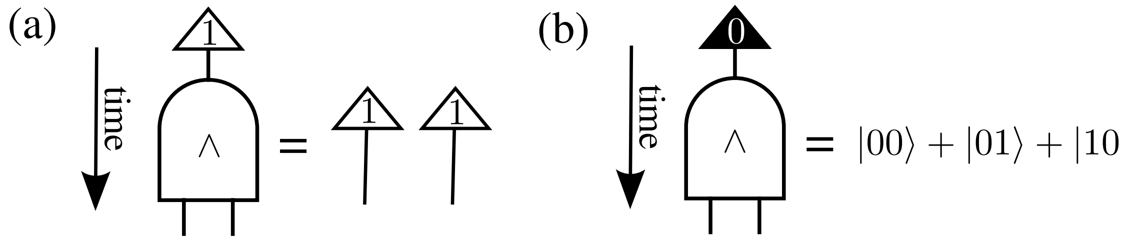


FIG. 5. Example of the Boolean quantum AND-state or tensor. In (a) the network is run backwards (post-selected) to $\langle 1|$ resulting in the product state $|11\rangle$. In (b) the tensor is post-selected to $\langle 0|$ resulting in the entangled state $|00\rangle + |01\rangle + |10\rangle$.

vise versa.) One can also connect inputs to outputs, contracting indices and creating larger and larger networks. With these ingredients in place let us now consider the new class of Boolean quantum states.

C. Boolean and multi-valued tensor network states

To illustrate the idea of defining Boolean and multi-valued logic gates as tensors, consider Figure 5 which depicts a simple but key network building block: the use of the so-called "quantum logic AND-tensor" which we define in Section III D. This is a representation of the familiar Boolean operation in the bit pattern of a tri-qubit quantum state as

$$|\psi_{\mathsf{AND}}\rangle \overset{\text{def}}{=} \sum_{x_1,x_2\in\{0,1\}} |x_1\rangle \otimes |x_2\rangle \otimes |x_1 \wedge x_2\rangle = |000\rangle + |010\rangle + |100\rangle + |111\rangle$$

and hence the truth table of a function is encoded in the bit pattern of the superposition state. This utilizes a linear representation of Boolean gates on quantum states as opposed to the typical direct sum representation common in Boolean algebra.

In this work we are particularly concerned with network constructions as a means to study many-body quantum states by tensor networks. First, we can compose AND-states (by connecting wires and hence contracting tensor indices) — together with NOT-gates, this enables one to create the class of Boolean states in Equation (2). That is, one will realize a network that outputs logical-one (represented here as $|1\rangle$) any time the input qubits represent a desired term in a quantum state (e.g. create a function that outputs logical-one on designated inputs $|00\rangle$, $|01\rangle$ and $|10\rangle$ and zero otherwise as shown in Figure 5). We then insert a $|1\rangle$ at the network output. This procedure recovers the desired Boolean state as illustrated in Figure 6(a) with the resulting state appearing in Equation (1).

$$\sum_{x_1,x_2,\ldots,x_n\in\{0,1\}} \langle 1|f(x_1,x_2,\ldots,x_n)\rangle|x_1,x_2,\ldots,x_n\rangle \tag{1}$$

The network representing the circuit is read backwards from output to input. Alternatively the full class of Boolean states is defined as:

Definition 1 (Boolean many-body qudit states)**.** We define the class of Boolean states as those states which can be expressed up to a global scalar factor in the form (2)

$$\sum_{x_1,x_2,\ldots,x_n\in\{0,1,\ldots,d-1\}} |x_1,x_2,\ldots,x_n\rangle|f(x_1,x_2,\ldots,x_n)\rangle \tag{2}$$

where $f : \mathbb{Z}_d^n \to \mathbb{Z}_d$ is a d-switching function and the sum is taken to be over all variables x_j taking 0 and 1 for qubits and $0, 1, ...d-1$ in the case of d-level qudits (see Figure 6 (a)).

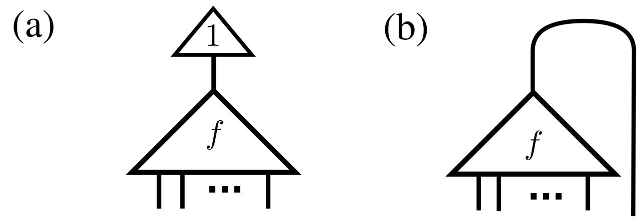


FIG. 6. A general multi-valued qudit state based on a d-switching function f can either be formed as (a) by inputting a logical-one at the output of the circuit as described by Equation (1) or (b) by bending the output of the circuit around to form an input as in Equation (2). For multi-valued qudit states, the boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ becomes a multi-valued qudit function $f : \mathbb{Z}_d^n \to \mathbb{Z}_d$. The network (a) is then post-selected to $\alpha_0|0\rangle + \alpha_1|1\rangle + \cdots \alpha_{d-1}|d-1\rangle$ where $\forall i, \alpha_i = 0/1$.

Examples of Boolean states include the familiar GHZ-state $|00\cdots 0\rangle + |11\cdots 1\rangle$ which on qudits in dimension d becomes

$$|\mathsf{GHZ}_d\rangle = \sum_{i=0}^{d-1} |i\rangle|i\rangle|i\rangle = |0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|1\rangle + \cdots + |d-1\rangle|d-1\rangle|d-1\rangle \tag{3}$$

as well as the W-state $|00\cdots 1\rangle + |01\cdots 0\rangle + \cdots + |10\cdots 0\rangle$ which again on qudits becomes (in Equation (4) the operator $X|m\rangle = |m+1(\text{mod } d)\rangle$ is one way to define negation in higher dimensions. The subscript labels the ket (labeled 1,2 or 3 from left to right) the operator acts on i times.)

$$\begin{aligned} |\mathsf{W}_d\rangle := \sum_{i=1}^{d-1}\sum_{j=1}^{3}(X_j)^i|0\rangle|0\rangle|0\rangle &= |0\rangle|0\rangle|1\rangle + |0\rangle|1\rangle|0\rangle + |1\rangle|0\rangle|0\rangle + |0\rangle|0\rangle|2\rangle + |0\rangle|2\rangle|0\rangle + |2\rangle|0\rangle|0\rangle + \cdots \\ &\cdots + |0\rangle|0\rangle|d-1\rangle + |0\rangle|d-1\rangle|0\rangle + |d-1\rangle|0\rangle|0\rangle \end{aligned} \tag{4}$$

What is clear from this definition is that Boolean states are always composed of equal superpositions of sets of computational basis states, as the allowed scalars take binary values, 0,1. Our main result is that, despite this apparent limitation, tensor networks composed only of Boolean components can nonetheless describe any quantum state. To do this we require a minor extension to include superposition input/output states, e.g. rank-1 tensors of the form $|0\rangle + \beta_1|1\rangle + \cdots + \beta_{d-1}|d-1\rangle$. This gives a universal class of generalized Boolean tensor networks which subsumes the important subclass of Boolean states. This class is then shown to form a nascent example of the exhaustiveness of CTNS and to give rise to a wide class of quantum states that we show are exactly and efficiently sampled.

D. Putting it all together: connecting the dots

The key point to this result is that the introduction of Boolean logic gate tensors into the tensor network context allows the seminal logic gate universality results from classical network theory to be applied in the setting of tensor network states. By extending this result we can construct a solution to the related quantum problem — that is, the decomposition or factorization of any quantum state into a CTNS. Thus our main result is captured by the following statement (see Theorem 23).

Result (Translating quantum states into categorical tensor networks). Given quantum state $|\psi\rangle$, Theorem 23 asserts a constructive method to factor $|\psi\rangle$ into a CTNS constructed from rank-3, rank-2 tensors taken solely from the fixed set in the presentation from Figure 3 and 4 together with arbitrary rank-1 tensors.

This example then demonstrates the exhaustiveness of the most extreme case of the CTNS approach, where almost all tensors are chosen from a small fixed set of tensors with precisely defined algebraic properties. Importantly, in Theorem 29 the form of this general construction is limited in such a way as to provide a new class of states which can be exactly and efficiently sampled.

III. CONSTITUENT NETWORK COMPONENTS: A TENSOR TOOL BOX

Any vector space $\mathcal{V}$ has a dual $\mathcal{V}^*$: this is the space of linear functions f from $\mathcal{V}$ to the ground field $\mathbb{C}$, that is $f : \mathcal{V} \to \mathbb{C}$. This defines the dual uniquely. We must however fix a basis to identify the vector space $\mathcal{V}$ with its dual. Given a basis, any basis vector $|i\rangle$ in $\mathcal{V}$ gives rise to a basis vector $\langle j|$ in $\mathcal{V}^*$ defined by $\langle j|i\rangle = \delta_i^j$ (Kronecker's delta). This defines an isomorphism $\mathcal{V} \to \mathcal{V}^*$ sending $|i\rangle$ to $\langle i|$ and allowing us to identify $\mathcal{V}$ with $\mathcal{V}^*$. In what follows, we will fix a particular arbitrarily chosen basis (called the computational basis in quantum information science). We will now concentrate on Boolean building blocks that are used in our construction.

A. COPY-tensors: the "diagonal"

The copy operation arises in digital circuits [41, 63] and more generally, in the context of category theory and Algebra, where it is called a diagonal in cartesian categories. The operation is readily defined in any finite dimension as

$$\triangle \stackrel{\text{def}}{=} \sum_{i=0}^{d-1} |ii\rangle\langle i| \tag{5}$$

As $|0\rangle$ and $|1\rangle$ are eigenstates of σ^z, we might give $\triangle$ the alternative name of Z-copy. In the case of qubits COPY is succinctly presented by considering the map $\triangle$ that copies σ^z-eigenstates:

$$\triangle : \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2 :: \begin{cases} |0\rangle \mapsto |00\rangle \\ |1\rangle \mapsto |11\rangle \end{cases}$$

This map can be written in operator form as $\triangle : |00\rangle\langle 0| + |11\rangle\langle 1|$ and under cup/cap induced duality (on the right bra) this state becomes a GHZ-state as $|\psi_{\mathsf{GHZ}}\rangle = |000\rangle + |111\rangle \cong |00\rangle\langle 0| + |11\rangle\langle 1|$. The standard properties of COPY are given diagrammatically in Figure 7 and a list of its relevant mathematical properties are found in Table I.

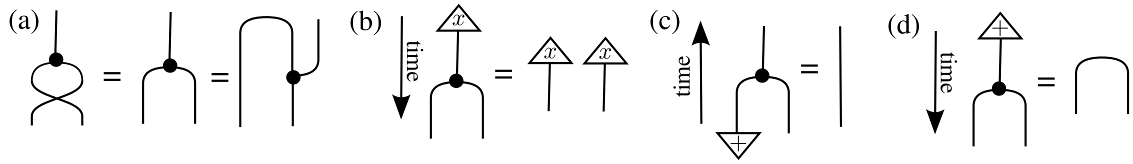
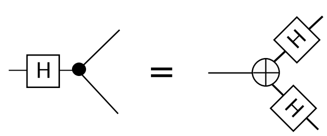


FIG. 7. Salient diagrammatic properties of the COPY-dot. (a) Full-symmetry. (b) Copy points, e.g. $|x\rangle \mapsto |xx\rangle$ for $x = 0, 1$ for qubits and $x = 0, 1, ..., d-1$ for d dimensional qudits. (c) The unit — in this case the unit corresponds to deletion, or a map to the terminal object which is given as $\langle +| \stackrel{\text{def}}{=} \langle 0| + \langle 1|$ for qubits and $\langle +| \stackrel{\text{def}}{=} \langle 0| + \langle 1| + \cdots + \langle d-1|$ for d dimensional qudits (the bi-direction of time is explained later by considering co-diagonals in Section III E). (d) Co-interaction with the unit creates a Bell state $\sum_{i=0}^{d-1} |ii\rangle$. This and the corresponding dual under the dagger form the compact structures of the †-category of quantum theory.

Remark 2 (The COPY-gate from CNOT)**.** *The* CNOT*-gate is defined as* $|0\rangle\langle 0|_1 \otimes \mathbf{1}_2 + |1\rangle\langle 1|_1 \otimes \sigma_2^x$*. We will set the input that the target acts on to* $|0\rangle$ *then calculate* $\mathsf{CNOT}(\mathbf{1}_1 \otimes |0\rangle_2) = |0\rangle\langle 0|_1 \otimes |0\rangle_2 + |1\rangle\langle 1|_1 \otimes |1\rangle_2$*. We have hence defined the desired* COPY *map copying states from the Hilbert space with label* 1 *(subscript) to the joint Hilbert space labeled* 1 *and* 2*.*

B. XOR-tensors: the "addition"

The XOR-gate implements exclusive disjunction or addition (mod 2 for qubits) and is denoted by the symbol $\oplus$ [64, 65]. We note that for multi-valued logic a modulo subtraction gate can also be defined as in [66]. By what could be called "dot-duality" the XOR-gate is simply a Hadamard transform of the COPY-gate, appropriately applied to all of the dots legs. (We denote the discrete Fourier transform gate by $H_{\mathcal{H}} := \frac{1}{\sqrt{d}} \sum_{a,b \in \{0,1,\ldots,d-1\}} e^{i2\pi ab/d} |a\rangle\langle b|_{\mathcal{H}}$, where $d = \dim \mathcal{H}$ is the dimension of the Hilbert space the gate acts in. We can see that $H^T = H$, and that in a qubit system H coincides with the one-qubit Hadamard gate [66].) This can be captured diagrammatically in the slightly different form:



To define the gate on the computational basis, we consider $f(x_1, x_2) = x_1 \oplus x_2$ then $f = 0$ corresponds to $(x_1, x_2) \in \{(0,0), (1,1)\}$ and $f = 1$ corresponds to $(x_1, x_2) \in \{(1,0), (0,1)\}$, where the truth table for XOR follows

x_1	x_2	$f(x_1, x_2) = x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Under cap/cap induced duality, the state defined by XOR is given as

$$|\psi_\oplus\rangle \stackrel{\text{def}}{=} \sum_{x_1, x_2 \in \{0,1\}} |x_1\rangle|x_2\rangle|f(x_1, x_2)\rangle = |000\rangle + |110\rangle + |011\rangle + |101\rangle \quad (6)$$

which is in the GHZ-class by LOCC equivalence viz. $|\psi_{\oplus}\rangle = \mathsf{H} \otimes \mathsf{H} \otimes \mathsf{H}(|000\rangle + |111\rangle)$. The operation of XOR is summarized in Table II. Since the XOR-gate is related to the COPY-gate by a change of basis, its diagrammatic laws have the same structure as those illustrated in Figure 7. The gate acting backwards (co-XOR) is defined on a basis as follows:

$$\oplus : \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2 :: \begin{cases} |0\rangle \mapsto |00\rangle + |11\rangle \\ |1\rangle \mapsto |10\rangle + |01\rangle \end{cases} \qquad \text{or equivalently} \qquad \begin{cases} |+\rangle \mapsto |++\rangle \\ |-\rangle \mapsto |--\rangle \end{cases}$$

C. Generating the affine class of networks

Thus far we have presented the XOR- and COPY- gates. This system allows us to create the linear class of Boolean functions. As explained in the present subsection, this class can be extended to to the affine class by introducing either a gate that acts like an inverter, or by appending a constant $|1\rangle$ into our system. This constant will allow us to use the XOR-gate to create an inverter.

A *complemented Boolean variable* is a Boolean variable that appears in negated form, that is $\neg x$ or written equivalently as $\overline{x}$. Negation of a Boolean variable x can be expressed as the XOR of the variable with constant 1 as $\overline{x} = 1 \oplus x$. Whereas *uncomplimneted Boolean variables* are Boolean variables that do not appear in negated form (e.g. negation is not allowed). Linear Boolean functions contain terms with uncomplemented Boolean variables that appear individually (e.g. variable products are not allowed such as x_1x_2 and higher orders etc., see Section B). Linear Boolean functions take the general form

$$f(x_1, x_2, ..., x_n) = c_1x_1 \oplus c_2x_2 \oplus ... \oplus c_nx_n \tag{7}$$

where the vector $(c_1, c_2, ..., c_n)$ uniquely determines the function. The affine Boolean functions take the same general form as linear functions. However, functions in the affine class allows variables to appear in both complemented and uncomplemented form. Affine Boolean functions take the general form

$$f(x_1, x_2, ..., x_n) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus ... \oplus c_nx_n \tag{8}$$

where $c_0 = 1$ gives functions outside the linear class. From the identities, $1 \oplus 1 = 0$ and $0 \oplus x = x$ we require the introduction of only one constant (c_0), see Appendix B.

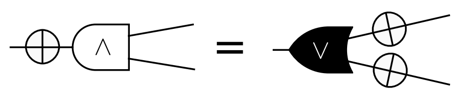
Together, XOR and COPY are not universal for classical circuits. When used together, XOR- and COPY-gates compose to create networks representing the class of linear circuits. The affine circuits are generated by considering the constant $|1\rangle$. The state $|1\rangle$ is indeed copied by the black dot. However, our axiomatization (Figure 3) proceeds through considering the XOR- and COPY-gates together with $|+\rangle$, the unit for COPY and $|0\rangle$ the unit for XOR. It is by appending the constant $|1\rangle$ into the formal system (Figure 3) that the affine class of circuits can be realized.

Remark 3 (Affine functions correspond to a basis)**.** *Each affine function is labeled by a corresponding bit pattern. This can be thought of as labeling the computational basis, as states of the form $|\{0,1\}^n\rangle$ are in correspondence with polynomials in algebraic normal form (see Appendix B).*

D. Quantum AND-state tensors: Boolean universality

The proceeding sections have introduced enough machinery to generate the linear and affine classes of classical circuits. These classes are not universal. To recover a universal system we must add a non-linear Boolean gate. We do this by representing the AND gate as a tensor. The unit for this gate is $\langle 1|$ and so can be used to elevate the linear fragment to the affine class.

The AND gate (that is, $\wedge$) implements logical conjunction [41, 63]. Using again "dot-duality", the AND-gate relates to the OR-gate via De Morgan's law. This can be captured diagrammatically as



To define the gate on the computational basis, we consider $f(x_1, x_2) = x_1 \wedge x_2$ which we write in short hand as x_1x_2. Here $f = 0$ corresponds to $(x_1, x_2) \in \{(0, 0), (0, 1), (1, 0)\}$ and $f = 1$ corresponds to $(x_1, x_2) = (1, 1)$.

Under cap/cap induced duality, the state defined by AND is given as

$$|\psi_\wedge\rangle \stackrel{\text{def}}{=} \sum_{x_1,x_2\in\{0,1\}} |x_1\rangle|x_2\rangle|f(x_1, x_2)\rangle = |000\rangle + |010\rangle + |010\rangle + |111\rangle \tag{9}$$

The key diagrammatic properties of AND are presented in Figure 8 and the gate is summarized in Table III. The gate acting backwards (co-AND) is defined on a basis as follows:

$$\wedge : \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2 :: \begin{cases} |0\rangle \mapsto |00\rangle + |01\rangle + |10\rangle \\ |1\rangle \mapsto |11\rangle \end{cases} \quad \text{or} \quad \begin{cases} |+\rangle \mapsto |++\rangle \\ |-\rangle \mapsto |00\rangle + |01\rangle + |10\rangle - |11\rangle \end{cases}$$

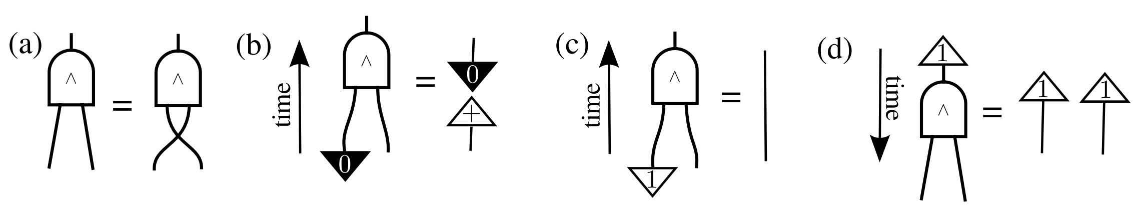


FIG. 8. Salient diagrammatic properties of the AND-tensor. (a) Input-symmetry. (b) Existence of a zero or fixed-point. (c) The unit $|1\rangle$. (d) Co-interaction with the unit creates a product-state. Note that the gate forms a valid quantum operation when run backwards as in (d).

Example 4 (AND-states from Toffoli-gates)**.** The AND-state is readily constructed from the Toffoli gate as illustrated in Figure 9. This allows some interesting states to be created experimentally, for instance, post-selection of the output to $|0\rangle$ would yield the state $|00\rangle + |01\rangle + |10\rangle$. (See the course notes [13] for more on how these techniques can be used as an experimental prescription to generate quantum states. See also Figure 10 for the connection to computationally universal quantum circuits.)

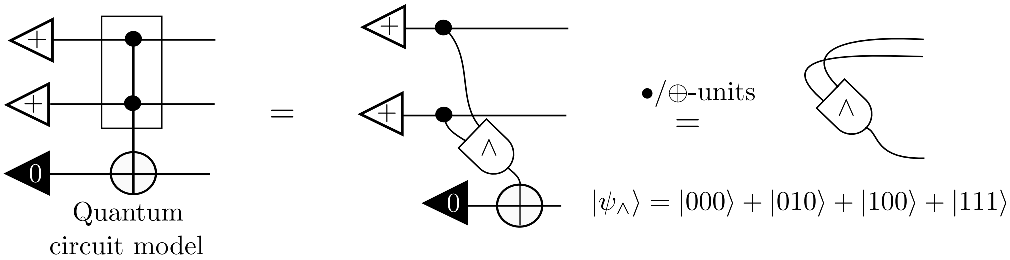


FIG. 9. Illustrates the use of units to prepare the AND-state. Using this state together with single qubit NOT-gates, one can construct any Boolean qubit state as well as any of the states appearing in Table IV. We note that the box around the Toffoli gate (left) is meant to illustrate a difference between our notation and that of quantum circuits.

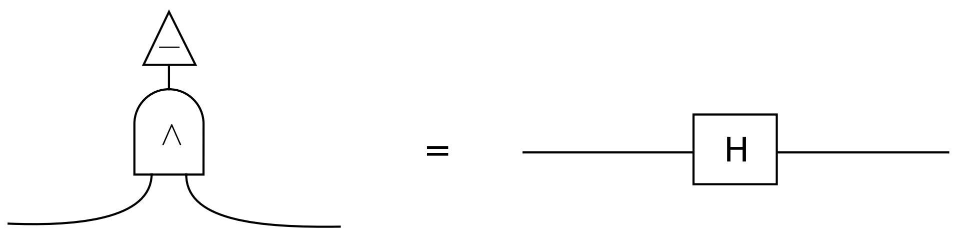


FIG. 10. Hadamard built from the AND-state together with $|-\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. We note that quantum computational universality is already possible by considering simple Hadamard states (e.g. $|\psi_{\mathsf{H}}\rangle = |00\rangle + |01\rangle + |10\rangle - |11\rangle$), COPY- and AND-states, which follows from the proof that Hadamard and Toffoli are universal for quantum circuits [67].

1. *Summary of the* XOR*-algebra on tensors*

We will now present the three previously referenced Tables (I, II and III) which summarize the quantum logic tensors we introduced in the previous subsections (III A, III B and III D). The tables contain entries listing properties that describe how the introduced network components interact. These interactions are defined diagrammatically and explained in Section IV.

Gate Type	Co-copy point(s)	Unit	Co-unit Interaction
COPY	$\|0\rangle$,$\|1\rangle$	$\|+\rangle$	Bell state: $\|00\rangle + \|11\rangle$
Symmetry	Associative	Commutative	Frobenius Algebra
Full	Yes	Yes	Yes (Node Equivalence)

TABLE I. Summary of the COPY-gate from Section III A.

Gate Type	Co-copy point(s)	Unit	Co-unit Interaction
XOR	$\|+\rangle$,$\|-\rangle$	$\|0\rangle$	Bell state: $\|00\rangle + \|11\rangle$
Symmetry	Associative	Commutative	Frobenius Algebra
Full	Yes	Yes	Yes (Node Equivalence)

TABLE II. Summary of the XOR-gate from Section III B.

Gate Type	Co-copy point(s)	Unit	Co-unit Interaction
AND	$\|1\rangle$	$\|1\rangle$	Product state: $\|11\rangle$
Symmetry	Associative	Commutative	Bialgebra Law
Inputs	Yes	Yes	Yes (with GHZ)

TABLE III. Summary of the AND-gate from Section III D.

E. co-COPY: the co-diagonal

What is evident from our subsequent discussions on logic gates is that in the context of tensors, the bending of wires implies that gates can be used both forwards in backwards. We can therefore form tensor networks from Boolean gates in a very different way from classical circuits. Indeed, it becomes possible to flip a COPY operation upside down, that is, instead of having a single leg split into two legs, have two legs merge into one. In terms of tensor networks, co-COPY is simply thought of as being a dual (transpose) to the familiar COPY operation. This is common in algebra: to consider the dual notation to algebra, that is co-algebra. In general, while a product is a joining or pairing (e.g. taking two vectors and producing a third) a co-product is a co-pairing taking a single vector in the space $\mathcal{A}$ and producing a vector in the space $\mathcal{A} \otimes \mathcal{A}$.

Remark 5 (co-algebras [68])**.** *co-algebras are structures that are dual (in the sense of reversing arrows) to unital associative algebras such as* COPY *and* AND *the axioms of which we formulated in terms of picture calculi (Sections III A and III D). Every co-algebra, by (vector space) duality, gives rise to an algebra, and in finite dimensions, this duality goes in both directions.*

Co-COPY can be thought of as applying a delta function in the transition from input to output. That is, given a copy point $x = 0, 1, ..., d-1$ for qudits on dim d. Depicting COPY as the map $\triangle$

$$\triangle\,(|x\rangle) = |x\rangle \otimes |x\rangle \tag{10}$$

we define co-COPY by the map $\bigtriangledown$ such that

$$\bigtriangledown\,(|i\rangle, |j\rangle) = \delta_{ij}|i\rangle \tag{11}$$

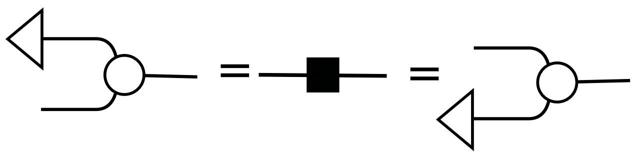
that is, the diagram is mapped to zero (or empty) if the inputs $|i\rangle$, $|j\rangle$ do not agree. This is succinctly expressed in terms of a delta-function dependent on inputs $|i\rangle$, $|j\rangle$ where $i, j = 0, 1, ..., d-1$ for qudits of dim d.

Example 6 (Simple co-pairing)**.** Measurement effects on tripartite quantum systems can be thought of as co-products. This is given as a map from one system (measuring the first) into two systems (the effect this has on the other two). GHZ-states are prototypical examples of co-pairings. In this case, the measurement outcome of $|0\rangle$ ($|1\rangle$) on a single subsystem sends the other qubits to $|00\rangle$ ($|11\rangle$) and by linearity this sends $|+\rangle$ to $|00\rangle + |11\rangle$.

F. The remaining Boolean tensors: NAND-states etc.

We have represented a logical system on tensors — this enables us to represent any Boolean function as a connected network of tensors and hence any Boolean state. We chose as our generators, constant $|1\rangle$, COPY, XOR, AND. Other generators could have also been chosen such as NAND-tensors. Our choice however, was made as a matter of convenience. If we had considered other generators, we could have ended up considering the following cases: weak-units (Definition 7) and fixed point pairs (Definition 9). We note that the NAND-states were used in [69] for fault-tolerant quantum computation — see also [70].

Definition 7 (Weak units)**.** An algebra (or product see Appendix A) on a tripartite state $|\psi\rangle$ has a unit (equivalently, one has that the state is unital) if there exists an effect $\langle\phi|$ which the product acts on to produce an invertible map B, where $B = \mathbf{1}$ (see Example 8). If no such $\langle\phi|$ exists to make $B = \mathbf{1}$, and B has an inverse, we call $\langle\phi|$ a weak unit, and say the state $|\psi\rangle$ is weak unital and if $B \neq \mathbf{1}$ and $B^2 = 1$ we call the algebra on $|\psi\rangle$ unital-involutive. This scenario is given diagrammatically as:



Example 8 (NAND and NOR)**.** NAND and NOR have weak units, respectively given by $|1\rangle$ and $|0\rangle$. These weak units are unital-involutive.

$$|\psi_{\mathsf{NAND}}\rangle = |001\rangle + |011\rangle + |101\rangle + |110\rangle \tag{12}$$

$$|\psi_{\mathsf{NOR}}\rangle = |001\rangle + |010\rangle + |100\rangle + |110\rangle \tag{13}$$

For $|\psi_{\mathsf{NAND}}\rangle$ to have a unit, there must exist a $|\phi\rangle$ such that

$$\langle\overline{\phi}|0\rangle|01\rangle + \langle\overline{\phi}|0\rangle|11\rangle + \langle\overline{\phi}|1\rangle|01\rangle + \langle\overline{\phi}|1\rangle|10\rangle = |00\rangle + |11\rangle \tag{14}$$

and hence no choice of $|\phi\rangle$ makes this possible, thereby confirming the claim.

Definition 9 (Fixed Point Pair)**.** An algebra (see Appendix A) on a tripartite state $|\psi\rangle$ has a *fixed point* if there exists an effect $\langle\phi|$ (the fixed point) which the product acts on to produce a constant output, independent of the other input value. For instance, in Figure 12(c) on the left hand side the effect $\langle 1|$ induces a map (read bottom to top) that sends $|+\rangle \mapsto |1\rangle$. Up to a scalar, this map expands linearly sending both basis effects $\langle 0|$, $\langle 1|$ to to the constant state $|1\rangle$. If the resulting output is the same as the fixed point, we say $\langle\phi|$ has a zero ($|1\rangle$ is the zero for the OR-gate in Figure 12(c)). A fixed point pair consists of two algebras with fixed points, such that the fixed point of one algebra is the unit of the other, and vise versa (see Figure 11 and 12). Diagrammatically this is given in Figure 11.

G. Summarizing: network composition of quantum logic tensors

We have considered sets of universal classical structures in our tensor network model. In classical computer science, a universal set of gates is able to express any n-bit Boolean function

$$f : \mathbb{B}^n \to \mathbb{B} :: (x_1, ..., x_n) \mapsto f(x_1, ..., x_n) \tag{15}$$

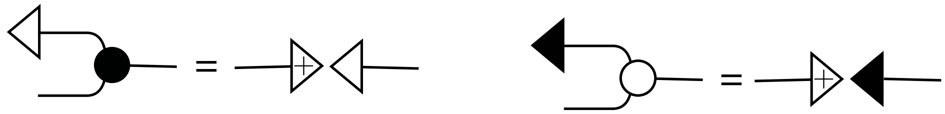


FIG. 11. Diagrammatic equations satisfied by a fixed point pair (see Definition 9).

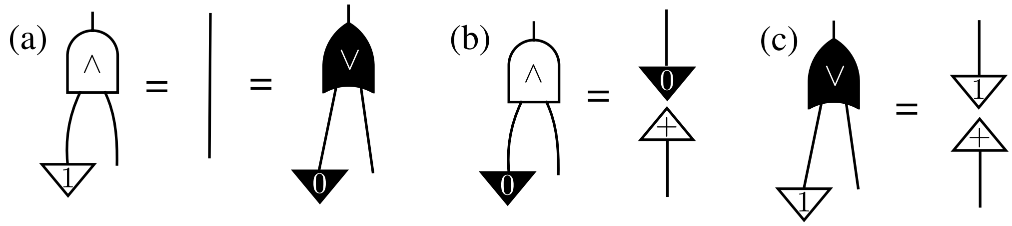


FIG. 12. AND and OR tensors form a fixed point pair. The unit for AND ($|1\rangle$ see a) is the zero for OR (c) and vise versa: the unit of OR ($|0\rangle$ see a) is the zero for AND (b).

where we note that $\mathbb{Z}_2 \cong \mathbb{B}$ allowing us to use the alternative notation for f as $f : \mathbb{Z}_d^n \to \mathbb{Z}_d$ with $d = 2$ for the binary case. Universal sets include $\{$COPY, NAND$\}$, $\{$COPY, AND, NOT$\}$, $\{$COPY, AND, XOR,$|1\rangle\}$, $\{$OR, XNOR,$|1\rangle\}$ and others. One can also consider the states $|\psi\rangle$ formed by the bit patterns of these functions $f(x_1, x_2)$ as

$$|\psi_f\rangle = \sum_{x_1,x_2\in\{0,1\}} |x_1\rangle|x_2\rangle|f(x_1, x_2)\rangle \tag{16}$$

This allows a wide class of states to be constructed effectively. In the following Table (IV) we illustrate the quantum states representing the classical function of two-inputs.

non-linear	linear (Frobenius Algebras)
$\lvert\psi_{\mathsf{AND}}\rangle = \lvert 000\rangle + \lvert 010\rangle + \lvert 100\rangle + \lvert 111\rangle$	
$\lvert\psi_{\mathsf{OR}}\rangle = \lvert 001\rangle + \lvert 011\rangle + \lvert 101\rangle + \lvert 111\rangle$	$\lvert\psi_{\mathsf{XOR}}\rangle = \lvert 000\rangle + \lvert 011\rangle + \lvert 101\rangle + \lvert 110\rangle$
$\lvert\psi_{\mathsf{NAND}}\rangle = \lvert 001\rangle + \lvert 011\rangle + \lvert 101\rangle + \lvert 110\rangle$	$\lvert\psi_{\mathsf{XNOR}}\rangle = \lvert 001\rangle + \lvert 010\rangle + \lvert 100\rangle + \lvert 111\rangle$
$\lvert\psi_{\mathsf{NOR}}\rangle = \lvert 001\rangle + \lvert 010\rangle + \lvert 100\rangle + \lvert 110\rangle$	

TABLE IV. The bit pattern of these quantum states represents a Boolean function (given by the subscript) such that the right most bit is the Boolean functions output, and the two left bits are the functions inputs, and the non-linear Boolean functions are on the left side of the table and the linear functions on the right. Consider the state $|\psi_{\mathsf{AND}}\rangle$, and Boolean variables x_1 and x_2, then the superposition $|\psi_{\mathsf{AND}}\rangle$ encodes the function $|x_1, x_2, x_1 \wedge x_2\rangle$ in each term in the superposition, and $|\psi_{\mathsf{AND}}\rangle = \sum_{x_1,x_2\in\{0,1\}} |x_1, x_2, x_1 \wedge x_2\rangle$. As outlined in the text, cup/cap induced-duality allows us (for instance) to express this state as the operator $|0\rangle\langle 00| + |0\rangle\langle 01| + |0\rangle\langle 01| + |1\rangle\langle 11| :: |x_1, x_2\rangle \mapsto |x_1 \wedge x_2\rangle$ which projects qubit states to the AND of their bit value.

IV. INTERACTION OF THE NETWORK COMPONENTS

Having outlined the Boolean components used in our tensor toolbox we now explore how these tensors interact when connected in a tensor network. The interactions can be defined diagrammatically and given simple rewrite rules for CTNS based on these component tensors.

1. *Merging* COPY*-dots by node equivalence*

COPY-dots are readily generalized to an arbitrary number of input and output legs. As one would rightly suspect, a COPY-dot with n inputs and m outputs corresponds to an $n+m$-partite GHZ-state. Neighboring dots of the same type can be merged into a single dot: this is called node equivalence in digital circuits. COPY-dots represent Frobenius algebras [68, 71, 72]. (Note that the recent online version of [72] was already influenced by the arXiv version of the present manuscript as well as [13, 66].)

Theorem 10 (Node equivalence or spider law)**.** *Given a connected graph with m inputs and n outputs comprised solely of* COPY *dots of equal dimension, this map can be equivalently expressed as a single m-to-n dot, as shown in Figure 13.*

Example 11 (Two-site reduced density operator of n-party GHZ-states)**.** GHZ-states on n-parties have a well known matrix product expression given as

$$|\mathsf{GHZ}_n\rangle = \mathrm{Tr}\left[\begin{pmatrix} |0\rangle & 0 & \cdots & 0 \\ 0 & |1\rangle & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & |d-1\rangle \end{pmatrix}^n\right] = |0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|1\rangle + \cdots + |d-1\rangle|d-1\rangle|d-1\rangle \tag{17}$$

Such MPS networks are known to be efficiently contactable. We note that the networks in Figure 13 do not appear *a priori* to be contractible due to the number of open legs. What makes them contractible (in their present from) is that the tensors obey node equivalence law allowing them to be deformed into a contractible MPS network: see Figure 14. The reduced density matrix of an n-party GHZ-state then becomes (a) in Figure 15 and the expectation value of an observable is shown in (b) where we included the normalisation constant.

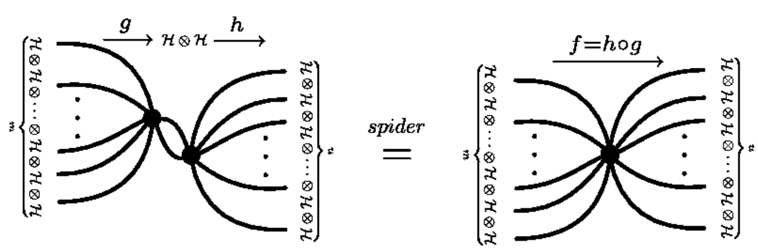


FIG. 13. Node equivalence or spider law [68]. Connected black-dots (•) as well as connected plus-dots (⊕) can be merged and also split apart at will. The intuition for digital or qudit circuits follows by connecting a state $|\phi\rangle$ to one of the legs and iterating over a complete basis $|0\rangle$, $|1\rangle$,...,$|d-1\rangle$.

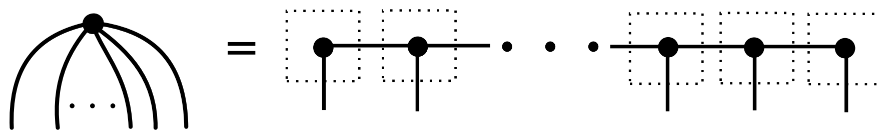


FIG. 14. The GHZ-state tensor is simply a rank-n COPY-dot. Node equivalence implies that this tensor can be deformed into any network geometry including a MPS comb-like structure (right).

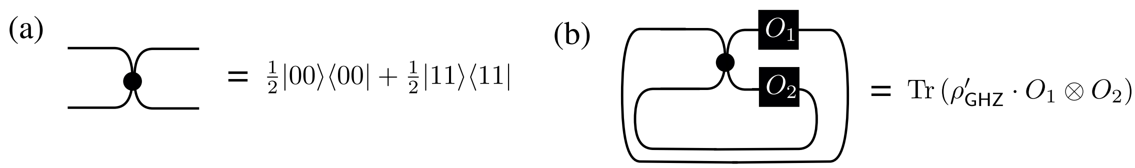


FIG. 15. Reduced density operator. Left (a) reduced density operator ρ'_{GHZ} found from applying the node equivalence law to a n-qubit GHZ-state. Right (b) the expectation value of observable $O_1 \otimes O_2$ found from connecting the observable and connecting the open legs (i.e. taking the trace).

A. Associativity, distributivity and commutativity

The products we have considered are all associative and commutative. As algebras, AND, XOR and COPY are associative, unital commutative algebras. This was already expressed diagrammatically in Figures 3(a) and Figure 4(c). The diagrammatic laws relevant for this subsection represent the following Equations

$$(x_1 \wedge x_2) \wedge x_3 = x_1 \wedge (x_2 \wedge x_3) \tag{18}$$

$$(x_1 \oplus x_2) \oplus x_3 = x_1 \oplus (x_2 \oplus x_3) \tag{19}$$

Distributivity of AND over XOR then becomes (see (h) in Figure 4)

$$(x_1 \oplus x_2) \wedge x_3 = (x_1 \wedge x_2) \oplus (x_1 \wedge x_2) \tag{20}$$

We have commutativity for any product symmetric in its inputs: this is the case for AND and XOR.

B. Bialgebras on tensors

There is a powerful type of algebra that arises in our setting: a bialgebra defined graphically on tensors in Figure 16 (see Kassel, Chapter III [73], [68]).

Such an algebra is simultaneously a unital associative algebra and co-algebra (for the associativity condition see (b) in Figure 16). Specifically, we consider the following two ingredients:

(i): a product (black dot) with a unit (black triangle) see the right hand side of Figure 16(a)

(ii): a co-product (white dot) with a co-unit (white triangle) see the left hand side of Figure 16(a)

To form a bialgebra, these two ingredients above must be characterized by the following four compatibility conditions:

(i): The unit of the black dot is a copy-point of the white dot as in (e) from Figure 16.
(ii): The (co)unit of the white dot is a copy-point of the black dot as in (d) from Figure 16.
(iii): The bialgebra-law is satisfied given in (c) from Figure 16.
(iv): The inner product of the unit (black triangle) and the co-unit (white triangle) is non-zero (not shown in Figure 16).

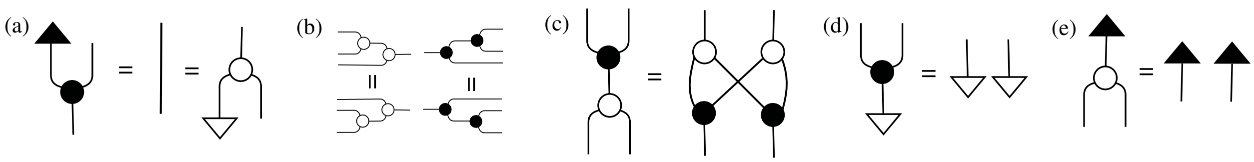


FIG. 16. Bialgebra axioms [68] (scalars are omitted). (a) unit laws (these are of course left and right units); (b) associativity; (c) bialgebra; (d,e) co-COPY points.

Example 12 (GHZ, AND form a bialgebra)**.** We are in a position to study the interaction of GHZ-AND. This interaction satisfies the equations in Figure 16: (a) the bialgebra law; (b) the co-copy point of AND is $|1\rangle$; and (c) the co-interaction with the unit for GHZ creates a compact structure. In addition, (a) and (b) show the copy points for the black GHZ-dot; in (c) we have the unit and fixed point laws.

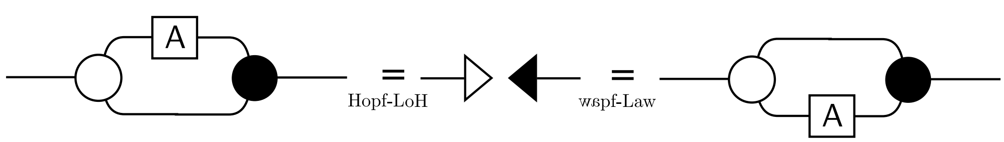
Even if a given product and co-product do not satisfy all of the compatibility conditions (given in (a), (b), (c), (d), (e) in Figure 16), and hence do not form bialgebras, they can still satisfy the bialgebra law which is given in Figure 16(c). Examples of states that satisfy the bialgebra law in Figure 16(c), but are not bialgebras are given in Definition 13. Notice that bialgebra provides a highly constraining characterization of the tensors involved and is tantamount to defining a commutation relation between them.

Definition 13 (Bialgebra Law [68])**.** A pair of quantum states (black, white dots) satisfy the bialgebra law if (c) in Figure 16 holds. The Boolean states, AND, OR, XOR, XNOR, NAND, NOR all satisfy the bialgebra law with COPY.

1. Hopf algebras on tensors

A particularly important class of bialgebras are known as Hopf-algebras [68]. This is characterized by the way in which algebras and co-algebras can interact. This is captured by the Hopf-law, where the linear map A is known as the antipode.

Definition 14 (Hopf-Law [68])**.** A pair of quantum states satisfy the Hopf-Law if an A can be found such that the following equations hold:



Example 15 (XOR and COPY are Hopf-algebras on Boolean States [35])**.** It is well known (see e.g. [35]) that the Boolean state XOR, satisfies the Hopf-algebra law with trivial antipode ($A = \mathbf{1}$) with COPY. Recall Figure 3(g).

C. Bending wires: compact structures

As mentioned in the preliminary section (II), we make use of what's called a *compact structure* in category theory which amounts to introducing cups and caps, to provide a formal way to bend wires and define transposition. See Figures 17 and 18.

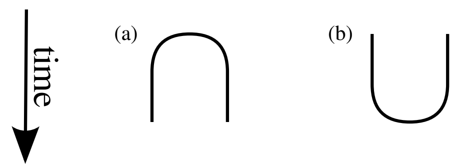
A *compact structure* on an object $\mathcal{H}$ consists of another object $\mathcal{H}^*$ together with a pair of morphisms (note that we use the equation $\mathcal{H}^* = \mathcal{H}$ in Hilbert space making objects self dual which simplifies what follows). The theory is well known in the modern mathematics of category theory and has been used to study teleportation in [74]. Consider

$$\eta_{\mathcal{H}} : \mathbf{1} \longrightarrow \mathcal{H} \otimes \mathcal{H} \qquad\qquad \epsilon_{\mathcal{H}} : \mathcal{H} \otimes \mathcal{H} \longrightarrow \mathbf{1}$$

where the standard representation in Hilbert space with dimension d and basis $\{|i\rangle\}$ is given by

$$\eta_{\mathcal{H}} = \sum_{i=0}^{d-1} |i\rangle \otimes |i\rangle \qquad\qquad \epsilon_{\mathcal{H}} = \sum_{i=0}^{d-1} \langle i| \otimes \langle i|$$

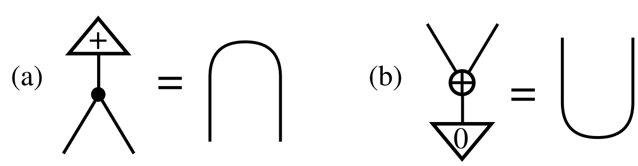
and in string diagrams (read from the top to the bottom of the page) as



These cups and caps give rise to cup/cap-induced duality: this amounts to being able to create a linear map that "flips" a bra to a ket (and vise versa) and at the same time taking an (anti-linear) complex conjugate. In other words, the cap $\sum_{i=0}^{1} \langle ii|$ sends quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to $\alpha\langle 0| + \beta\langle 1|$ which is equal to the complex conjugate of $|\psi\rangle^\dagger = \langle\psi| = \overline{\alpha}\langle 0| + \overline{\beta}\langle 1|$. Diagrammatically, the dagger is given by mirroring operators across the page, whereas transposition is given by bending wire(s). Clearly, $\langle\overline{\psi}| = \alpha\langle 0| + \beta\langle 1|$.

In the case of relating the Bell-states and effects to the identity operator, under cup/cap-induced duality, we flip the second ket on $\eta_{\mathcal{H}}$ and the first bra on $\epsilon_{\mathcal{H}}$. This relates these maps and the identity $\mathbf{1}_{\mathcal{H}}$ of the Hilbert space: that is, we can fix a basis and construct invertible maps sending $\eta_{\mathcal{H}} \cong \mathbf{1}_{\mathcal{H}} \cong \epsilon_{\mathcal{H}}$. More generally, the maps $\eta_{\mathcal{H}}$ and $\epsilon_{\mathcal{H}}$ satisfy the equations given in Figure 17 and their duals under the dagger.

A second way to introduce cups and caps is to consider a *Frobenius form* [68] on either of the structures in the linear fragment from Figure 3 (COPY and XOR). This is simply a functional that turns a product/co-product into a cup/cap. This allows one to recover the above compact structures (that is, the cups and caps given above) as



Again, we will use these cups and caps as a formal way to bend wires in tensor networks: this can be thought of simply as a reshape of a matrix.

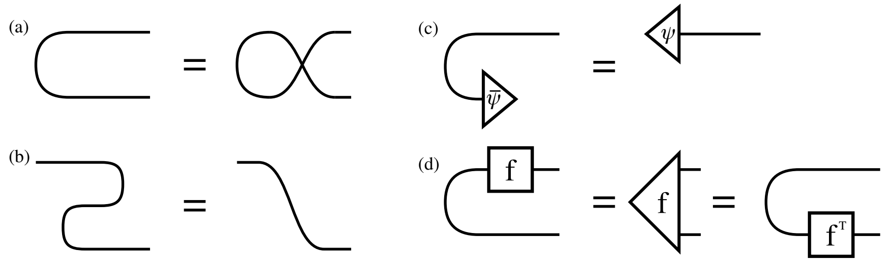


FIG. 17. Cup identities. (a) Symmetry. (b) Conjugate state. (c) Teleportation [61] or the *snake equation*. (d) Sliding an operator around a cup transposes it.

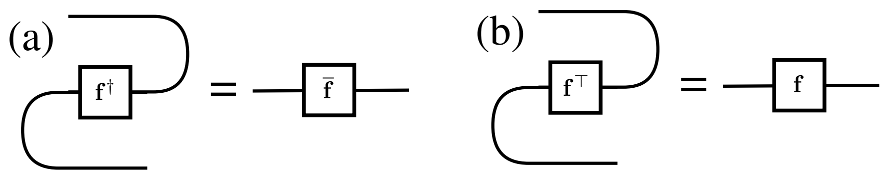


FIG. 18. Diagrammatic adjoints. Cups and caps allow us to take the transpose of a linear map. Note that care must be taken, as flipping a ket $|\psi\rangle$ to a bra $\langle\psi|$ is conjugate transpose, and bending a wire is simply transposition, so the conjugate must be taken: e.g. acting on $|\psi\rangle$ with a cap given as $\sum_i \langle ii|$ results in $\langle\overline{\psi}|$.

V. EXAMPLES OF CATEGORICAL TENSOR NETWORK STATES

Our categorical approach enables one to translate a quantum state directly into a new type of network: a so-called CTNS. We have focused on Boolean network components and have already presented in detail their algebraic properties and defining characteristics. Here we will illustrate their expressive power by considering a few elementary examples before presenting our main theorem (23), precisely showing how to determine a categorical tensor network to represent any given quantum state.

A. Constructing Boolean states

Since the fixed building blocks of our tensor networks are the logic tensors AND, OR, XOR and COPY, along with ancilla bits, we can immediately apply the universality of these elements for classical circuit construction to guarantee that any Boolean state has a categorical tensor network decomposition. However our construction goes beyond this because as we have seen, categorical tensor networks can be deformed and rewired in ways which are not ordinarily permitted in the standard acyclic-temporal definition of classical circuits. The W-state will be shown to provide a non-trivial example of this.

Example 16 (Functions on W- and GHZ-states)**.** We consider the function f_{W} which outputs logical-one given input bit string 001, 010 and 100 and logical-zero otherwise. Likewise the function f_{GHZ} is defined to output logical-one on input bit strings 000 and 111 and logical-zero otherwise. See Examples 19 and 20 which consider representation of these functions as polynomials. We will continue to work with a linear representation of functions on quantum states; here bit string $000 \mapsto |000\rangle$ (etc.).

Example 17 (MPS form for W-state)**.** Like the GHZ state, the W-state has a simple MPS representation

$$|\mathsf{W}_n\rangle = \langle 0| \begin{pmatrix} |0\rangle & 0 \\ |1\rangle & |0\rangle \end{pmatrix}^n |1\rangle = |10...0\rangle + |01...0\rangle + ... + |00...1\rangle. \tag{21}$$

This description (21) is succinct. All MPS-states have essentially this same topological or network structure. In contrast, our categorical construction described below breaks this network up further.

Remark 18 (Exact-value functions)**.** *The function f_{W} takes value logical-one on input vectors with k ones for a fixed integer k. Such functions are known in the literature as Exact-value symmetric Boolean functions. When cast into our framework, exact-value functions give rise to tensor networks which represent what are known as Dicke states [38].*

Example 19 (Function realization of f_{W} and f_{GHZ}: the Boolean case)**.** One can express (using $\overline{x}$ to mean Boolean variable negation)

$$f_{\mathsf{W}}(x_1, x_2, x_3) = \overline{x}_1\overline{x}_2x_3 \oplus x_1\overline{x}_2\overline{x}_3 \oplus \overline{x}_1x_2\overline{x}_3 \tag{22}$$

by noting that each term in the disjunctive normal form of f_{W} are disjoint, and hence OR maps to XOR as $\vee \mapsto \oplus$. The algebraic normal form (see Appendix B) becomes

$$f_{\mathsf{W}}(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 \oplus x_1x_2x_3 \tag{23}$$

$$f_{\mathsf{GHZ}}(x_1, x_2, x_3) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \tag{24}$$

Example 20 (Function realization of f_{W} and f_{GHZ}: the set function case)**.** Set functions are mappings from the family of subsets of a finite ground set (e.g. Booleans) to the real or complex numbers. In the circuit theory literature, functions from the Booleans to the reals are known as pseudo-Boolean functions and more commonly as multi-linear polynomials or forms (see [75] where these functions are used to embed a co-algebraic theory of logic gates in the ground state energy configuration of spin models). There exists an algebraic normal form and hence a unique multi-linear polynomial representation for each pseudo-Boolean function (see Appendix B). This is found by mapping the negated Boolean variable as $\overline{x} \mapsto (1 - x)$. For the GHZ- and W-functions defined in Example 16 we arrive at the unique polynomials (25) and (26).

$$f_{\mathsf{GHZ}}(x_1, x_2, x_3) = 1 - x_1 - x_2 + x_1x_2 - x_3 + x_1x_3 + x_2x_3 \tag{25}$$

$$f_{\mathsf{W}}(x_1, x_2, x_3) = x_1 + x_2 + x_3 - 2x_1x_2 - 2x_1x_3 - 2x_2x_3 + 3x_1x_2x_3 \tag{26}$$

These polynomials (25) and (26) are readily translated into categorical tensor networks.

Example 21 (Network realisation of W- and GHZ-states)**.** A network realization of W- and GHZ-states in our framework then follows by post-selecting the relevant network to $|1\rangle$ on the output bit — leaving the input qubits to represent a W- or GHZ-state respectively. An example of this is shown in Figure 19.

Two different categorical constructions for the building blocks of the W-state are shown in Figure 20 and Figure 21. Notice that in Figure 21 the resulting tensor network forms an atemporal classical circuit and is much more efficient than the naïve construction in Figure 20. Moreover by appropriately daisy-chaining the networks in Figure 21 we construct a categorical tensor network for an n-party W-state as shown in Figure 22. The resulting form of this tensor network is entirely equivalent (up to regauging) to the MPS description given earlier, but now reveals internal structure of the state in terms of CTNS building blocks.

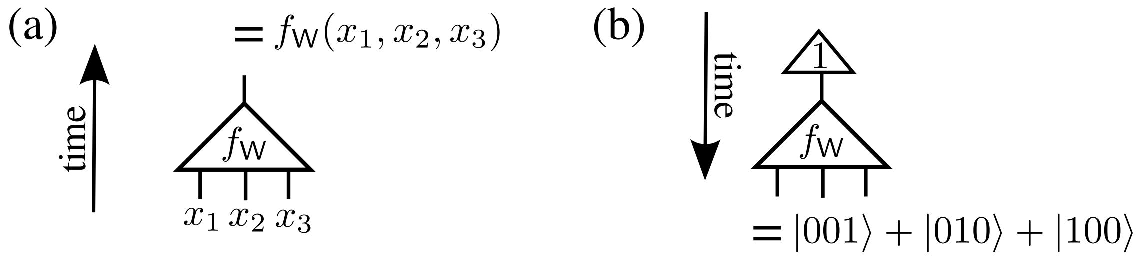


FIG. 19. Left (a) the circuit realization (internal to the triangle) of the function f_{W} of e.g. (23) which outputs logical-one given input $|x_1x_2x_3\rangle = |001\rangle$, $|010\rangle$ and $|100\rangle$ and logical-zero otherwise. Right (b) reversing time and setting the output to $|1\rangle$ (e.g. post-selection) gives a network representing the W-state. The naïve realization of f_{W} is given in Figure 21 with an optimized co-algebraic construction shown in Figure 21.

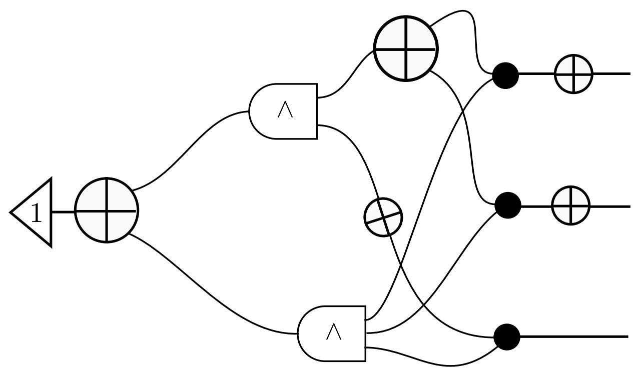


FIG. 20. Naïve CTNS realization of the familiar W-state $|001\rangle + |010\rangle + |100\rangle$. A standard (temporal) acyclic classical circuit decomposition in terms of the XOR-algebra realizes the function f_{W} of three bits. This function is given a representation on tensors. As illustrated, the networks input is post selected to $|1\rangle$ to realize the desired W-state.

B. Describing states with complex coefficients

Boolean states, such as the GHZ- and W-states, are typified by being superpositions of computational basis states with equal real coefficients (in both cases, these coefficients take only binary values, 0 and 1). In this section, we will permit a minor extension to binary superposition input/output states by considering arbitrary rank-1 tensors within our otherwise Boolean tensor networks. This is illustrated by a simple example:

Example 22 (Network realization of $|\psi\rangle = |01\rangle + |10\rangle + \alpha_k|11\rangle$)**.** We will now design a network to realize the state $|01\rangle + |10\rangle + \alpha_k|11\rangle$. The first step is to write down a function f_S such that

$$f_S(0,1) = f_S(1,0) = f_S(1,1) = 1 \tag{27}$$

and $f_S(00) = 0$ (in the present case, f_S is the logical OR-gate). We post select the network output on $|1\rangle$, which yields the state $|01\rangle + |10\rangle + |11\rangle$, see Figure 23(a). The next step is to realize a diagonal operator, that acts as identity on all inputs, except $|11\rangle$ which gets sent to $\alpha_k|11\rangle$. To do this, we design a function f_d such that

$$f_d(0,1) = f_d(1,0) = f_d(0,0) = 0 \tag{28}$$

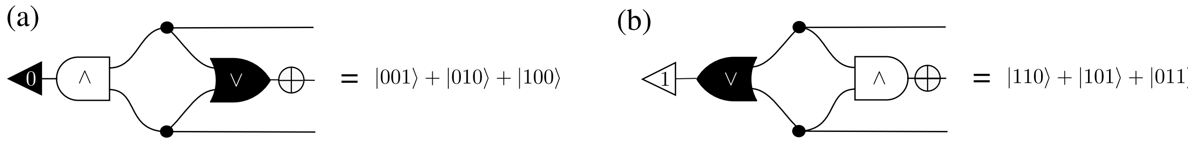


FIG. 21. W-class states in the categorical tensor network state formalism. (a) is the standard W-state. (b) is found from applying De Morgan's law (see Section III D) to (a) and rearranging after inserting inverters on the output legs. Notice the atemporal nature of the circuits, as one gate is used forwards, and the other backwards.

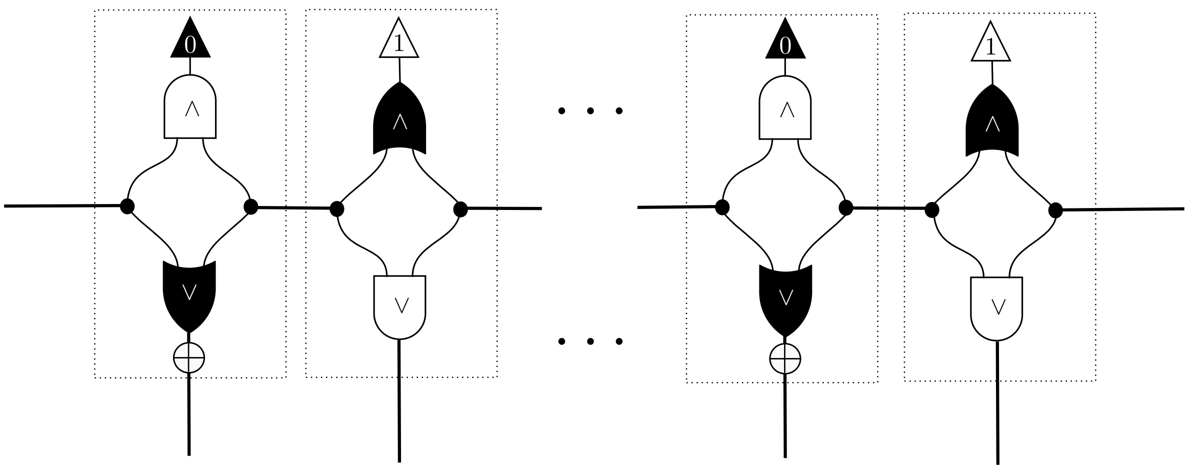


FIG. 22. W-state (n-party) in the categorical tensor network state formalism. The comb-like feature of efficient network contraction remains, with the internal structure of the network components exposed in terms of well understood algebraic structures.

and $f_d(1,1) = 1$ (in the present case, f_d is the logical AND-gate). This diagonal, takes the form in Figure 23(b). The final state $|\psi\rangle = |01\rangle + |10\rangle + \alpha_k|11\rangle$ is realized by connecting both networks, leading to Figure 23(c).

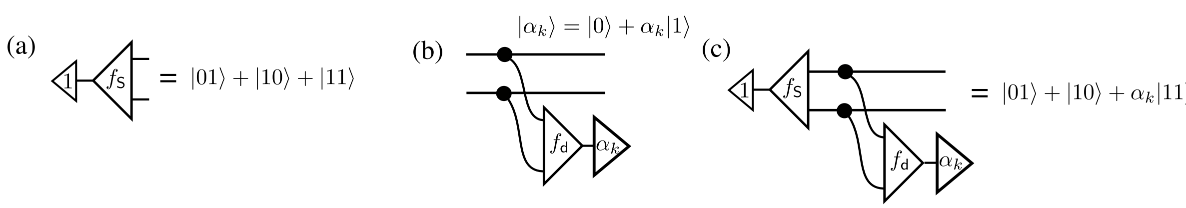


FIG. 23. Categorical tensor network representing state $|\psi\rangle = |01\rangle + |10\rangle + \alpha_k|11\rangle$, as explained in Example 22.

VI. PROOF OF THE MAIN THEOREMS

We are now in a position to state the main theorem of this work. Specifically, we have a constructive method to realize any quantum state in terms of a categorical tensor network. (A corollary of our exhaustive factorization of quantum states into tensor networks is a new type of quantum network universality proof. To avoid confusion, we point out that past universality proofs in the gate model already imply that the linear fragment (Figure 3) together with local gates is quantum universal. However, the known universality results clearly do not provide a method to factor a state into a tensor network! Indeed, the decomposition or factorization of a state into a tensor network is an entirely different problem which we address here.) We state and prove the theorem for the case of qubit. The higher dimensional case of qudits follows from known results that any d-state switching function can be expressed as a polynomial and realized as a connected network [43, 76, 77]. The theorem can be stated as

Theorem 23 (Tensor network representation of quantum states)**.** *For any state $|\psi\rangle$ of n-qubits with the form*

$$|\psi\rangle = \sum_{j=1}^{k} \alpha_j |\phi_j\rangle, \tag{29}$$

where α_j are complex coefficients and for each j the state $|\phi_j\rangle$ is an equal superposition of a set of computational basis states, it can be represented as a network containing tensors from the quantum Boolean calculus (Figures 3 and 4), together with input/ouput states of the form $|\alpha_j\rangle := |0\rangle + \alpha_j|1\rangle$.

Notice that an arbitrary state can be brought into the form required of $|\psi\rangle$ by composing it as $k = 2^n$ terms with each state $|\phi_j\rangle$ being a single distinct computational basis state. The proof is simplified by invoking some supporting lemmas.

Lemma 24. *There exists a map g represented by a tensor network taking diagonal maps in $\bigotimes_n \mathbb{C}^2 \to \bigotimes_n \mathbb{C}^2$ onto quantum states in $\bigotimes_n \mathbb{C}^2$.*

Proof. Let $\mathcal{D}$ be a diagonal map in $\bigotimes_n \mathbb{C}^2 \to \bigotimes_n \mathbb{C}^2$. We write $\mathcal{D} = \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{x}|$ and proceed as follows (where the term $\mathcal{D} \circ \bigotimes_n (|0\rangle + |1\rangle)$ immediately yields the desirable tensor network depiction)

$$\begin{aligned} g\{\mathcal{D}\} &:= \mathcal{D} \circ \bigotimes_n (|0\rangle + |1\rangle) = \mathcal{D} \circ \sum_{\mathbf{y}\in\{0,1\}^n} |\mathbf{y}\rangle = \sum_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{x}|\mathbf{y}\rangle = \\ &= \sum_{\mathbf{x}\in\{0,1\}^n} \delta_{\mathbf{x}\mathbf{y}} \alpha_{\mathbf{x}} |\mathbf{x}\rangle = \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle \end{aligned}$$

□

Lemma 25. *There exists a map h represented by a tensor network taking quantum states in $\bigotimes_n \mathbb{C}^2$ onto diagonal maps in $\bigotimes_n \mathbb{C}^2 \to \bigotimes_n \mathbb{C}^2$.*

Proof. Let $|\psi\rangle$ be a quantum state in $\bigotimes_n \mathbb{C}^2$. Let $\mathcal{D}$ be a diagonal map in $\bigotimes_n \mathbb{C}^2 \to \bigotimes_n \mathbb{C}^2$. We write $\mathcal{D} = \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{x}|$ and proceed as follows (where the term $\bigotimes_n (\sum_{i=0/1} |ii\rangle\langle i|) \circ \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$ immediately yields the desirable tensor network depiction)

$$\begin{aligned} h'\{|\psi\rangle\} &:= \bigotimes_n (\sum_{i=0/1} |ii\rangle\langle i|) \circ \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle = \sum_{\mathbf{y}\in\{0,1\}^n} |\mathbf{y}\mathbf{y}\rangle\langle\mathbf{y}| \circ \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle = \\ &= \sum_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} |\mathbf{y}\mathbf{y}\rangle \alpha_{\mathbf{x}} \delta_{\mathbf{x}\mathbf{y}} = \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_x |\mathbf{x}\mathbf{x}\rangle \end{aligned}$$

and then we now write h in terms of h'

$$h\{|\psi\rangle\} := \bigotimes_n (\sum_{i=0/1} \langle ii|) \circ h'\{\mathcal{D}\} = \sum_{\mathbf{y}\in\{0,1\}^n} \langle \mathbf{yy}| \circ \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{xx}\rangle =$$
$$= \sum_{\mathbf{y},\ \mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{y}|\delta_{\mathbf{yx}} = \sum_{\mathbf{x}\in\{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{x}|$$

□

Corollary 26. *It follows that $g\{h\{|\psi\rangle\}\} = \mathbf{1}_\psi \circ |\psi\rangle = |\psi\rangle$ and $h\{g\{\mathcal{D}\}\} = \mathbf{1}_\mathcal{D} \circ \mathcal{D} = \mathcal{D}$ and hence we have inverses for g and h establishing an isomorphism between diagonal operators in $\bigotimes_n \mathbb{C}^2 \to \bigotimes_n \mathbb{C}^2$ and quantum states in $\bigotimes_n \mathbb{C}^2$.*

With the supporting lemmas in place, we will now proceed to prove Theorem 23.

Proof. Returning to our particular expression for an arbitrary quantum state (note that in Equation (30) we now append an extra term, by letting the sum run from $j = 0$ instead of $j = 1$, the use of this will become clear below)

$$|\psi\rangle = \sum_{j=0}^{k} \alpha_j |\phi_j\rangle, \tag{30}$$

where α_j are complex coefficients and for each j the state $|\phi_j\rangle$ is an equal superposition of a set of computational basis states, we will explain how $k+1$ asynchronous circuits [41] are used to factor the state, and express the state as a CTNS (here and in what follows k is the highest term in the sum from Equation (29)).

We proceed by returning to our original expression (29) from Theorem 23 (starting from $j = 1$) with the coefficients removed

$$|\psi'\rangle = \sum_{j=1}^{k} |\phi_j\rangle \tag{31}$$

Each individual term in Equation (31) is then expressed in the computational basis and used to form a set denoted $\mathcal{L}^+$. All corresponding bit patterns of the same dimension not appearing in this expression form a second set $\mathcal{L}^-$ (where clearly $\mathcal{L}^+ \cap \mathcal{L}^- = \emptyset$ and $\mathcal{L}^+ \cup \mathcal{L}^- = \{0,1\}^n$, where n is the number of qubits in the desired state). We proceed to construct a function f_0 that outputs logical-one on all input bit strings in $\mathcal{L}^+$ and outputs local zero on all input bit strings in $\mathcal{L}^-$. The function acts on $n+1$ bits, the inputs are given on the right of the tensor symbol and the output on the left of the tensor symbol in (32)

$$|\psi_f\rangle = \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle \otimes |f(\mathbf{x})\rangle \tag{32}$$

where $f(\mathbf{x}) : \mathbb{B}^n \to \mathbb{B} :: \mathbf{x} \mapsto f(\mathbf{x})$ was given the representation on quantum states in Section V. Post selecting the networks output (the rightmost bit in Equation (32)) $|1\rangle$ realizes the desired superposition of terms, with all coefficients of the terms and hence relative phases equal.

$$|\psi_f'\rangle = \sum_{\mathbf{x}\in\{0,1\}^n} \langle 1|f(\mathbf{x})\rangle |\mathbf{x}\rangle \tag{33}$$

For our specific construction, depicted in Figure 24, we proceed by inverting the output of the function (e.g. $f_0 \mapsto f_0 \oplus 1$). We then post select the output of the function to the state $|\alpha_0\rangle = |0\rangle + \alpha_0|1\rangle = |0\rangle$ for the choice $\alpha_0 = 0$. This handles the $j = 0$ term in the sum Equation (30).

To adjust the amplitudes of the desired state from Equation (30), we will construct tensors that represent diagonal operators. For the jth term in $|\psi\rangle$ with coefficient α_j, we again construct a function f_j. We represent $|\phi_j\rangle$ in the computational basis, and each term in this expression is used to form a set denoted $\mathcal{L}^+$. All corresponding bit patterns of the same dimension not appearing in this expression form a second set $\mathcal{L}^-$. We proceed to construct f_j to output logical-one on all input bit strings in $\mathcal{L}^+$ and outputs logical-zero on all input bit strings in $\mathcal{L}^-$. The network is then post selected to $|0\rangle + \alpha_j|1\rangle$ which results in states of the form

$$|\psi_D\rangle = \sum_{\mathbf{x}\in\{0,1\}^n} \langle 0|f(\mathbf{x})\rangle|\mathbf{x}\rangle + \alpha_j \sum_{\mathbf{x}\in\{0,1\}^n} \langle 1|f(\mathbf{x})\rangle|\mathbf{x}\rangle \tag{34}$$

and we transform f_j into a diagonal operator having entries $\in \{1, \alpha_j\}$ by applying the map h from Lemma 25 resulting in the diagonal map

$$\mathcal{D}_j = \sum_{\mathbf{x}\in\{0,1\}^n} \langle 0|f(\mathbf{x})\rangle|\mathbf{x}\rangle\langle\mathbf{x}| + \alpha_j \sum_{\mathbf{x}\in\{0,1\}^n} \langle 1|f(\mathbf{x})\rangle|\mathbf{x}\rangle\langle\mathbf{x}| \tag{35}$$

We will apply k such commuting maps $\mathcal{D}_j$ to the initial state, accounting for k asynchronous circuits. The operators are composed by means of n co-COPY-dots from Section III E (see Figure 23 and Example 22). There will be a single output with open legs which gives the state. Each of the n COPY-dots will then require $k+2$ legs. The resulting construction then gives tensor networks with the form shown in Figure 24. $\square$

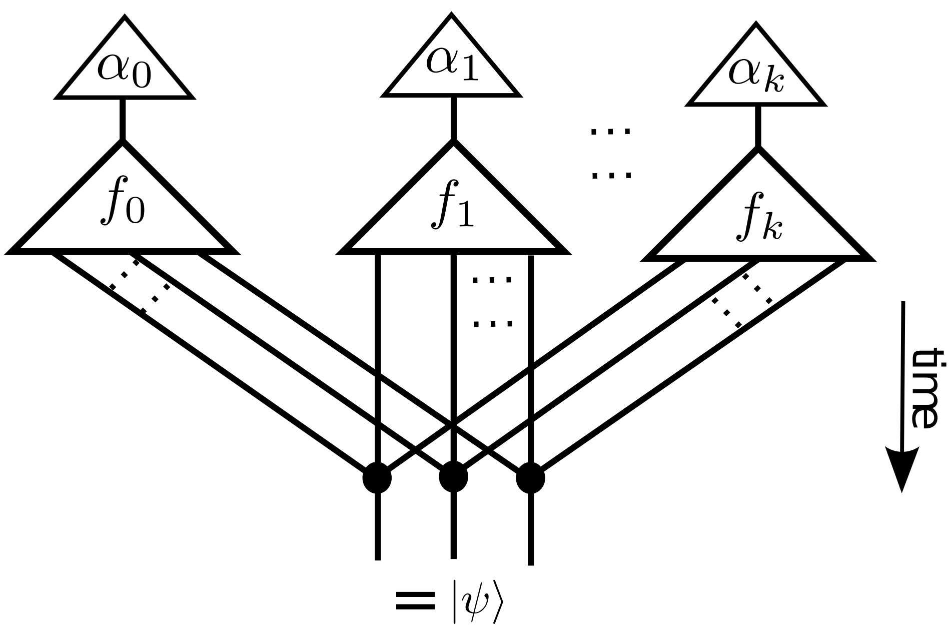


FIG. 24. The CTNS for a state $|\psi\rangle$ resulting from our exhaustive construction procedure.

Remark 27 (Qudit states)**.** *In Theorem 23 we considered the arbitrary states of n-qubits. By using multi-valued logic (also called d-state switching, or many-valued logic), it is possible to define a universal gate set similar to what was done for the case of qubits and so equivalently construct a CTNS for n-body qudit systems [43, 76, 77].*

Definition 28 (Generalized Polynomial Boolean States (GPBS))**.** The size of a tensor network is the number of tensors it contains and its depth is the maximal length of a path from any tensor to any other. Consider families of uniform circuits, built from the XOR-algebra, that is the bounded fan-in gates AND, XOR, COPY, of arity two and the constant $|1\rangle$. We will index these circuit families by bounding the circuit depth, which also has the impact of bounding the maximum fan-in and the circuit size. We will then consider categorical tensor networks to represent states $|\psi\rangle$ of n interacting d-level systems constructed from bounded circuit families to realize each f_j with the form given in Figure 24. We will proceed by indexing these families of categorical tensor networks in terms of k, the maximum depth of any given circuit realizing any function f_j in the network. We will then bound the number of such functions f_j to be at most some polynomial in k. We then determine how $k(n)$ changes. This works by considering circuit families and categorical tensor networks of a given form, used to represent quantum states on increasingly many subsystems n. If $k(n)$ is bounded by a polynomial in n the categorical tensor network has an efficient description. We index such families as $\mathsf{C}(k)$, and refer to them as Generalized Polynomial Boolean States (GPBS).

Theorem 29. GPBS *from Definition 28 are sampled exactly in the computational basis in time and space complexity bounded by* $\mathrm{poly}(k)$*.*

Proof. To prove Theorem 29 we begin first by considering a qudit state vector $|x_0, x_1, ..., x_n\rangle$ for specific $x_0, x_1, ..., x_n \in \{0, 1, ..., d-1\}$. We wish to know the coefficient $\langle x_0, x_1, ..., x_n|\mathsf{C}\rangle$, where C is a CTNS representing a GPBS. The COPY-gates in the construction from Figure 24 map

$$|x_0, x_1, ..., x_n\rangle \mapsto \bigotimes_{\mathrm{poly}(k)} |x_0, x_1, ..., x_n\rangle \tag{36}$$

each of these $\mathrm{poly}(k)$ vectors will be acted on by a network realizing f_j post selected to the state $|\alpha_j\rangle$. Hence, to sample the network C amounts evaluating the sum

$$\langle x_0, x_1, ..., x_n|\mathsf{C}\rangle = \sum_{j\in\mathrm{poly}(k)} \langle f_j(x_0, x_1, ..., x_n)|\alpha_j\rangle = c \in \mathbb{C} \tag{37}$$

The proof then follows by simply evaluating each of the $\mathrm{poly}(k)$ $\mathrm{poly}(k)$-depth functions $f_j(x_0, x_1, ..., x_n)$ and then summing the inner products $\langle f_j(x_0, x_1, ..., x_n)|\alpha_j\rangle$. □

We note that the construction in Theorem 23 automatically groups basis states with the same coefficients α_j, of the k terms. Further reductions are also possible if say a given set of coefficients are given by products of other coefficients. While this construction does prove the existence of a CTNS (along with how to build it) our construction will not render efficient representations for general cases, as one might expect. Indeed, there is no guarantee that any of the $k+1$ switching functions are efficient in their complexity, nor that the resulting complete network is contractible. The latter property is in fact confounded by the presence of fan-in (up to $k+2$ legs) of the n co-COPY-dots (the presence also implies that the network cannot represent a deterministic physical preparation procedure [66]). However, as we saw earlier with string-bond states in Figure 2, the COPY dot breaks up when computational basis states are inputted. For our general decomposition in in Figure 24, this case causes the $k+1$ Boolean switching functions to similarly decouple. Intuitively, if we further restrict ourselves to $k+1$ being polynomial in n, and additionally that each switching function has a depth which is also polynomial in n, then the subsequent evaluation of the amplitude of the state is efficient for any computational basis state (see Definition 28 and Theorem 29). This is a weak requirement in practice and interestingly, does not depend on the internal geometry of the networks representing the functions, but only on their depth and size. Thus we have found a new general class of states which can be sampled exactly and efficiently. Finally the construction was based on using acyclic-temporal Boolean circuits. However, we have already seen that in the tensor context wires can be bent around: it is not necessary for a tensor network to correspond to a valid classical circuit. As the W-state example (see Figures 21 and 22) illustrated the tensor networks can be much simpler once this new freedom is exploited.

VII. OUTLOOK AND CONCLUDING REMARKS

We have introduced methods from category theory and algebra to tensor network states. Our main focus has been on tensor networks built from contracting Boolean tensors and we have outlined their algebraic properties. The logical conclusion of this approach has led us to a exhaustive CTNS decomposition of quantum states. From this we obtained a class of quantum states, which can be sampled in the computational basis efficiently and exactly. The expressiveness and power of this new method was further illustrated by considering several simple test cases: we considered internal structure of some MPS states, e.g. GHZ- and W-states. We have opened up some future potential research directions. In particular, beyond the form of our construction there is an open question as to whether the algebraic properties of some subset of the tensors in CTNS can enable efficiently contractible networks beyond those already known which are based on topology (like MPS) and additional unitary/isometric constraints (MERA). In this way future studies of CTNS may lead to new classes of states and algorithms which will help challenge and shape our understanding of many-body physics.

ACKNOWLEDGMENTS

We thank John Baez, Tomi Johnson, Martin Plenio, Vlatko Vedral, Mike Shulman, Ville Bergholm, Samson Abramsky, Bob Coecke, Chris Heunen and Guifré Vidal. JDB received support from the EPSRC and completed large parts of this work visiting the Centre for Quantum Technologies, at the National University of Singapore (these visits were hosted by Vlatko Vedral). SRC and DJ thank the National Research Foundation and the Ministry of Education of Singapore for support. DJ acknowledges support from the ESF program EuroQUAM (EPSRC grant EP/E041612/1), the EPSRC (UK) through the QIP IRC (GR/S82176/01), and the European Commission under the Marie Curie programme through QIPEST. Earlier versions of this work circulated for just over one year prior to being uploaded to the arXiv, the most widespread appearing under the title *Algebra and co-algebra on Categorical Tensor Network States.*

Appendix A: Algebra on quantum states

We are concerned with a network theory of quantum states. This on the one hand can be used as a tool to solve problems about states and operators in quantum theory, but does have a physical interpretation on the other. This is not foundational *per se* but instead largely based on what one might call an operational interpretation of quantum states and processes. We call an algebra a pairing on a vector space, taking two vectors and producing a third (you might instead call it a monoid if there is a unit, and then a group if the set of considered vectors is closed under the product). Let's now examine how every tripartite quantum state forms an algebra [13].

Consider a tripartite quantum state (subsystems labeled 1,2 and 3), and then ask the question: "how would the state of the third system change after measurement of systems one and two?" Enter Algebras: as stated, an algebra on a vector space, or on a Hilbert space is formed by a *product* taking two elements from the vector space to produce a third element in the vector space. Algebra on states can then be studied by considering duality of the state, that is considering the adjunction between the maps of type

$$\mathbf{1} \to \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \qquad \text{and} \qquad \overline{\mathcal{H}} \otimes \overline{\mathcal{H}} \to \mathcal{H} \tag{A1}$$

This duality is made evident by using the †-compact structure of the category (e.g. the cups and caps). It is given vivid physical meaning by considering the effect measuring (that is two events) two components of a state has on the third component.

Remark 30 (Overbar notation on Spaces)**.** *Given a Hilbert space $\mathcal{H}$, we can consider the Hilbert space $\overline{\mathcal{H}}$ which can be simply thought of as the Hilbert space $\mathcal{H}$ will all basis vectors complex conjugates (overbar). That is, $\overline{\mathcal{H}}$ is a vector space whose elements are in one-to-one correspondence with the elements of $\mathcal{H}$:*

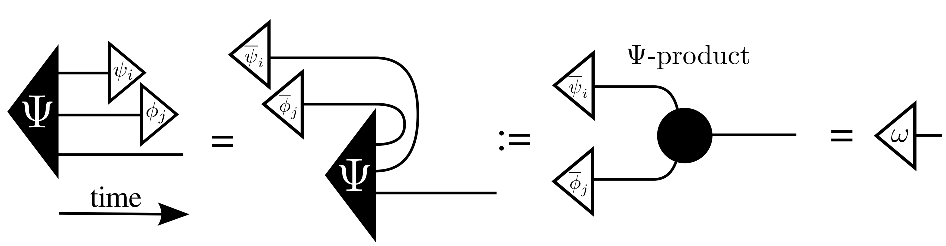
$$\overline{\mathcal{H}} = \{\overline{v} \mid v \in \mathcal{H}\}, \tag{A2}$$

with the following rules for addition and scalar multiplication:

$$\overline{v} + \overline{w} = \overline{v + w} \quad \textit{and} \quad \alpha\,\overline{v} = \overline{\overline{\alpha}\,v}\,. \tag{A3}$$

Remark 31 (Definition of Algebra)**.** *We consider an algebra as a vector space $\mathcal{A}$ endowed with a product, taking a pair of elements (e.g. from $\mathcal{A} \otimes \mathcal{A}$) and producing an element in $\mathcal{A}$. So the product is a map $\mathcal{A} \otimes \mathcal{A} \to \mathcal{A}$, which may not be associative or have a unit (that is, a multiplicative identity — see Example 8 for an example of an algebra on a quantum state without a unit).*

Observation 32 (Every tripartite Quantum State Forms an Algebra)**.** *Let $|\psi\rangle \in \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H}$ be a quantum state and let M_i, M_j be complete sets of measurement operators. Then $(|\psi\rangle, M_i, M_j)$ forms an algebra.*



The quantum state $|\Psi\rangle = \sum_{ijk} \psi^{ijk}|ijk\rangle$ is drawn as a triangle, with the identity operator on each subsystem acting as time goes to the right on the page (represented as a wire). Projective measurements with respect to M_i and M_j are made. We define these complete measurement operators as

$$M_1 = \sum_{i=1}^{N} i|\psi_i\rangle\langle\psi_i| \tag{A4}$$

$$M_2 = \sum_{j=1}^{N} j|\phi_j\rangle\langle\phi_j| \tag{A5}$$

such that we recover the identity operator on the N-level subsystem viz

$$\sum_{j=1}^{N} |\phi_j\rangle\langle\phi_j| = \sum_{i=1}^{N} |\psi_i\rangle\langle\psi_i| = \mathbf{1}_N \tag{A6}$$

The measurements result in eigenvalues i, j leaving the state of the unmeasured system in

$$|\omega\rangle = \sum_{xyz} \psi^{xyz}\langle\overline{\psi}^x|x\rangle\langle\overline{\phi}^y|y\rangle|z\rangle \tag{A7}$$

where $\langle\overline{Q}| \stackrel{\text{def}}{=} |Q\rangle^\top$ that is, the transpose is factored into: (i) taking the dagger (diagrammatically this mirrors states across the page) and (ii) taking the complex conjugate. Hence,

$$|\overline{Q}\rangle^\dagger = |Q\rangle^\top = \langle\overline{Q}| = \overline{|Q\rangle^\dagger} \tag{A8}$$

and if we pick a real valued basis for $|x\rangle, |y\rangle, |z\rangle = |0\rangle, |1\rangle$ we recover

$$|\omega\rangle = \sum_{xyz} \psi^{xyz}\langle x|\psi_x\rangle\langle y|\phi_y\rangle|z\rangle \tag{A9}$$

As stated, this physical interpretation is not our main interest. Even in its absence, we're able to write down and represent a quantum state purely in terms of a connected network, where each component is fully defined in terms of algebraic laws.

Appendix B: XOR-algebra

Here we review the concept of an algebraic normal form (ANF) for Boolean polynomials, commonly known as PPRMs. See the reference book [63] and the historical references [64, 65] for further details.

Definition 33. The XOR-algebra forms a commutative ring with presentation $M = \{\mathbb{B}, \wedge, \oplus\}$ where the following product is called XOR

$$-\oplus- : \mathbb{B} \times \mathbb{B} \mapsto \mathbb{B} :: (a, b) \to a + b - ab \bmod 2 \tag{B1}$$

and conjunction is given as

$$— \wedge — : \mathbb{B} \times \mathbb{B} \mapsto \mathbb{B} :: (a, b) \to a \cdot b, \tag{B2}$$

where $a \cdot b$ is regular multiplication over the reals. One defines left negation $\neg(—)$ in terms of $\oplus$ as $\neg(—) \equiv$

$$1 \oplus (—) : \mathbb{B} \mapsto \mathbb{B} :: a \to 1 - a. \tag{B3}$$

In the XOR-algebra, 1-5 hold. (i) $a \oplus 0 = a$, (ii) $a \oplus 1 = \neg a$, (iii) $a \oplus a = 0$, (iv) $a \oplus \neg a = 1$ and (v) $a \vee b = a \oplus b \oplus (a \wedge b)$. Hence, 0 is the unit of XOR and 1 is the unit of AND. The 5th rule reduces to $a \vee b = a \oplus b$ whenever $a \wedge b = 0$, which is the case for disjoint (mod 2) sums. The truth table for AND follows

x_1	x_2	$f(x_1, x_2) = x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Definition 34. Any Boolean equation may be uniquely expanded to the fixed polarity Reed-Muller form as:

$$\begin{aligned} f(x_1, x_2, ..., x_k) = {} & c_0 \oplus c_1 x_1^{\sigma_1} \oplus c_2 x_2^{\sigma_2} \oplus \cdots \oplus c_n x_n^{\sigma_n} \oplus \\ & c_{n+1} x_1^{\sigma_1} x_n^{\sigma_n} \oplus \cdots \oplus c_{2k-1} x_1^{\sigma_1} x_2^{\sigma_2}, ..., x_k^{\sigma_k}, \end{aligned} \tag{B4}$$

where selection variable $\sigma_i \in \{0, 1\}$, literal $x_i^{\sigma_i}$ represents a variable or its negation and any c term labeled c_0 through c_j is a binary constant 0 or 1. In Equation (B4) only fixed polarity variables appear such that each is in either un-complemented or complemented form.

Let us now consider derivation of the form from Definition 34. Because of the structure of the algebra, without loss of generality, one avoids keeping track of indices in the N node case, by considering the case where $N \equiv 2^n = 8$.

Example 35. The vector $\underline{c} = (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7,)^\intercal$ represents all possible outputs of any function $f(x_1, x_2, x_3)$ over the algebra formed from linear extension of $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$. We wish to construct a normal form in terms of the vector $\underline{c}$, where each $c_i \in \{0, 1\}$, and therefore $\underline{c}$ is a selection vector that simply represents the output of the function $f : \mathbb{B} \times \mathbb{B} \times \mathbb{B} \to \mathbb{B} :: (x_1, x_2, x_3) \mapsto f(x_1, x_2, x_3)$. One may expand f as:

$$\begin{aligned} f(x_1, x_2, x_3) = {} & (c_0 \cdot \neg x_1 \cdot \neg x_2 \cdot \neg x_3) \vee (c_1 \cdot \neg x_1 \cdot \neg x_2 \cdot x_3) \vee (c_2 \cdot \neg x_1 \cdot x_2 \cdot \neg x_3) \\ & \vee (c_3 \cdot \neg x_1 \cdot x_2 \cdot x_3) \vee (c_4 \cdot x_1 \cdot \neg x_2 \cdot \neg x_3) \vee (c_5 \cdot x_1 \cdot \neg x_2 \cdot x_3) \\ & \vee (c_6 \cdot x_1 \cdot x_2 \cdot \neg x_3) \vee (c_7 \cdot x_1 \cdot x_2 \cdot x_3) \end{aligned} \tag{B5}$$

Since each disjunctive term is disjoint the logical OR operation may be replaced with the logical XOR operation. By making the substitution $\neg a = a \oplus 1$ for all variables and rearranging terms one arrives at the following normal form. (For instance, $\neg x_1 \cdot \neg x_2 \cdot \neg x_3 = (1 \oplus x_1) \cdot (1 \oplus x_2) \cdot (1 \oplus x_3) = (1 \oplus x_1 \oplus x_2 \oplus x_2 \cdot x_3) \cdot (1 \oplus x_3) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_1 \cdot x_3 \oplus x_2 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3$.)

$$f(x_1, x_2, x_3) = c_0 \oplus (c_0 \oplus c_4) \cdot x_1 \oplus (c_0 \oplus c_2) \cdot x_2 \oplus (c_0 \oplus c_1) \cdot x_3 \oplus (c_0 \oplus c_2 \oplus c_4 \oplus c_6) \cdot x_1 \cdot x_2$$
$$\oplus (c_0 \oplus c_1 \oplus c_4 \oplus c_5) \cdot x_1 \cdot x_3 \oplus (c_0 \oplus c_1 \oplus c_2 \oplus c_3) \cdot x_2 \cdot x_3$$
$$\oplus (c_0 \oplus c_1 \oplus c_2 \oplus c_3 \oplus c_4 \oplus c_5 \oplus c_6 \oplus c_7) \cdot x_1 \cdot x_2 \cdot x_3 \quad \text{(B6)}$$

The set of linearly independent vectors, $\{x_1, x_2, x_3, x_1 \cdot x_2, x_1 \cdot x_3, x_2 \cdot x_3, x_1 \cdot x_2 \cdot x_3\}$ combined with a set of scalars from Equation B6 spans the eight dimensional space of the Hypercube representing the Algebra. A similar form holds for arbitrary N.

$$f(x_1, x_2, x_3) = (a_1) \cdot x_1 \oplus (a_2) \cdot x_2 \oplus (x_3) \cdot x_3 \oplus (a_1 \oplus a_2 \oplus a_1 \oplus c_2) \cdot x_1 \cdot x_2$$
$$\oplus (a_1 \oplus a_3 \oplus a_1 \oplus c_3) \cdot x_1 \cdot x_3 \oplus (a_2 \oplus a_3 \oplus a_2 \oplus c_3) \cdot x_2 \cdot x_3$$
$$\oplus (a_1 \oplus a_2 \oplus a_3 \oplus a_1 \oplus a_2 \oplus a_3) \cdot x_1 \cdot x_2 \cdot x_3 \quad \text{(B7)}$$

---

[1] Saunders Mac Lane. Categories for the working mathematician 2nd ed. *Graduate Texts in Mathematics, Springer*, 1998.

[2] S. J. Denny, J. D. Biamonte, D. Jaksch, and S. R. Clark. Algebraically contractible topological tensor network states. *J. Phys. A: Math. Theor.*, 45:015309, 2012. arXiv:1108.0888.

[3] S. Al-Assam, S. R. Clark, C. J. Foot, and D. Jaksch. Capturing long range correlations in two-dimensional quantum lattice systems using correlator product states. *ArXiv e-prints*, July 2011.

[4] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992.

[5] U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77(1):259–315, Apr 2005.

[6] M. J. Hartmann, J. Prior, S. R. Clark, and M. B. Plenio. Density Matrix Renormalization Group in the Heisenberg Picture. *Physical Review Letters*, 102(5):057202, February 2009. arXiv:0808.0666.

[7] Stellan Östlund and Stefan Rommer. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.*, 75(19):3537–3540, Nov 1995.

[8] G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91:147902, 2003.

[9] G. Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93:040502, 2004.

[10] T. Barthel, C. Pineda, and J. Eisert. Contraction of fermionic operator circuits and the simulation of strongly correlated fermions. *Phys. Rev. A*, 80(4):042333, October 2009. arXiv:0907.3689.

[11] R. N. C. Pfeifer, P. Corboz, O. Buerschaper, M. Aguado, M. Troyer, and G. Vidal. Simulation of anyons with tensor network algorithms. *Phys. Rev. B*, 82(11):115126, September 2010. arXiv:1006.3532.

[12] R. König and E. Bilgin. Anyonic entanglement renormalization. *Phys. Rev. B*, 82(12):125118, September 2010. arXiv:1006.2478.

[13] Jacob Biamonte. Lectures on Tensor Network States, QIC 890/891 Selected Advanced Topics in Quantum Information. *IQC, The University of Waterloo, Waterloo Ontario, Canada (youtube lecture series)*, 2011. `http://www.qubit.org/iqc2011`.

[14] A. J. Daley, S. R. Clark, D. Jaksch, and P. Zoller. Numerical analysis of coherent many-body currents in a single atom transistor. *Phys. Rev. A*, 72(4):043618, October 2005.

[15] R. Steinigeweg, S. Langer, F. Heidrich-Meisner, I. P. McCulloch, and W. Brenig. Coherent spin-current oscillations in transverse magnetic fields. *ArXiv e-prints*, October 2010.

[16] S. R. Clark, J. Prior, M. J. Hartmann, D. Jaksch, and M. B. Plenio. Exact matrix product solutions in the Heisenberg picture of an open quantum spin chain. *New Journal of Physics*, 12(2):025005, February 2010. arXiv:0907.5582.

[17] Martin Bruderer, Alexander Klein, Stephen R. Clark, and Dieter Jaksch. Polaron physics in optical lattices. *Phys. Rev. A*, 76(1):011605, Jul 2007.

[18] M. Bruderer, T. H. Johnson, S. R. Clark, D. Jaksch, A. Posazhennikova, and W. Belzig. Phonon resonances in atomic currents through bose-fermi mixtures in optical lattices. *Phys. Rev. A*, 82(4):043617, Oct 2010.

[19] J Schachenmayer, G Pupillo, and A J Daley. Time-dependent currents of one-dimensional bosons in an optical lattice. *New Journal of Physics*, 12(2):025014, 2010.

[20] T. H. Johnson, S. R. Clark, and D. Jaksch. Dynamical simulations of classical stochastic systems using matrix product states. *Phys. Rev. E*, 82(3):036702, Sep 2010.

[21] F. Verstraete, V. Murg, and J. I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57:143–224, 2008.

[22] C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac. Fermionic projected entangled pair states. *Phys. Rev. A*, 81(5):052338, 2010. arXiv:0904.4667.

[23] P. Corboz, R. Orús, B. Bauer, and G. Vidal. Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states. *Phys. Rev. B*, 81(16):165104, 2010. arXiv:0912.0646.

[24] G. Vidal. Class of Quantum Many-Body States That Can Be Efficiently Simulated. *Physical Review Letters*, 101(11):110501, 2008.

[25] G. Vidal. Entanglement Renormalization: an introduction. *ArXiv e-prints*, 2009. `0912.1651`.

[26] P. Corboz and G. Vidal. Fermionic multiscale entanglement renormalization ansatz. *Phys. Rev. B*, 80(16):165129, October 2009. arXiv:0907.3184.

[27] P. Corboz, G. Evenbly, F. Verstraete, and G. Vidal. Simulation of interacting fermions with entanglement renormalization. *Phys. Rev. A*, 81(1):010303, January 2010. arXiv:0904.4151.

[28] J. C. Baez and A. Lauda. A Prehistory of n-Categorical Physics. *ArXiv e-prints*, 2009. `arXiv:0908.2469`.

[29] A. Joyal and R. Street. The geometry of tensor calculus i. *Advances in Mathematics*, 88(55), 1991.

[30] P. Selinger. A survey of graphical languages for monoidal categories. *ArXiv e-prints*, August 2009. arXiv:0908.3347.

[31] Yves Lafont. Penrose diagrams and 2-dimensional rewriting. In *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Note Series*, pages 191–201. Cambridge University Press, 1992.

[32] Yves Lafont. Equational reasoning with 2-dimensional diagrams. In *Term Rewriting*, volume 909 of *Lecture Notes in Computer Science*, pages 170–195. Springer-Verlag, 1995.

[33] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial Mathematics and*

its Applications, Academic Press, 1971.

[34] J. C. Baez and M. Stay. Physics, Topology, Logic and Computation Rosetta Stone. *ArXiv e-prints*, March 2009. arXiv:0903.0340.

[35] Yves Lafont. Towards an algebraic theory of boolean circuits. *Journal of Pure and Applied Algebra*, 184:2003, 2003.

[36] Y. Guiraud. The three dimensions of proofs. *ArXiv Mathematics e-prints*, December 2006. arXiv:math/0612089.

[37] Carolyn Brown and Graham Hutton. Categories, Allegories, and Circuit Design. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, Los Alamitos, California, July 1994.

[38] M. Aulbach, D. Markham, and M. Murao. The maximally entangled symmetric state in terms of the geometric measure. *New Journal of Physics*, 12(7):073025, July 2010. arXiv:1003.5643.

[39] J. I. Cirac and F. Verstraete. Renormalization and tensor product states in spin chains and lattices. *J. Phys. A Math. Gen.*, 42:4004, 2009.

[40] J. Smith and M. Mosca. Algorithms for Quantum Computers. *ArXiv e-prints*, January 2010. arXiv:1001.0767.

[41] I. Wegener. The complexity of boolean functions. *Wiley-Teubner*, 1987. online at: http://eccc.hpi-web.de/.

[42] E. Boros and P.L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[43] Grzegorz Malinowski. Many-valued logics. *Clarendon Press: Oxford University Press*, 1993. SERIES :Oxford logic guides.

[44] A. Kitaev, A. Shen, and M. Vyalyi. Classical and quantum computation. *AMS, Graduate Studies in Mathematics*, 47, 2002.

[45] Michael Nielsen and Isaac Chuang. Quantum computation and quantum information. *Cambridge University Press*, 2000.

[46] L. Amico, R. Fazio, A. Osterloh, and V. Vedral. Entanglement in many-body systems. *Reviews of Modern Physics*, 80:517–576, April 2008.

[47] R. B. Griffiths, S. Wu, L. Yu, and S. M. Cohen. Atemporal diagrams for quantum circuits. *Phys. Rev. A*, 73(5):052309, May 2006.

[48] D. Gross, J. Eisert, N. Schuch, and D. Perez-Garcia. Measurement-based quantum computation beyond the one-way model. *Phys. Rev. A*, 76(5):052315, November 2007.

[49] L. Lamata, J. León, D. Pérez-García, D. Salgado, and E. Solano. Sequential Implementation of Global Quantum Operations. *Physical Review Letters*, 101(18):180506, October 2008. arXiv:0711.3652.

[50] C. Schön, K. Hammerer, M. M. Wolf, J. I. Cirac, and E. Solano. Sequential generation of matrix-product states in cavity QED. *Phys. Rev. A*, 75(3):032311, March 2007. arXiv:quant-ph/0612101.

[51] S. Ostlund and S. Rommer. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.*, 75:3537, 1995.

[52] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Lett. Math. Phys.*, 25:249, 1992.

[53] F. Verstraete, V. Murg, and J. I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.

[54] G. Vidal. Entanglement renormalization. *Phys. Rev. Lett.*, 99:220405, 2007.

[55] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Phys. Rev. A*, 74(2):022320, Aug 2006.

[56] L. Tagliacozzo, G. Evenbly, and G. Vidal. Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law. *Phys. Rev. B*, 80(23):235127, December 2009.

[57] Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac. Simulation of quantum many-body systems with strings of operators and monte carlo tensor contractions. *Phys. Rev. Lett.*, 100(4):040501, Jan 2008.

[58] F. Mezzacapo, N. Schuch, M. Boninsegni, and J. I. Cirac. Ground-state properties of quantum many-body systems: entangled-plaquette states and variational Monte Carlo. *New Journal of Physics*, 11(8):083026, August 2009.

[59] H. J. Changlani, J. M. Kinder, C. J. Umrigar, and G. K.-L. Chan. Approximating strongly correlated wave functions with correlator product states. *Phys. Rev. B*, 80(24):245116, December 2009.

[60] John Baez et al. Bimonoids from biproducts. *The n-Category Cafe Blog.* online at: `http://golem.ph.utexas.edu/category/2010/09/bimonoids_from_biproducts.html`.

[61] Samson Abramsky and Bob Coecke. Categorical quantum mechanics. *Chapter in the Handbook of Quantum Logic and Quantum Structures vol II, Elsevier*, 2008.

[62] Peter Selinger. Dagger compact closed categories and completely positive maps: (extended abstract). *Electronic Notes in Theoretical Computer Science*, 170:139 – 163, 2007. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).

[63] *Discrete and switching functions.* McGraw-Hill Int. Book Co., 1978.

[64] M. Cohn. Inconsistent canonical forms of switching functions. *IRE Transactions of Electronic Computers*, 1962.

[65] A. Mukhopadhyay and G. Schmitz. Minimization of exclusive-or and logical-equivalence switching circuits. *IEEE Trans. on Computers*, 1970.

[66] V. Bergholm and J. D. Biamonte. Categorical quantum circuits. *Journal of Physics A Mathematical General*, 44(24):245304, June 2011.

[67] D. Aharonov. A simple proof that toffoli and hadamard are quantum universal. 2003. `quant-ph/0301040`.

[68] Joachim Kock. Frobenius algebras and 2-d topological quantum field theories. *Cambridge University Press*, 2003.

[69] P. W. Shor. Fault-tolerant quantum computation. *ArXiv Quantum Physics e-prints*, 1996. arXiv:quant-ph/9605011.

[70] Eric Dennis. Toward fault-tolerant quantum computation without concatenation. *Phys. Rev. A*, 63(5):052314, Apr 2001.

[71] A. Carboni and R.F.C. Walters. Cartesian bicategories i. *Journal of Pure and Applied Algebra*, 49:11–32, 1987.

[72] Bob Coecke and Ross Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. *aXriv preprint 0906.4725*, 2011.

[73] C. Kassel. Quantum groups. *Springer Graduate Texts in Mathematics*, 1994.

[74] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. *Proceedings of the 19th IEEE conference on Logic in Computer Science (LiCS'04)*, 2004.

[75] J. D. Biamonte. Nonperturbative k -body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins. *Phys. Rev. A*, 77(5):052331, May 2008.

[76] Dhiraj K. Pradhan. A Theory of Galois Switching Functions. *IEEE Tr. Computers*, 27(3):239–

248, 1978.

[77] S.L. Hurst. Multiple-Valued Logic - Its Status and its Future. *IEEE Tr. Computers*, 33(12):1160–1179, 1984.