

Regular language quantum states

Marta Florido-Llinàs^{1,2}, Álvaro M. Alhambra³, David Pérez-García⁴, and J. Ignacio Cirac^{1,2}

¹ Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Str. 1, 85748 Garching, Germany

² Munich Center for Quantum Science and Technology (MCQST), Schellingstr. 4, 80799 München, Germany

³ Instituto de Física Teórica UAM/CSIC, C/ Nicolás Cabrera 13-15, Cantoblanco, 28049 Madrid, Spain

⁴ Departamento de Análisis Matemático, Universidad Complutense de Madrid, 28040 Madrid, Spain

March 27, 2026

We introduce regular language states, a family of quantum many-body states. They are built from a special class of formal languages, called regular, which has been thoroughly studied in the field of computer science. They can be understood as the superposition of all the words in a regular language and encompass physically relevant states such as the GHZ-, W- or Dicke-states. By leveraging the theory of regular languages, we develop a theoretical framework to describe them. First, we express them in terms of matrix product states, providing efficient criteria to recognize them. We then develop a canonical form which allows us to formulate a fundamental theorem for the equivalence of regular language states, including under local unitary operations. We also exploit the theory of tensor networks to find an efficient criterion to determine when regular languages are shift-invariant.

1 Introduction

The identification of physically relevant families of many-body quantum states admitting a concise theoretical description has played a crucial role in the development of quantum science. Prominent examples of such families are stabilizer- [36], graph- [40], Gaussian- [81], or tensor network-states [59], which have turned into fundamental tools across diverse areas such as quantum error correction, measurement-based quantum computation, the classification of phases of matter, and quantum communication [12, 16, 66, 76, 80]. These families, among others, have allowed to establish unexpected but tight connections across different fields, ranging from quantum information theory, to condensed matter or quantum gravity [16, 23, 69, 85].

In this work, we introduce a family of quantum many-body states that we call *regular language states* (RLS). This family is inspired by the concept of formal languages, which lie at the heart of theoretical computer science, and consist of collections of strings

described by simple mathematical rules [2, 44, 77]. Formal languages provide the basic framework underlying modern models of computation, parsing, and program semantics; their hierarchy (regular, context-free, Turing decidable, etc.) organizes computational tasks by the amount of memory or control structure they require. Because of this, they form a unifying framework for describing both classical and quantum computational processes, ranging from compiler design to the theory of quantum finite automata and quantum regular grammars.

A fundamental subset of formal languages is the class of the so-called regular languages [83], which are widely used for efficient pattern manipulation in applications such as text processing, search engines, and compiler design [1, 3, 28]. The structural simplicity of regular languages naturally leads us to define sets of quantum states associated to them as the superposition of the words in the language for each given length. It also allows us to study their properties by leveraging the vast range of tools that have been developed for regular languages in the aforementioned contexts. This definition is also motivated by the fact that prominent examples of states appearing in quantum information and condensed matter physics are RLS, such as the GHZ- [37], W- [20], or Dicke-states [19].

Regular languages can be generated by finite automata [44, 71], which are a special type of restricted Turing machines that follow a simple set of rules. Finite automata are the simplest and most well-understood memory-bounded computational models: they capture exactly those tasks that can be carried out with constant internal memory, independent of the input size. Their transparency and mathematical rigidity have made them central not only in classical computation, but also in quantum computational models such as quantum finite automata, interactive protocols, and streaming algorithms [4, 11, 57]. For our purposes, automata provide a compact and operational way to encode the structure of a regular language, and therefore of the corresponding RLS. As we will see, this connection allows one to express RLS as matrix product states (MPS), the one-dimensional class of tensor network states.

Marta Florido-Llinàs: marta.florido.llinas@mpq.mpg.de

While the theoretical description of MPS is well established by now [16, 24, 63], it turns out that it cannot be applied to general RLS. Indeed, two basic tools, the canonical form and the fundamental theorem, require some technical conditions that are not met by such states. The former one removes the ambiguity in the description in terms of MPS, while the latter allows one to recognize if two MPS are LU-equivalent; that is, if they can be related to each other by local unitary operators. Both play a very relevant role in many applications of MPS, ranging from computational methods [73, 87] to the classification of symmetry-protected topological phases of matter in one dimension [14, 74]. Here we combine and extend some of the tools and techniques devised for regular languages and MPS in order to develop a theory that encompasses RLS. Specifically, we define a canonical form that can be found using RL tools, and a fundamental theorem that allows us to analyze the LU-equivalence of RLS. We also illustrate with examples how, in certain aspects, RLS significantly differ from standard MPS.

2 Regular language quantum states

Regular languages form one of the most fundamental classes in formal language theory, and their structure underlies the quantum states introduced in this work.

Given an alphabet Σ , a *word* is any finite sequence of letters of Σ , and Σ^* denotes the set of all possible words of arbitrary length. We will take $\Sigma = \{0, 1, \dots, d-1\}$, so that Σ^* is the set of all sequences composed of those numbers. A language, L , is simply a subset of Σ^* . There exist different types of languages depending on how they are characterized, but *regular languages* (RL) form the simplest nontrivial class. They admit two fully equivalent characterizations: one relies on an algebraic viewpoint, and the other on an automata-theoretic viewpoint.

From the algebraic perspective, regular languages are defined as the smallest class of languages built from the basic languages $\emptyset$, ε (the *empty string*) and single letters $a \in \Sigma$, which is closed under the so-called *regular operations*: concatenation ($L_1 L_2$), union ($L_1 \cup L_2$) and Kleene star ($L^* := \varepsilon \cup L \cup L L \cup \dots$). The expressions obtained under these rules are known as *regular expressions*. Therefore, they are patterns denoting (possibly infinite) sets of strings, whose manipulation and memory consumption are computationally very efficient, which explains their widespread use in applications such as pattern matching and text processing [28].

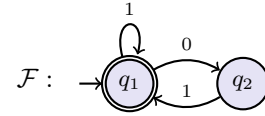
For $\Sigma = \{0, 1\}$ ($d = 2$) some examples are (a) $0^* 1^*$, including all words composed entirely of 0s or entirely of 1s, (b) $0^* 11^*$, with all words consisting of some 0's followed by one or more 1's, (c) $0^* 11^* 00^*$, with all words with a block of 0's, then a nonempty block of 1's and then again a block of 0's, i.e. all strings $0^i 1^j 0^k$

where $i \geq 0, j, k \geq 1$, (d) $0^* 10^*$, with all words with exactly one 1 and the rest 0s, and (e) $0^* 10^* 10^*$, with all words with exactly two ones in any locations and the rest 0s.

Equivalently, regular languages can be defined from an automata-theoretic viewpoint. In this picture, a regular language is precisely one that can be recognized by a *finite automaton*, a device with a finite set of internal memory states and a set of transition rules specifying how it reacts to each symbol of the input. Formally, they are structures characterized by a tuple $\mathcal{F} = \langle Q, \Sigma, \delta, I, F \rangle$, where Q is the set of internal states, $I, F \subseteq Q$ are the sets of initial and accepting states, respectively, and $\delta : Q \times (\Sigma \cup \varepsilon) \rightarrow \mathcal{P}(Q)$ is the transition function, where $\mathcal{P}(Q)$ denotes the power set of Q .

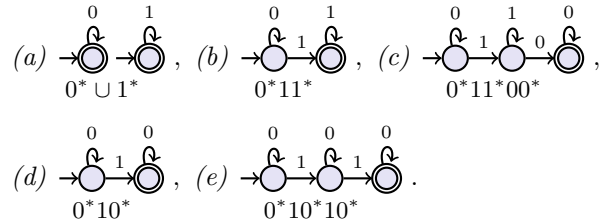
A word is *accepted* by the automaton if there exists at least one path through the automaton, following these transitions, that starts in an initial state and ends in an accepting one. Formally, a language is said to be *accepted* by $\mathcal{F}$ if for any word $w = x_1 x_2 \dots x_n \in L$, there is at least one path along the NFA starting at some $r_0 \in I$ and ending in $r_n \in F$, such that $r_0 \xrightarrow{x_1} r_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} r_n$, where $r_{i+1} \in \delta(r_i, x_{i+1})$ [77].

Given $\Sigma = \{0, 1\}$, we can consider as a first example the language $L := 1^*(011^*)^*$ of all words where every 0 is followed by at least one 1. It is accepted by the automaton $\mathcal{F} = \langle Q, \Sigma, \delta, I, F \rangle$, depicted below.



The short arrow marks the initial state, $I = \{1\}$. An arrow going from state q_i to state q_j , with the symbol x on top, indicates that $j \in \delta(i, x)$. For instance, we have $\delta(1, 0) = \{2\}$, meaning that if at some step of a computation we are in state q_1 , and emit output symbol 0, then we transition to q_2 . Similarly, $\delta(1, 1) = \{1\}$ and $\delta(2, 1) = \{1\}$. The double circle indicates the accepting state, $F = \{1\}$.

As further examples, we depict below automata accepting each of the regular expressions (a)-(e) introduced above:



The equivalence between the two viewpoints is known as Kleene's theorem [49] and is one of the most fundamental results in automata theory: it asserts that for every regular expression there exists an NFA that accepts exactly the same words, and vice versa.

We will be interested in a subset of NFA's called *unambiguous finite automata* (UFA) [71], in which any word has at most one accepting path. In fact, for any RL there always exists an UFA that accepts all and only its words [52, 67].

The automata viewpoint also provides intuition about which languages are regular. Since finite automata have only a constant amount of memory, they cannot count or store unbounded information about the input. For example, the language $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ is not regular because accepting it requires remembering exactly how many zeros have appeared before switching to ones, for arbitrarily long inputs.

To every regular language we can associate a family of quantum states consisting of the superposition with weights equal to one of all the words of length N in the language.

Definition 1. Given a RL L , the family of regular language states (RLS) associated to L is $L_q := \{|L_N\rangle\}_{N \in \mathbb{N}}$, where

$$|L_N\rangle = \sum_{w \in L \cap \Sigma^N} |w\rangle. \quad (1)$$

For the examples (a)-(e) introduced above in terms of their regular expressions and automata, the corresponding RLS are: (a) GHZ states on qubits [37], (b) the simplified ansatz for a domain-wall excitation [38], (c) a state with two domain-wall excitations, (d) W-states [20], and (e) Dicke states with two excitations of the same type [19].

Altogether, these RLS examples show that regular languages naturally encode many familiar quantum states from quantum information and quantum many-body physics. Moreover, they also show how RLS can systematically and compactly represent excited states of simple models, such as the transverse-field Ising model: (a) corresponds to the ground state, (b) and (c) represent the first and second excited states in the ferromagnetic phase (i.e. single and double domain-wall excitations), while (d) and (e) capture the structure of single and double quasiparticle excitations in the paramagnetic phase [70].

Note that, in order to keep a simple notation, we have not normalized the states. Furthermore, we have chosen all the coefficients in front of the states to be identical. One can extend RLS to include complex coefficients, although here we will mainly concentrate on the first case.

3 MPS representation of RLS

All the examples of the previous section can be written as MPS with a constant bond dimension, independent of N . In order to show that this holds for all RLS, we use a subset of NFA's called unambiguous finite automata (UFA) [71], for which any word has at most one accepting path. In fact, for any RL there always

exists an UFA that accepts all and only its words [52, 67]. With this, we can now establish a connection between RLS and MPS as follows.

Theorem 1. Any family of RLS $L_q = \{|L_N\rangle\}$ admits an MPS description with binary entries and constant bond dimension. In particular, given any UFA $\mathcal{F} = \langle Q, \Sigma, \delta, I, F \rangle$ that accepts L , then

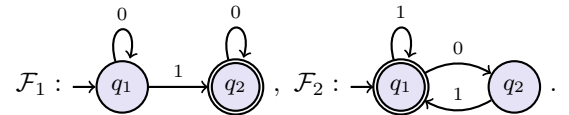
$$|L_N\rangle := \text{v}_l \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \text{---} \boxed{A} \text{---} \text{v}_r, \quad (2)$$

where the bond dimension is $D = |Q|$, and

$$\begin{cases} \text{v}_l := \sum_{i \in I} |i\rangle, \\ \text{v}_r := \sum_{f \in F} |f\rangle, \end{cases} \quad i \text{---} \boxed{A} \text{---} j = \begin{cases} 1 & \text{if } j \in \delta(i, x), \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The proof is provided in Appendix A. The resulting states are such that the coefficients in the computational basis belong to $\{0, 1\}$. Additionally, note that the resulting MPS is composed of a single tensor A , although it is not necessarily translationally invariant because it has open boundary conditions (OBC). Therefore, the entanglement of RLS obeys an area law, since they always admit an MPS representation; in particular, the size of any UFA accepting the RL upper bounds the entanglement of the associated RLS.

To illustrate the connection between RLS and MPS established in Theorem 1, we provide below the explicit MPS representation of two of the examples of RLS introduced in the previous section. Given $\Sigma = \{0, 1\}$, we consider the languages $L_1 := 0^*10^*$ of all words containing a single 1, and $L_2 := 1^*(011^*)^*$, with all words where every 0 is followed by at least one 1. They are accepted by



Using Eq. (3), the MPS associated to the first automaton has the following tensors,

$$A^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A^1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \langle v_l | = (1 \quad 0), \quad |v_r\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

while the automaton for L_2 corresponds to

$$A^0 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A^1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad \langle v_l | = (1 \quad 0), \quad |v_r\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Given any tensor A and vectors v_l, v_r with binary entries, is the corresponding MPS a RLS? To answer this, we note that the coefficient in front of each $|w\rangle$ equals the number of accepting paths for w , which is not necessarily one unless the automaton is unambiguous. Therefore, the MPS is a RLS if and only if the automaton associated to it according to Eq. (3) is an UFA.

Fortunately, RL theory provides a criterion to decide whether an NFA is actually an UFA in time

$O(m^2)$, where m is the number of transitions in the underlying NFA [71], which scales at worst as $O(d^2 D^4)$ since $m \leq dD^2$. By expressing this criterion in terms of the algebra of the MPS, consisting of all finite products of the MPS matrices and any linear combinations thereof [25], we present an alternative characterization of unambiguity in the lemma below, which can be checked in time $O(dD^3)$. We denote the algebra of the MPS as $\mathcal{A} := \text{Alg}(\{A^x\}) \subseteq \mathcal{M}_D(\mathbb{C})$, and define the sets $\mathcal{V}_L := \{(a^T | v_l)\}^{\otimes 2} \mid a \in \mathcal{A}\}$ and $\mathcal{V}_R := \{(a | v_r)\}^{\otimes 2} \mid a \in \mathcal{A}\}$, where a^T is the transpose of a .

Lemma 2. *Given an MPS defined by the binary tensors $\{A, v_l, v_r\}$ according to Eq. (2), it is a RLS if and only if, for all $m, n \in \{1, \dots, D\}$ with $m \neq n$,*

$$|m\rangle \otimes |n\rangle \in (\mathcal{V}_L)^\perp \cup (\mathcal{V}_R)^\perp.$$

This condition can be checked in time $O(dD^3)$.

The proof is provided in Appendix B.

A particularly relevant set of MPS are those that are translationally invariant (TI). Thus, we now address the problem of determining whether a RLS is TI. The first thing to notice is that these states correspond to the so-called shift-invariant languages [46]. A naive algorithm based on automata theory for deciding whether a RL is shift-invariant can scale as $O(dD^2 4^{D^2})$ with the bond dimension D of any MPS representation. However, using the theory of MPS through the lemma below, we can tackle this issue more efficiently.

Lemma 3. *A family of MPS-X states $\{|\psi_N(X, A)\rangle\}_N$,*

$$|\psi_N(X, A)\rangle := \text{---} \boxed{X} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \text{---} \boxed{A} \text{---} \quad (4)$$

where $X, A^i \in \mathcal{M}_D(\mathbb{C})$ is TI for all N , if and only if,

$$\text{Tr}[X[a, b]] = 0, \quad \forall a, b \in \text{Alg}(\{A^i\}).$$

As a corollary, we demonstrate that shift-invariant RL's can be identified in time $O(dD^3)$. The proofs can be found in Appendix C. This shows that tensor networks tools can also add a new perspective on the study of certain problems in computer science.

Corollary 4. *A RL is shift-invariant if and only if, for any MPS representation of the associated RLS with tensors $\{A, v_l, v_r\}$ of bond dimension D , it holds that $\langle v_l | [a, b] | v_r \rangle = 0$ for all $a, b \in \text{Alg}(\{A^i\})$. This condition can be checked in time $O(dD^3)$.*

4 Canonical form of RLS

Given two RLS and their MPS representations with bond dimensions D_1 and D_2 , how can we determine if they are equivalent? RL theory provides an algorithm to solve this problem in time $O(d(D_1 + D_2)^3)$

for the case of UFA's [17, 75, 78], and even in $O(\max_i dD_i \log D_i)$ when the automata are deterministic (i.e. when each state has at most one outgoing transition per output symbol) [44]. Note that standard MPS methods are ill-suited here: those for fixed string length N with site-dependent tensors scale as $(O(dN(D_1 + D_2)^3))$, while methods for all N with uniform tensors in the bulk assume periodic boundary conditions and therefore do not apply to RLS, whose MPS representations have open boundaries.

But is it possible to find a canonical form that is efficiently computable and uniquely associated with each RLS? Ideally, this canonical form would enable us not only to determine the equivalence between RLS, but also to address physically relevant questions such as the LU-equivalence of RLS, which is key to understand the presence of symmetries [14, 74], or to study the entanglement properties of RLS in terms of their interconvertibility [50, 72].

As a first attempt, we might consider using the already existing canonical form of MPS, but this is not generally possible, because many RLS do not fit into the traditional MPS framework. An example of this fact is the RLS consisting of W-states, for which imposing a TI MPS representation with periodic boundary conditions results in a bond dimension scaling with the system size as $\Omega(N^{1/(3+\delta)})$ for any $\delta > 0$ [56, 63]. On the other hand, note that the so-called left or right canonical form in the fixed N setting for non-uniform MPS (which refers to the tensors being left or right isometries [63]) are not applicable to our setting for all N .

In what follows, we propose an alternative representation that enables us to formulate a fundamental theorem for LU-equivalent RLS. For this purpose, we introduce first a canonical decomposition for regular languages, and use it as the basis to define a canonical form for RLS.

Given any RL, L , on alphabet Σ , we will relabel it as $\Sigma = \Sigma_\infty \cup \Sigma_f = \{0, 1, \dots\} \cup \{|\Sigma_\infty|, |\Sigma_\infty| + 1, \dots, d - 1\}$, where Σ_f contains the symbols of Σ whose number of appearances in any word of L is upper bounded by a constant. We will also use operation $S^{(m)} : \mathcal{L}_m \times (\Sigma_f)^m \rightarrow \Sigma^*$, where $\mathcal{L}_m$ is the set of all RL's in $(\Sigma_\infty \cup \{f\})^*$ with exactly m appearances of the symbol f in any word, such that $S^{(m)}(L_1, L_2)$ substitutes every occurrence of f in L_1 with the strings indicated by L_2 .

The *tree width* of an automaton measures the maximum number of distinct partial computations that any word can generate. Deterministic automata have tree width 1, while nondeterministic transitions on cycles can lead to arbitrarily many partial computations and hence infinite tree width. More details are included in Appendix D.

Theorem 5 (Canonical decomposition of a RL). *Any RL, L , can be decomposed in a unique way in terms of*

the pairs of RL's $\{(L_j^m, X_j^m)\}_{j,m}$, where $m \leq M$, as

$$L = \cup_m \cup_j S^{(m)}(L_j^m, X_j^m),$$

where $L_j^m \in \mathcal{L}_m$ and X_j^m consists of a union of strings of $(\Sigma_f)^m$, with the property that $L_j^m \cap L_k^m = \emptyset$ and $X_j^m \neq X_k^m$ unless $j = k$.

This decomposition is efficiently computable: the runtime scales polynomially with D if the input RL L is given as a DFA or an NFA of finite tree width k ($O(D^{2d^M})$ and $O(D^{2kd^M})$, respectively), or exponentially for an NFA of infinite tree width ($O(4^{(M+1)d^M D})$).

The proof is included in Appendix D, where the explicit time scaling of the algorithm with D, d, M is also included. For instance, $L_1 = 0^*10^*$ can be canonically decomposed as $\{(0^*f0^*, 1)\}$, so $L_1 = S^{(1)}(0^*f0^*, 1)$, and $L_2 = 0^*10^*20^* \cup 0^*20^*10^*$ can be canonically decomposed as $\{(0^*f0^*f0^*, 12 \cup 21)\}$, so $L_2 = S^{(2)}(0^*f0^*f0^*, 12 \cup 21)$.

We now embed L_j^m and X_j^m into quantum states, where $|L_{j,N}^m\rangle$ is the superposition of all strings of length N in L_j^m , and similarly $|X_j^m\rangle$ is the sum of all strings in X_j^m , which have length m by construction. We also let $\hat{S}^{(m),N}$ be the linear operator such that $\hat{S}^{(m),N}|x\rangle|y\rangle$ is the sum of the words of length N in $S^{(m)}(x, y)$. The dependence on N will be omitted for ease of notation.

Corollary 6. *Given a RLS L_q , it can be uniquely written in terms of RLS $\{(|L_j^m\rangle, |X_j^m\rangle)\}_{j,m}$ as*

$$|L^N\rangle = \sum_m \sum_j \hat{S}^{(m)} |L_j^m\rangle |X_j^m\rangle,$$

where L_j^m and X_j^m have the same properties as in Theorem 5, meaning that $\langle L_j^m | L_k^m \rangle = 0$ and $|X_j^m\rangle \neq |X_k^m\rangle$ unless $j = k$.

As the last step towards a canonical form of RLS, we need to look for a canonical choice of the MPS tensors for the RLS $\{(|L_j^m\rangle, |X_j^m\rangle)\}_{j,m}$ in Corollary 6. A RL can be accepted by many UFA's, each being a valid MPS representation according to Theorem 1. However, we cannot use the traditional canonical form of MPS because it is not generally applicable to RLS, as previously mentioned.

Automata theory provides a solution for this, by showing that any RL admits a canonical deterministic finite automaton (DFA). This canonical DFA is minimal and unique, up to relabelling of the internal states, among all DFA's accepting the language. Given an UFA of bond dimension D , the minimal equivalent DFA can be found in time $O(dD2^D)$, or in $O(dD \log D)$ if the automaton is already deterministic [42, 44, 52]. Note that there might be UFA's or ansätze as the one of Eq. (4) with $\text{rank}(X) \geq 1$ yielding smaller bond dimensions than the minimal DFA (see examples in Appendix E).

Definition 2 (Canonical form of RLS). *The canonical MPS representation of L_q thus corresponds to*

$$|L^N\rangle = \sum_m \sum_j \left(\text{MPS tensor with } \hat{x}_j^m \text{ and } l_j^m \text{ repeated } N \text{ times} \right), \quad (5)$$

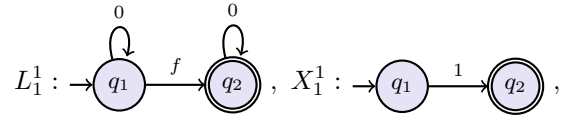
where the tensors l_j^m, x_j^m are the canonical DFA tensors of L_j^m, X_j^m , and the MPO on top is defined as

$$\hat{x}_j^m = \text{MPO} \otimes \mathbb{P}_\infty + \text{MPO} \otimes (f),$$

where $\mathbb{P}_\infty := \sum_{i=0}^{|\Sigma_\infty|-1} |i\rangle\langle i|$.

Therefore, the MPO effectively implements the action of $\hat{S}^{(m)}$, by letting the elements in Σ_∞ stay the same, and substituting the f symbols by the strings of X_j^m .

Let us illustrate Definition 2 with our previous example $L_1 = 0^*10^*$, which is canonically decomposed as $\{(L_1^1, X_1^1)\} \equiv \{(0^*f0^*, 1)\}$, whose canonical DFAs are



respectively. Using the correspondence in Theorem 1, this results in the MPS tensors

$$\hat{l}_1^1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \hat{x}_1^1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \hat{x}_1^1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

and the boundary vectors are $\langle v_l | = (1, 0)$ and $|v_r\rangle = (0, 1)^T$ for both.

Therefore, the final tensors appearing in the $|L_N\rangle$ expression of Eq. (5) are given by

$$\hat{x}_1^1 = \text{MPO} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \text{MPO} \otimes \begin{pmatrix} x_1^1 \\ f \end{pmatrix},$$

which has the following non-zero entries:

$$\hat{x}_1^1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \hat{x}_1^1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

5 Fundamental theorem of RLS

Let us now address the equivalence of two RLS under local unitary operations, using the canonical form of RLS. To achieve this, we restrict to the class of *sparse* regular languages, which have a number of words that scales polynomially with the word length,

i.e. $\|L_N\| = O(\text{poly}(N))$. These languages are well-characterized [79] and widely used due to the fact that their reduced complexity can simplify certain problems [9, 21, 26, 39, 41, 41, 79]. It can also be efficiently decided whether a language is sparse or not [31]. Note that the physically relevant examples (a)-(e) introduced in Section 2 are all sparse.

Theorem 7 (Fundamental theorem of sparse RLS). *Given two sparse RLS L_1, L_2 with canonical forms $\{(L_{1,j}^m, X_{1,j}^m)\}_{j,m}$ and $\{(L_{2,j}^m, X_{2,j}^m)\}_{j,m}$, they are related as*

$$|L_2^N\rangle = U^{\otimes N} |L_1^N\rangle, \quad \forall N,$$

for some unitary U if and only if there exists a relabeling π of the Σ_∞ symbols and a unitary $U_f \in \mathcal{U}_{|\Sigma_f|}$ such that

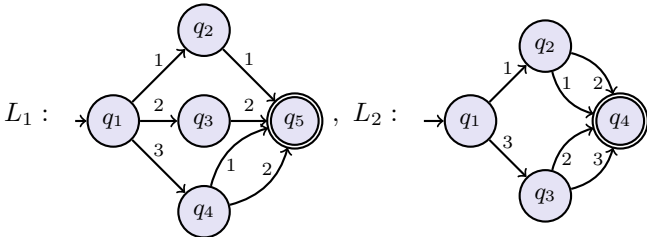
$$\begin{cases} L_{2,j}^m = \pi(L_{1,j}^m), \\ |X_{2,j}^m\rangle := U_f^{\otimes m} |X_{1,j}^m\rangle, \end{cases} \quad \forall j, m,$$

upon some reordering of the canonical form elements.

The above theorem demonstrates that LU equivalence for all N can be tested in terms of a constant number of finite-size sub-problems. The proof is provided in Appendix F.

As an illustration of this result, let us understand which languages are related to the W-state language $L_1 = 0^*10^*$ through local unitary operations. Since its canonical decomposition is $\{(L_1^1, X_1^1)\} = \{(0^*f0^*, 1)\}$, and thus $\Sigma = \{0\}$, $\Sigma_f = \{1\}$, the only possible relabelling is either the identity or the swap (i.e. $\pi(0) = 1$ and $\pi(1) = 0$), and the unitary is 1-dimensional so it trivializes and $|X_{1,1}^1\rangle = |X_{2,1}^1\rangle$, meaning that any other language L_2 that is locally unitarily equivalent to L_1 for all N must necessarily be either $L_2 = 0^*10^*$ or 1^*01^* .

There is a significant difference between the fundamental theorems of TI MPS and RLS. In the standard case of TI MPS, it enables us to translate local unitaries acting at the physical level into invertible matrices at the bond level. Therefore, the bond dimension does not change, which reflects the fact that local changes of basis do not affect the entanglement content of the state. However, this is not necessarily the case for RLS, as illustrated by the following example. Consider the finite languages L_1, L_2 with canonical representations $\{(ff, 11 \cup 22 \cup 31 \cup 32)\}$ and $\{(ff, 11 \cup 12 \cup 32 \cup 33)\}$. Even though they are related as $|L_2\rangle = U^{\otimes 2} |L_1\rangle$ for a unitary U , their minimal DFA's have bond dimensions 5 and 4, as shown below.



Since the size of the minimal DFA is a fundamental measure that quantifies the descriptive complexity of RL's [13, 84], we see that, even though LU operations do not change the entanglement content of RLS, they can still modify them in a way that is not captured merely by the bond dimension of the TI PBC canonical form.

6 Outlook

We have delved into the connection between tensor networks and regular languages, exploring a potential cross-fertilization of ideas and tools between them. Indeed, we have seen how RL techniques can aid in studying physically relevant families of states and in answering open questions about MPS. In this regard, we characterized MPS representing RLS and TI RLS, and developed a canonical form for them, since the standard canonical form of MPS is not generally applicable to RLS. Then, we used it to study the LU-equivalence of sparse RLS, thus making a first step towards both a phase classification for RLS, and a generalized understanding of their LU- and SLOCC-equivalence, a key task that is generally difficult for arbitrary multipartite quantum states [53]. Additional examples are provided in Appendix G.

One could also consider RLS with positive or complex coefficients. The resulting states would be linked to the areas of probabilistic graphical models and weighted finite automata (WFA), whose connection with tensor networks has been explored in previous works [18, 29, 34, 54, 68]. These, in turn, are commonly used in machine learning contexts: not only for probabilistic modeling, but also to improve the interpretability of black-box models and as the basis of learning algorithms for recurrent architectures [6, 54, 55, 58, 82]. Therefore, it would be interesting to clarify the relationship between canonical forms developed in the WFA literature (e.g. the minimal form or the Krohn-Rodes decomposition [10, 51]) and both our canonical decomposition and the standard MPS canonical form. A systematic understanding of these connections could enhance the interpretability of WFA-based representations and offer new tools to extend the LU-equivalence result of Theorem 7 beyond the sparse RLS setting and to SLOCC operations.

The next natural question that arises is whether there is a generalization of the RLS framework to two dimensions, or to broader types of languages such as the so-called context-free languages [35, 86]. Regarding the two-dimensional setting, the concept of *regular pictures* in 2D has been explored in the literature, aiming to retain the properties of 1D regular languages. Among the proposed extensions, the *2D online tessellation automaton* (OTA) stands out for having an equivalent characterization in terms of 2D regular expressions, as well as other desirable prop-

erties [32, 45, 86]. OTA, in turn, can be formulated as PEPS with tensors of a special, simple form (see Appendix H). However, unlike 1D RLS, there is no efficient characterization of the class of PEPS representing 2D RLS, nor a canonical form, since these problems are undecidable even when restricted to the very special form of the OTA PEPS tensors [5, 45].

This work lays the foundation for extending MPS theory beyond RLS. Following on this, [27] introduces a generalized canonical form for uniform MPS with an arbitrary boundary matrix, in which any such MPS is decomposed in terms of a backbone family of states called *algebraic RLS* that encodes its long-range, scale-invariant information. Algebraic RLS slightly generalize a subclass of the RLS introduced in this work, and inherit both their structural properties and the canonical decomposition developed here. This could enable extending results in the quantum many-body literature that are currently restricted to standard MPS or to highly specific families such as W- or Dicke-type MPS (e.g. the parent Hamiltonian construction for ground states [30] or for eigenstates that arise as quantum many-body scars [33]).

Moreover, identifying RLS as a structured and physically relevant class of quantum states, and formulating a theoretical framework for them, allows designing a compiler that explicitly exploits their automaton-like structure to yield more efficient quantum circuits for the preparation of RLS than current general-purpose or sparse-state compilers [8]. Since state preparation is a central subroutine in many quantum algorithms, improvements in this step can have a significant practical impact.

Beyond circuit synthesis, it remains an open direction for future work to determine whether specific families of RLS may offer advantages in certain tasks, in analogy with how GHZ states and their generalizations (which are RLS themselves) are exploited in many quantum information protocols and in quantum sensing [22, 61, 64].

Acknowledgements

MFL thanks Georgios Styliaris for helpful discussions and acknowledges support from the International Max-Planck Research School for Quantum Science and Technology (IMPRS-QST). This research is part of the Munich Quantum Valley (MQV), which is supported by the Bavarian State Government with funds from the High-tech Agenda Bayern Plus. AMA acknowledges support from the Spanish Agencia Estatal de Investigación through the grants “IFT Centro de Excelencia Severo Ochoa SEV-2016-0597” and “Ramón y Cajal RyC2021-031610-I”, financed by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR. DPG acknowledges support from the Spanish Ministry of Science and Innovation

MCIN/AEI/10.13039/501100011033 (grants CEX2023-001347-S and PID2020-113523GB-I00). This work has been financially supported by the Ministry for Digital Transformation and of Civil Service of the Spanish Government through the QUANTUM ENIA project call - Quantum Spain project, and by the European Union through the Recovery, Transformation and Resilience Plan – NextGenerationEU within the framework of the Digital Spain 2026 Agenda.

A Proof of Theorem 1 connecting RLS and MPS

Theorem 1. *Any family of RLS $L_q = \{|L_N\rangle\}$ admits an MPS description with binary entries and constant bond dimension. In particular, given any UFA $\mathcal{F} = \langle Q, \Sigma, \delta, I, F \rangle$ that accepts L , then*

$$|L_N\rangle := \textcircled{v_l} - \boxed{A} - \boxed{A} \overset{N \text{ times}}{\cdots} \boxed{A} - \textcircled{v_r}, \quad (2)$$

where the bond dimension is $D = |Q|$, and

$$\begin{cases} \textcircled{v_l} := \sum_{i \in I} \langle i |, \\ \textcircled{v_r} := \sum_{f \in F} |f\rangle, \end{cases} \quad i - \boxed{A} - j = \begin{cases} 1 & \text{if } j \in \delta(i, x), \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Proof. Kleene’s theorem asserts that every RL can be equivalently defined either through a regular expression or a finite automaton [49]. Then, given a RL L and a finite automaton that accepts it, $\mathcal{F} = \langle Q, \Sigma, \delta, I, F \rangle$, if we define tensor A and vectors v_l, v_r according to Eq. (3), then the language L is equal to

$$L = \{x_1 x_2 \dots x_N \in \Sigma^* \mid \textcircled{v_l} - \boxed{A} \overset{x_1}{\overset{x_2}{\cdots}} \boxed{A} \overset{x_N}{\cdots} \boxed{A} - \textcircled{v_r} > 0\},$$

and the states $|L_N\rangle$ defined in Eq. (2) are

$$|L_N\rangle = \sum_{w \in L \cap \Sigma^N} c_w |w\rangle,$$

where c_w equals the number of accepting paths for word w through the automaton. Therefore, when $\mathcal{F}$ is an UFA, all of the coefficients c_w are one, and thus the family of states $\{|L_N\rangle\}_N$ is a RLS. $\square$

B Proof of Lemma 2 to determine which MPS are RLS

Lemma 2. *Given an MPS defined by the binary tensors $\{A, v_l, v_r\}$ according to Eq. (2), it is a RLS if and only if, for all $m, n \in \{1, \dots, D\}$ with $m \neq n$,*

$$|m\rangle \otimes |n\rangle \in (\mathcal{V}_L)^\perp \cup (\mathcal{V}_R)^\perp.$$

This condition can be checked in time $O(dD^3)$.

Proof. The proof relies on Proposition 1.15 in [71]. First, we need some preliminary definitions.

An automaton $\mathcal{F}$ is said to be *trim*, if all of its states are *useful*. This means that each state is accessible from at least one of the initial states, and at least one path that reaches one of the accepting states arises from it.

Equivalently, state n is useful if there exist two strings $v, w \in \Sigma^*$ such that

$$\langle v_l | A^v | n \rangle \neq 0 \text{ and } \langle n | A^w | v_r \rangle \neq 0. \quad (6)$$

where tensor A and vectors v_l, v_r are obtained from $\mathcal{F}$ according to Theorem 1, and $A^y := A^{y_1} A^{y_2} \dots A^{y_l}$ for any string $y := y_1 y_2 \dots y_l \in \Sigma^*$.

Therefore, n is useless if and only if

$$|n\rangle \in (\mathcal{A}^T |v_l\rangle)^\perp \cup (\mathcal{A} |v_r\rangle)^\perp, \quad (7)$$

where $\mathcal{A} = \text{Alg}(\{A^x\})$. To assert this, we used the facts that $\mathcal{A}$ admits a basis $\{A^{w_1}, \dots, A^{w_M}\}$ in terms of a set of words $w_1, \dots, w_M \in \Sigma^*$ (it is enough to consider w_i of length upper bounded by $D^2 - 1$ [62]), and also $\mathcal{A}^\dagger = \mathcal{A}^T$ because there are no complex entries.

On the other hand, the *product automaton* of $\mathcal{F}$, denoted as $\mathcal{F} = \langle Q \times Q, \Sigma, \delta', I \times I, F \times F \rangle$, is the Cartesian product of $\mathcal{F}$ with itself, where

$$(k, l) \in \delta'((i, j), x) \iff k \in \delta(i, x) \text{ and } l \in \delta(j, x).$$

The MPS tensors A', v'_l, v'_r associated to $\mathcal{F} \times \mathcal{F}$ are thus $(A')^x = A^x \otimes A^x$, $\langle v'_l | = \langle v_l | \otimes \langle v_l |$ and $|v'_r\rangle = |v_r\rangle \otimes |v_r\rangle$. The *diagonal part* of $\mathcal{F} \times \mathcal{F}$ is the sub-automaton that arises from restricting to the set of internal states $\{(i, i)\}_{i \in Q}$.

By defining the vector subspaces $\mathcal{V}_L := \{(a^T | v_l)\rangle^{\otimes 2} \mid a \in \mathcal{A}\}$ and $\mathcal{V}_R := \{(a | v_r)\rangle^{\otimes 2} \mid a \in \mathcal{A}\}$, Eq. (7) tells us that state (m, n) in $\mathcal{F} \times \mathcal{F}$ being useless is equivalent to

$$|m\rangle \otimes |n\rangle \in (\mathcal{V}_L)^\perp \cup (\mathcal{V}_R)^\perp. \quad (8)$$

To prove the first part of the lemma, we can now use Proposition 1.15 in [71], which states that a finite automaton $\mathcal{F}$ is unambiguous if and only if the trim part of $\mathcal{F} \times \mathcal{F}$ is equal to its diagonal part, meaning that all states $(m, n) \in Q \times Q$ with $m \neq n$ are *useless*. Therefore, the MPS is a RLS if and only if Eq. (8) is true for all $m \neq n$.

Finally, we show that this condition can be checked in time $O(dD^3)$ in Algorithm 1 box. The procedure we propose scales better than the standard way to check unambiguity according to [71], since it does not require the construction of the product automaton. The latter runs in time $O(m^2)$, where m is the number of transitions in the NFA, which scales at worst as $O(d^2 D^4)$ since $m \leq dD^2$.

□

Algorithm 1: Check if an MPS is a RLS

Data: NFA $\mathcal{F} = \langle Q, \Sigma, \delta, I, F \rangle$ with $|Q| = D$, $|\Sigma| = d$.
Result: Decide if the MPS associated to $\mathcal{F}$ is a RLS in time $O(dD^3)$.

```

begin
   $\{|l_i\rangle\}_i \leftarrow$  Basis of  $\mathcal{A}^T |v_l\rangle$ ;
   $\{|r_i\rangle\}_i \leftarrow$  Basis of  $\mathcal{A} |v_r\rangle$ ;
  // These bases can be found in time
   $O(dD^3)$  [48].
  foreach  $i \in \{1, \dots, D\}, |l_j\rangle, |r_j\rangle$  do
     $L_{ij} \leftarrow \langle i | l_j \rangle$ ;
     $R_{ij} \leftarrow \langle i | r_j \rangle$ ;
  end
  // Runtime of the block above:  $O(D^3)$ 
  foreach  $m \neq n, m, n \in \{1, \dots, D\}$  do
    if  $L_{mi} L_{ni} = 0, \forall i$  then  $|m\rangle \otimes |n\rangle \in (\mathcal{V}_L)^\perp$ ;
    if  $R_{mi} R_{ni} = 0, \forall i$  then  $|m\rangle \otimes |n\rangle \in (\mathcal{V}_R)^\perp$ ;
    if  $|m\rangle \otimes |n\rangle \notin (\mathcal{V}_L)^\perp \cup (\mathcal{V}_R)^\perp$  then
      return The MPS is not a RLS;
    end
  end
  // Runtime of the block above:  $O(D^3)$ 
  return The MPS is a RLS;
end

```

C Proof of Lemma 3 and Corollary 4 about translationally invariant RLS

Lemma 3. A family of MPS-X states $\{|\psi_N(X, A)\rangle\}_N$,

$$|\psi_N(X, A)\rangle := \boxed{X \text{---} A \text{---} A \text{---} \dots \text{---} A}^N, \quad (4)$$

where $X, A^i \in \mathcal{M}_D(\mathbb{C})$ is TI for all N , if and only if,

$$\text{Tr}[X[a, b]] = 0, \quad \forall a, b \in \text{Alg}(\{A^i\}).$$

Proof. First, let us reformulate the TI condition as follows. Given the state in Eq. (4)

$$|\psi_N(X, A)\rangle = \sum_{i_1, \dots, i_N} \text{Tr}[X A^{i_1} A^{i_2} \dots A^{i_N}] |i_1 i_2 \dots i_N\rangle,$$

it is TI for all N , if and only if, for all $r \leq N - 1$, and for all values of $i_1, \dots, i_N \in \{0, \dots, d - 1\}$, it holds that

$$\text{Tr}[X A^{i_1} A^{i_2} \dots A^{i_N}] = \text{Tr}[X A^{i_{r+1}} \dots A^{i_N} A^{i_1} \dots A^{i_r}]$$

Regrouping the terms together, this is equivalent to

$$\text{Tr}[X[A^{i_1} \dots A^{i_r}, A^{i_{r+1}} \dots A^{i_N}]] = 0, \quad \forall i_1, \dots, i_N. \quad (9)$$

On the other hand, the algebra of the MPS matrices, $\mathcal{A} := \text{Alg}(\{A^i\})$, admits a basis of the form $\{A^{w_1}, \dots, A^{w_M}\}$, where $A^{w_i} := A^{w_{i,1}} \dots A^{w_{i,l_i}}$ and we can take $l_i \leq D^2 - 1$ [62].

Then, for any elements $a, b \in \mathcal{A}$, we know that there exist $\alpha_i, \beta_j \in \mathbb{C}$ such that

$$a = \sum_i \alpha_i A^{w_i}, \quad b = \sum_j \beta_j A^{w_j}.$$

Therefore, $|\psi_N(X, A)\rangle$ are TI for all N if and only if

$$\begin{aligned} \text{Tr}[X[a, b]] &= \sum_i \sum_j \alpha_i \beta_j \text{Tr}[X[A^{w_i}, A^{w_j}]] \\ &= \sum_i \sum_j \alpha_i \beta_j \underbrace{\text{Tr}[X[A^{w_{i,1}} \dots A^{w_{i,l_i}}, A^{w_{j,1}} \dots A^{w_{j,l_j}}]]}_{=0 \text{ by Eq. (9)}} \\ &= 0, \quad \forall a, b \in \mathcal{A}. \end{aligned}$$

□

Corollary 4. *A RL is shift-invariant if and only if, for any MPS representation of the associated RLS with tensors $\{A, v_l, v_r\}$ of bond dimension D , it holds that $\langle v_l | [a, b] | v_r \rangle = 0$ for all $a, b \in \text{Alg}(\{A^i\})$. This condition can be checked in time $O(dD^3)$.*

Proof. The first part of the statement is a direct application of Lemma 3. To prove the runtime, we outline a scheme to check shift invariance in Algorithm 2 box.

Algorithm 2: Decide if a RL is shift-invariant

Data: RL L and any MPS representation $\{A, v_l, v_r\}$, with bond dimension D and $|\Sigma| = d$.

Result: Decide if L is shift-invariant in time $O(dD^3)$.

```

begin
   $\mathcal{S} := \{(w_i, \langle l_i |, |r_i \rangle)\} \leftarrow$  Set of triples consisting
  of strings  $w_i \in \Sigma^*$  and vectors  $\langle l_i | := \langle v_l | A^{w_i}$ ,
   $|r_i \rangle := A^{w_i} |v_r\rangle$ , which fully span  $\langle v_l | \mathcal{A}$  and
   $\mathcal{A} |v_r\rangle$ , respectively.
  // One can compute the set  $\mathcal{S}$  by finding
  bases of  $\langle v_l | \mathcal{A}$  and  $\mathcal{A} |v_r\rangle$ , which is
  doable in time  $O(dD^3)$  [48]. Note
  that  $\mathcal{S}$  contains at most  $2D$  elements.
  foreach  $(w_i, \langle l_i |, |r_i \rangle), (w_j, \langle l_j |, |r_j \rangle) \in \mathcal{S}$  do
    if  $\langle l_i | r_j \rangle \neq \langle l_j | r_i \rangle$  then return  $L$  is not
    shift-invariant;
  end
  // Runtime of the block above:  $O(D^3)$ .
  return  $L$  is shift-invariant;
end
```

□

Note that a naive automata implementation to determine if a RL L is shift-invariant can result in an algorithm scaling much worse with the bond dimension, $O(dD^2 4^{D^2})$. We show this in Algorithm 3 box. The improvement of our scheme relies on the fact that, while the construction of an equivalent DFA leads to an exponential overhead in the naive automata implementation, we can avoid doing so by leveraging Lemma 3.

Algorithm 3: Decide if a RL is shift-invariant (naive automata implementation)

Data: RL L and any NFA representation with D internal states and $|\Sigma| = d$.

Result: Decide if L is shift-invariant in time $O(dD^2 4^{D^2})$.

```

begin
   $L' \leftarrow \text{shift}(L) := \{vu \mid uv \in L\}$ , through an
  NFA with at most  $2D^2 + 1$  internal states [46]
  ; // Runtime:  $O(D^2)$ 
  Construct an equivalent DFA accepting  $L \leftarrow$ 
  Can be done with Rabin-Scott powerset
  construction, resulting in a DFA with at most
   $2^D$  states [65] ; // Runtime:  $O(dD2^D)$ .
  Construct an equivalent DFA accepting  $L' \leftarrow$ 
  Results in at most  $2^{2D^2+1}$  states ;
  // Runtime:  $O(dD^2 4^{D^2})$ .
  if  $L = L'$  then
    return  $L$  is shift-invariant ;
  else
    return  $L$  is not shift-invariant ;
  end
  // Equivalence checking of these DFA's
  can be done in time  $O(dD^2 4^{D^2})$  [43].
end
```

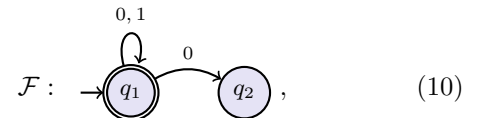
D Proof of Theorem 5 about the existence and uniqueness of a canonical form for RL's

We first recall the notion of tree width for finite automata. A partial computation of a word w on an automaton $\mathcal{F}$ is a path that reads w for as long as transitions are defined, terminating once no outgoing transition exists. We denote by $\text{tw}(\mathcal{F}, w)$ the total number of such partial computations.

Definition 3 (Tree width). *The tree width of automaton $\mathcal{F}$ is defined as*

$$\text{TW}(\mathcal{F}) := \sup_{w \in \Sigma^*} \text{tw}(\mathcal{F}, w).$$

By definition, any DFA has tree width 1. In contrast, even unambiguous NFAs may have infinite tree width. A simple example is the following automaton,



which provides an unambiguous representation of the language $(0 \cup 1)^*$. The presence of a nondeterministic transition on a cycle implies infinite tree width [60]: for any $\ell \in \mathbb{N}$, the word $w = 0^\ell$ admits $\ell + 1$ partial computations, namely the accepting path q_1^ℓ and the incomplete paths $q_1^i q_2$ for $i = 0, \dots, \ell - 1$.

The relevance of tree width in what follows stems from its role in bounding the cost of determinization.

The standard Rabin–Scott powerset construction [65] transforms an NFA with D states into an equivalent DFA with at most 2^D states, in time $O(dD2^D)$ in the worst case. However, when the NFA has finite tree width k , the resulting DFA has only $O(D^k)$ states and can be constructed in time $O(dD^{k+1})$ [60]. This distinction underlies the polynomial-versus-exponential complexity behavior in the canonical decomposition algorithm.

Theorem 5 (Canonical decomposition of a RL). *Any RL, L , can be decomposed in a unique way in terms of the pairs of RL's $\{(L_j^m, X_j^m)\}_{j,m}$, where $m \leq M$, as*

$$L = \cup_m \cup_j S^{(m)}(L_j^m, X_j^m),$$

where $L_j^m \in \mathcal{L}_m$ and X_j^m consists of a union of strings of $(\Sigma_f)^m$, with the property that $L_j^m \cap L_k^m = \emptyset$ and $X_j^m \neq X_k^m$ unless $j = k$.

This decomposition is efficiently computable: the runtime scales polynomially with D if the input RL L is given as a DFA or an NFA of finite tree width k ($O(D^{2d^M})$ and $O(D^{2kd^M})$, respectively), or exponentially for an NFA of infinite tree width ($O(4^{(M+1)d^M D})$).

Proof. To begin with, we see that a decomposition with such properties is unique. Indeed, assume that L admits two such decompositions, $\{(L_{1,j}^m, X_{1,j}^m)\}_{j,m}$ and $\{(L_{2,j}^m, X_{2,j}^m)\}_{j,m}$.

First, we see that $L_{1,j}^m = L_{2,j}^m$ upon some relabelling of the elements. Suppose that there exist $v, w \in L_{2,k}^m$ such that $v \in L_{1,i}^m$ and $w \in L_{1,j}^m$, for some $i \neq j$. This would imply that $X_{1,i}^m = X_{2,k}^m = X_{1,j}^m$, which is not possible unless $i = j$. Due to the symmetry of the argument under exchange of L_1, L_2 , it follows that $L_{1,j}^m = L_{2,j}^m$ upon relabelling, and as a consequence, $X_{1,j}^m = X_{2,j}^m$.

Now, we show that it is always possible to find the canonical decomposition of L by providing an explicit algorithm to do so. We assume that we are given an NFA representation of L of bond dimension D .

First of all, we find for each m all finite words $x_1 \dots x_m \in \Sigma_f^m$ such that the languages

$$L_{x_1 \dots x_m} := T_{\Sigma_f \rightarrow f}(L \cap \Sigma_\infty^* x_1 \Sigma_\infty^* x_2 \dots \Sigma_\infty^* x_m \Sigma_\infty^*) \quad (11)$$

are non-empty, where $T_{\Sigma_f \rightarrow f}$ replaces all symbols in Σ_f by the single symbol f . The intersection in Eq. (11) can be obtained through the product automaton construction [71], resulting in a bond dimension of at most $(m+1)D$, and computable in time $O(t_1 t_2)$, where t_i is the number of edges in the NFA. Note that $t_1 \leq dD^2$ for the NFA accepting L , and $t_2 = (m+1)|\Sigma_\infty| + m$ [71].

Moreover, the substitution $T_{\Sigma_f \rightarrow f}$ can also be done in time $O(t_1 t_2)$, since we just need to go through the $t_1 t_2$ edges of the product automaton and change the ones with label in Σ_f by the output symbol f .

Therefore, the language can be expressed as

$$L = \cup_m \cup_{x_1 \dots x_m} S^{(m)}(L_{x_1 x_2 \dots x_m}, x_1 x_2 \dots x_m).$$

However, this decomposition does not satisfy the desired properties yet, since $\{L_{x_1 \dots x_m}\}$ are not necessarily pairwise disjoint. For example, for the language $L = 0^*10^*20^* \cup 0^*20^*10^*$, we would have that $L_{12} = L_{21} = 0^*f0^*f0^*$. We proceed to define a new partition $\{L_j^m\}$ of pairwise disjoint RL's, with the desirable property that $L_j^m \cap L_{x_1 \dots x_m} \neq \emptyset \implies L_j^m \subseteq L_{x_1 \dots x_m}$.

For each m , let $\mathcal{S} := \{S_1, S_2, \dots\}$ be the power set of the words $\{x_1 \dots x_m \in \Sigma_f^m \mid L_{x_1 \dots x_m} \neq \emptyset\}$. Then, the sets defined as

$$L_i^m := (\cap_{x_1 \dots x_m \in S_i} L_{x_1 \dots x_m}) \cap (\cap_{x_1 \dots x_m \notin S_i} L_{x_1 \dots x_m}^c), \quad (12)$$

where L^c denotes the complement of L , achieve the desired properties, and can be computed again using automata tools [43]. We neglect the L_i^m that are empty. For clarity, an example of this partition appears in Figure 1.

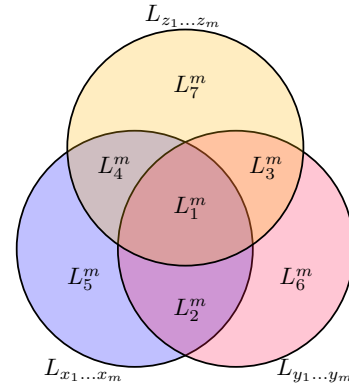


Figure 1: Example of how the partition in Eq. (12) looks like in the proof of Theorem 5, given three initial sets $L_{x_1 \dots x_m}, L_{y_1 \dots y_m}, L_{z_1 \dots z_m}$.

Then, we can rewrite the language as

$$L = \cup_m \cup_j S^{(m)}(L_j^m, X_j^m),$$

where $X_j^m := \cup_{x_1 \dots x_m \in S_j} x_1 \dots x_m$. Note that, by construction, $X_j^m \neq X_k^m$ unless $j = k$. Therefore, the desired conditions on L_j^m, X_j^m are satisfied.

We show in Algorithm 4 that this scheme to find the canonical decomposition has a worst-case complexity of $O(M(d2^{1+2(M+1)D})^{d^M})$ when the input automaton is an NFA of infinite tree width, which scales with the bond dimension D as $O(4^{(m+1)d^M D})$. The exponential dependence on D originates entirely from the determinization step in Algorithm 4, which is required to construct the complements $L_{x_1 \dots x_m}^c$.

We now show that, when the input automaton accepting L has finite tree width k , the overall scaling becomes polynomial in D , rather than exponential. The central observation is that the tree width of the automata recognizing $L_{x_1 \dots x_m}$ (defined in Eq. (11)) can be uniformly bounded by k , for each $x_1, \dots, x_m \in \Sigma_f$.

We begin with the intermediate language $\tilde{L}_{x_1 \dots x_m} := L \cap \Sigma_\infty^* x_1 \Sigma_\infty^* x_2 \dots x_m \Sigma_\infty^*$. The intersection does not increase the tree width. Indeed, one has $\text{TW}(L_1 \cap L_2) \leq \text{TW}(L_1) \cdot \text{TW}(L_2)$, and the automaton recognizing $\Sigma_\infty^* x_1 \Sigma_\infty^* x_2 \dots x_m \Sigma_\infty^*$ has tree width 1 whenever $x_i \in \Sigma_f$ for all i .

Therefore, after trimming useless states, which can be done in time $O(dD^2)$ [71], we obtain an NFA for $\tilde{L}_{x_1 \dots x_m}$ of size at most $(m+1)D$ and tree width at most k . [71].

Next, we analyze the effect of substituting the symbols of Σ_f by the single letter f in $L_{x_1 \dots x_m} = T_{\Sigma_f \rightarrow f}(\tilde{L}_{x_1 \dots x_m})$. Replacing symbols in Σ_∞ could introduce nondeterministic transitions on cycles (illustrated e.g. in Eq. (10)) and thus infinite tree width [60, Cor. 3.1]. This cannot occur here: by definition of Σ_f , symbols in Σ_f appear only a bounded number of times in any word of L , so no cycle of the automaton can carry a transition labeled by Σ_f . Consequently, substituting them by the single symbol f cannot generate infinitely many partial computations, and the tree width remains finite.

More precisely, fix any string $w \in (\Sigma_\infty \cup \{f\})^*$, which we write without loss of generality as

$$w = s_0 f s_1 f s_2 \dots f s_p$$

for some $p \in \mathbb{N}$, $y_i \in \Sigma_f$ and $s_i \in \Sigma_\infty^*$. The number of partial computations of w in the automaton for $L_{x_1 \dots x_m}$ coincides with the number of partial computations of the string

$$\tilde{w} := \begin{cases} s_0 x_1 s_1 \dots x_p s_p & \text{if } p \leq m, \\ s_0 x_1 s_1 \dots x_m s_m y_{m+1} s_{m+1} \dots y_p s_p & \text{if } p > m, \end{cases}$$

in the trimmed automaton for $\tilde{L}_{x_1 \dots x_m}$, for any arbitrary choice of $y_{m+1}, \dots, y_p \in \Sigma_f$. Indeed, once a symbol different from the prescribed x_i is encountered, the trimmed automaton has no outgoing transition and the computation halts. It follows that

$$\text{TW}(L_{x_1 \dots x_m}) \leq k.$$

By Lemma 8.1 of [7], determinizing the NFA accepting $L_{x_1 \dots x_m}$, of size $(m+1)D$ and tree width $\leq k$ yields a DFA with $O([(m+1)D]^k)$ states, constructible in time $O(d[(m+1)D]^{1+k})$.

In the next block of Algorithm 4, each language L_i^m is obtained as the intersection of $|\Sigma_f|^m$ automata, each of size at most $(m+1)D$ (for $L_{x_1 \dots x_m}$) or $O([(m+1)D]^k)$ (for $L_{x_1 \dots x_m}^c$). The resulting automaton for L_i^m therefore has size $O([(m+1)D]^k)$ and can be computed in time $O(\prod_{i=1}^m t_i)$ where t_i is the number of transitions of each automaton in the intersection. Since $t_i \leq O(d[(m+1)D]^{2k})$, the total computation time is $O((d[(m+1)D]^{2k})^{d^m})$. This must be repeated for all $S_i \in \mathcal{S}$, with $|\mathcal{S}| = 2^{|\Sigma_f|^m}$, so the total runtime of the block is $O((2d[(m+1)D]^{2k})^{d^m})$. Treating m, d as constants, this scales as $O(D^{2kd^m})$.

Finally, performing the above steps for all $m \leq M$ yields an overall runtime of $O(M(2d[(M+1)D]^{2k})^{d^M})$, which scales with the bond dimension as $O(D^{2kd^M})$. In particular, for DFAs one has $k = 1$, so this becomes $O(D2d^M)$, completing the proof.

Algorithm 4: Canonical decomposition of a RL

Data: RL L and any NFA representation of size D , $|\Sigma| = d$, and with at most M appearances of symbols of Σ_f in all words.

Result: Compute the canonical decomposition $\{(L_j^m, X_j^m)\}_{j,m}$ of L in time $O(M(d2^{1+2(M+1)D})^{d^M})$.

```

begin
  forall  $m \leq M$  do
    foreach  $x_1 x_2 \dots x_m \in (\Sigma_f)^m$  do
       $\tilde{L}_{x_1 \dots x_m} \leftarrow$ 
         $L \cap \Sigma_\infty^* x_1 \Sigma_\infty^* x_2 \dots \Sigma_\infty^* x_m \Sigma_\infty^*$  ;
       $L_{x_1 \dots x_m} \leftarrow T_{\Sigma_f \rightarrow f}(\tilde{L}_{x_1 \dots x_m})$  ;
      if  $L_{x_1 \dots x_m} = \emptyset$  then Discard it;
      // Runtime:  $O(md^2 D^2)$  (Prop. 1.4
        and 1.11 in [71])
       $L_{x_1 \dots x_m}^c \leftarrow$  Complement of  $L_{x_1 \dots x_m}$ 
        (first, determinize the NFA using
        Rabin-Scott powerset construction [65],
        then calculate the complement of the
        DFA by changing  $F \leftarrow Q \setminus F$  [71] ;
      // Runtime:  $O(d(m+1)D2^{(m+1)D})$ 
    end
    // Runtime of the block:  $O(d^m 2^{mD})$ 
     $\mathcal{S} \leftarrow$  Power set of words  $x_1 \dots x_m \in (\Sigma_f)^m$ 
      such that  $L_{x_1 \dots x_m} \neq \emptyset$  ( $|\mathcal{S}| \leq 2^{|\Sigma_f|^m}$ ).
    foreach  $S_i \in \mathcal{S}$  do
       $L_i^m \leftarrow$  Eq. (12) (involves computing the
        intersection of  $|\Sigma_f|^m$  NFA's of size
        either  $\leq (m+1)D$  or  $\leq 2^{(m+1)D}$ , and
        checking emptiness) ;
      // Runtime:  $O((d4^{(m+1)D})^{d^m})$ 
      if  $L_{x_1 \dots x_m} \neq \emptyset$  then
         $X_i^m \leftarrow \cup_{x_1 \dots x_m \in S_i} x_1 \dots x_m$  ;
        // Runtime:  $O(d^m)$ 
      end
    end
    // Runtime of the block:
       $O((d2^{1+2(M+1)D})^{d^m})$ 
  end
  return  $\{(L_i^m, X_i^m)\}$  ;
  // Total runtime:  $O(M(d2^{1+2(M+1)D})^{d^M})$ 
end

```

□

E Examples of the non-minimality of the canonical DFA

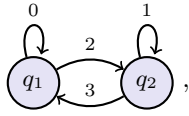
The minimal canonical DFA is not necessarily minimal with respect to all valid MPS representations of

RLS, as has been studied in the RL literature [67]. However, if we consider a more general ansatz of the form of Eq. (4) motivated by MPS theory, then we can obtain even smaller bond dimensions than the minimal UFA.

The first example illustrating this corresponds to $L = (0^*21^*3)^* \cup (0^*31^*2)^*$. While the Schmidt rank imposes a lower bound of 4 on any UFA representation of L , it can also be described by the following MPS-X with $D = 2$ and $\text{rank}(X) = 2$,

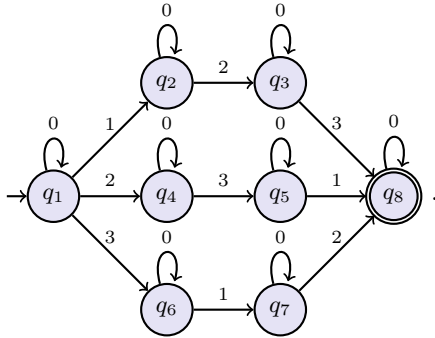
$$X = \begin{pmatrix} 1 & \\ & 1 \end{pmatrix}, A^0 = \begin{pmatrix} 1 & \\ & 0 \end{pmatrix}, A^1 = \begin{pmatrix} 0 & \\ & 1 \end{pmatrix}, \\ A^2 = \begin{pmatrix} 0 & 1 \\ & 0 \end{pmatrix}, A^3 = \begin{pmatrix} 0 & \\ 1 & 0 \end{pmatrix}.$$

Diagrammatically, this can be seen as the automaton



where X tells us that any path through the automaton can either start at q_1 and accept at q_1 , or start at q_2 and accept at q_2 .

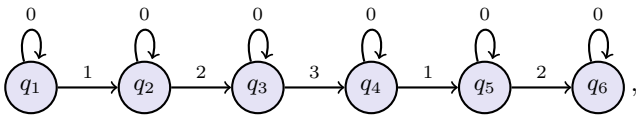
As a second example, we consider $L = 0^*(10^*20^*3 \cup 20^*30^*1 \cup 30^*10^*2)0^*$. The Schmidt rank gives a lower bound of 8 for the size of the minimal UFA, which is equal to the minimal DFA in this case,



However, it can also be described by an MPS-X with $D = 6$ and $\text{rank}(X) = 3$ with

$$X = |4\rangle\langle 1| + |5\rangle\langle 2| + |6\rangle\langle 3|, \\ A^0 = \mathbb{1}_6, A^1 = |1\rangle\langle 2| + |4\rangle\langle 5|, \\ A^2 = |2\rangle\langle 3| + |5\rangle\langle 6|, A^3 = |3\rangle\langle 4|,$$

which can be pictured as the automaton



where X tells us that any path through it can either start at q_1 and accept at q_4 , start at q_2 and accept at q_5 , or start at q_3 and accept at q_6 .

F Proof of Theorem 7 about the LU-equivalence of sparse RLS

A RL L is sparse, with a number of words scaling as $O(N^k)$ with the word length N for some natural number k , if and only if it can be described with a regex of the form

$$L = \bigcup_{i=1}^Q z_{i,0}(y_{i,1})^* z_{i,1} \dots (y_{i,t_i})^* z_{i,t_i}, \quad (13)$$

where Q is a constant, $t_i \leq k + 1$ for all i , and y_i, z_i are strings in Σ^* [79]. Whether a RL is sparse or not can be decided in time $O(D + m)$, where m is the number of transitions of an automaton accepting L ($m \leq dD^2$) [31].

Note that, for a RL L accepted by a UFA of size D , the number of words of fixed length N in L can be obtained as the norm of the associated MPS, as well as through the adjacency matrix of the UFA [15] in time $O(dD^2 + \log(N)D^3)$, or with a dynamic programming procedure [47] in time $O(dND^2)$.

Using the canonical representation in Corollary 6, we can characterize RLS that are equivalent under the action of $U^{\otimes N}$ for some unitary U .

Theorem 7 (Fundamental theorem of sparse RLS). *Given two sparse RLS L_1, L_2 with canonical forms $\{(L_{1,j}^m, X_{1,j}^m)\}_{j,m}$ and $\{(L_{2,j}^m, X_{2,j}^m)\}_{j,m}$, they are related as*

$$|L_2^N\rangle = U^{\otimes N} |L_1^N\rangle, \quad \forall N,$$

for some unitary U if and only if there exists a relabeling π of the Σ_∞ symbols and a unitary $U_f \in \mathcal{U}_{|\Sigma_f|}$ such that

$$\begin{cases} L_{2,j}^m = \pi(L_{1,j}^m), \\ |X_{2,j}^{(m)}\rangle := U_f^{\otimes m} |X_{1,j}^m\rangle, \end{cases} \quad \forall j, m,$$

upon some reordering of the canonical form elements.

Proof. First, we assume that $|L_2^N\rangle = U^{\otimes N} |L_1^N\rangle$ for all N . As the initial step, we show that U acts as a permutation when restricted to Σ_∞ .

Given $x \in \Sigma_\infty$, assume first for simplicity that it appears in L_2 in one of the strings $y_{i,j}$ of Eq. (13) as $y_{i,j} = x$. Then, we know that for each $x \in \Sigma_\infty$ there exists a set of words $\{s_x^N\}_N \in L_2$ of the form $s_x^N = s_1 x^{N-a} s_2$, where $a := |s_1| + |s_2|$, and s_1, s_2 remain the same for all N . Then,

$$\begin{aligned} 1 &= \langle s_i^N | U^{\otimes N} | L_1^N \rangle \\ &= \sum_{\substack{w_1 w_2 w_3 \in L_1 \cap \Sigma_\infty^N \\ |w_2| = N-a}} \langle s_1 | U^{\otimes |w_1|} | w_1 \rangle \langle x x \dots x | U^{\otimes (N-a)} | w_2 \rangle \\ &\quad \langle s_2 | U^{\otimes |w_3|} | w_3 \rangle. \end{aligned} \quad (14)$$

Assume that $|\langle x | U | y \rangle| \leq \lambda < 1$ for all $y \in \Sigma_\infty$. Since L_1 contains $O(\text{poly}(N))$ words of length N , Eq. (14) would necessarily imply that $1 = O(\text{poly}(N)\lambda^N)$, which leads to a contradiction as $N \rightarrow \infty$.

Similarly, if $|y_{i,j}| > 1$ for all strings where $x \in \Sigma_\infty$ appears, the argument above still holds if we just choose $s_x^N = s_1(y_{i,j})^{N-a} s_2$ in L_2 , for any $y_{i,j}$ containing x .

Therefore, there is a permutation $\pi \in \mathcal{S}_{|\Sigma_\infty|}$ such that $U|x\rangle = |\pi(x)\rangle$ for $x \in \Sigma_\infty$, so $U = U_\pi \oplus U_f$, where $U_f \in \mathcal{U}_{|\Sigma_f|}$. The unitary U thus acts on L_1 as

$$\begin{aligned} U^{\otimes N} |L_1^N\rangle &= \sum_m \sum_j \hat{S}^{(m)} (U_\pi^{\otimes N} |L_{1,j}^m\rangle) (U_f^{\otimes m} |X_{1,j}^m\rangle) \\ &= \sum_m \sum_j \hat{S}^{(m)} |\pi(L_{1,j}^m)\rangle (U_f^{\otimes m} |X_{1,j}^m\rangle) \\ &= \sum_m \sum_j \hat{S}^{(m)} |L_{2,j}^{m,N}\rangle |X_{2,j}^m\rangle = |L_2^N\rangle. \end{aligned}$$

Due to the uniqueness of the canonical representation, we have that the above is only possible if, for each m , there exists some reordering of the elements in the canonical representation, such that for all j ,

$$\begin{cases} |L_{2,j}^{m,N}\rangle = |\pi(L_{1,j}^{m,N})\rangle \\ |X_{2,j}^m\rangle := U_f^{\otimes m} |X_{1,j}^m\rangle \end{cases} \quad (15)$$

The forward direction of the claim follows since Eq. (15) holds for all N .

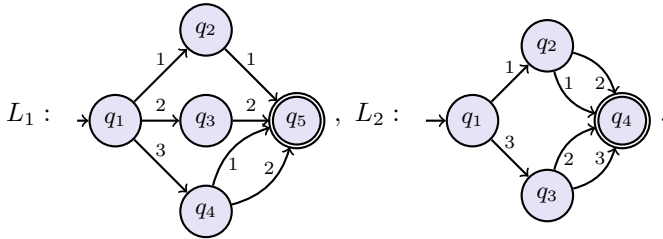
To prove the other direction, we just need to note that, defining $U := U_\pi \oplus U_f$, we would have $|L_2^N\rangle = U^{\otimes N} |L_1^N\rangle$ for all N as desired. $\square$

G Examples of LU and SLOCC-equivalent RLS

First, we reproduce the example of LU-equivalent RLS in the main text and provide the explicit unitary relating them for completeness. Consider

$$\begin{cases} L_1 = 11 \cup 22 \cup 31 \cup 32, \\ L_2 = 11 \cup 12 \cup 32 \cup 33, \end{cases}$$

whose minimal DFAs have bond dimensions 5 and 4, respectively, as shown below,



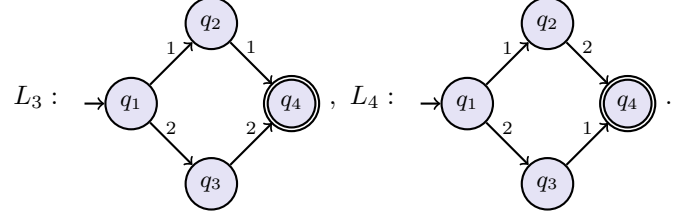
They are related as $|L_2\rangle = U^{\otimes 2} |L_1\rangle$ by the unitary

$$U = \frac{1}{\sqrt{3}} \begin{pmatrix} \frac{1}{2}(1+\sqrt{3}) & \frac{1}{2}(1-\sqrt{3}) & 1 \\ 1 & 1 & -1 \\ \frac{1}{2}(1-\sqrt{3}) & \frac{1}{2}(1+\sqrt{3}) & 1 \end{pmatrix}.$$

Another example of LU equivalent RLS involving Bell pairs consists of

$$\begin{cases} L_3 = 11 \cup 22, \\ L_4 = 12 \cup 21, \end{cases}$$

whose minimal DFAs have size 2 as shown below, and therefore the bond dimension does not change in this case,



They are related as $|L_4\rangle = U^{\otimes 2} |L_3\rangle$ by $U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}$.

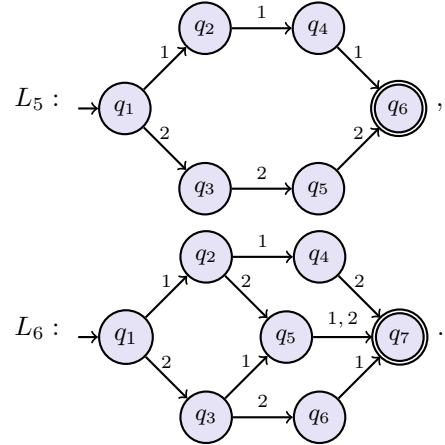
Finally, we give an example involving the GHZ on three particles, where the minimal DFA of a language that is SLOCC-equivalent to it has a different number of internal states,

$$\begin{cases} L_5 = 111 \cup 222, \\ L_6 = 112 \cup 121 \cup 211 \cup 221 \cup 212 \cup 122, \end{cases}$$

related as $|L_6\rangle = P^{\otimes 3} |L_5\rangle$ where

$$P = - \begin{pmatrix} \frac{i}{3^{1/6}} & (-\frac{1}{3})^{1/6} \\ (-\frac{1}{3})^{1/6} & \frac{i}{3^{1/6}} \end{pmatrix}.$$

Their minimal DFAs have bond dimensions 6 and 7, respectively, as shown below,



H Generalization of RLS to 2D regular pictures

For completeness, we provide here more details on the generalization of RLs to two dimensions following the notation in [32], and their explicit expression in terms of PEPS.

A *picture* can be defined as a 2D rectangular array of elements of an alphabet Σ . A *2D language* is then defined as a subset of Σ^{**} , where Σ^{**} denotes the set of all pictures over Σ . 1D RL's thus correspond to pictures with a single row or column.

Definition (4.2 in [32]). A (non-deterministic) 2D online tessellation automata (OTA) is defined by $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$, where Q is the set of internal states, Σ is the input alphabet, q_0 is the initial state, F is the set of accepting states, and $\delta : Q \times Q \times \Sigma \rightarrow \mathcal{P}(Q)$ is the transition function.

A run of $\mathcal{A}$ on picture $p \in \Sigma^{**}$ consists of associating a state from Q to each element (i, j) of p . The state is determined by the transition function δ which receives as inputs the states already associated to positions $(i-1, j)$ and $(i, j-1)$, and the symbol $p(i, j)$. At time $t = 0$, the internal states of the first row and the first column are set to the initial state indicated by q_0 .

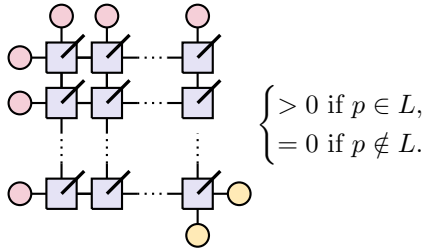
Then, we say that an OTA $\mathcal{A}$ recognizes or accepts a picture p of size $(l_1(p), l_2(p))$ denoting its number of rows and columns, respectively, if there exists a run of $\mathcal{A}$ on p such that the state associated to position $(l_1(p), l_2(p))$ is an accepting state belonging to F .

In fact, we can express the above definition of OTA as a PEPS, with physical dimension equal to $|\Sigma|$, bond dimension $D = |Q|$, and the following tensors:

$$\begin{array}{c} x \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ z \end{array} \begin{array}{c} a \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ z \end{array} = \begin{cases} 1 & \text{if } z \in \delta(x, y, a), \\ 0 & \text{otherwise,} \end{cases}$$

$$-\text{pink circle} = |q_0\rangle, \quad -\text{yellow circle} = \sum_{f \in F} |f\rangle,$$

so that the PEPS evaluated on picture p is



By construction, all these states satisfy an area law, with the entanglement upper bounded by $|\partial A| \log |Q|$, $|\partial A|$ being the number of virtual bonds cut by the boundary of the considered region A .

As an example, we provide the explicit representation of a 2D language over $\Sigma = \{a\}$ consisting of all pictures with an odd number of columns. An OTA that recognizes it is given by (4.3 in [32]):

- $Q = \{0, 1, 2\}$,
- $q_0 = \{0\}$,
- $F = \{1\}$,
- $\delta(0, 0, a) = \delta(0, 2, a) = \delta(1, 0, a) = \delta(1, 2, a) = 1$;
 $\delta(0, 1, a) = \delta(2, 1, a) = 2$.

The equivalent PEPS representation has $d = 1$, $D = 3$, and tensors

$$\begin{array}{c} 0 \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 1 \end{array} \begin{array}{c} a \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 1 \end{array} = \begin{array}{c} 0 \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 1 \end{array} \begin{array}{c} a \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 1 \end{array} = \begin{array}{c} 1 \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 1 \end{array} \begin{array}{c} a \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 1 \end{array} = 1$$

$$\begin{array}{c} 0 \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 2 \end{array} \begin{array}{c} a \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 2 \end{array} = \begin{array}{c} 1 \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 2 \end{array} \begin{array}{c} a \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ 2 \end{array} = 1$$

$$-\text{pink circle} = |0\rangle, \quad -\text{yellow circle} = |1\rangle.$$

References

- [1] Alfred V. Aho. Chapter 5 - algorithms for finding patterns in strings. In Jan Van Leeuwen, editor, *Algorithms and Complexity*, Handbook of Theoretical Computer Science, pages 255–300. Elsevier, Amsterdam, 1990. ISBN 978-0-444-88071-0. DOI: [10.1016/B978-0-444-88071-0.50010-2](https://doi.org/10.1016/B978-0-444-88071-0.50010-2).
- [2] Alfred V. Aho and Jeffrey D. Ullman. *The theory of parsing, translation, and compiling*. Prentice-Hall, Inc., USA, 1972. ISBN 0139145567.
- [3] Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: principles, techniques, and tools*. Always learning. Pearson, 2014. ISBN 978-1-292-02434-9.
- [4] Andris Ambainis and Abuzer Yakaryilmaz. Automata and quantum computing, 2018. URL <https://arxiv.org/abs/1507.01988>.
- [5] Anselmo, Marcella, Giammarresi, Dora, Madonia, Maria, and Restivo, Antonio. Unambiguous recognizable two-dimensional languages. *RAIRO-Theor. Inf. Appl.*, 40(2):277–293, 2006. DOI: [10.1051/ita:2006008](https://doi.org/10.1051/ita:2006008).
- [6] Stéphane Ayache, Rémi Eyraud, and Noé Goudian. Explaining black boxes on sequential data using weighted automata. In Olgierd Unold, Witold Dyrka, and Wojciech Wic-zorek, editors, *Proceedings of The 14th International Conference on Grammatical Inference 2018*, volume 93 of *Proceedings of Machine Learning Research*, pages 81–103. PMLR, feb 2019. URL <https://proceedings.mlr.press/v93/ayache19a.html>.
- [7] Ivan Baburin and Ryan Cotterell. An analysis of on-the-fly determinization of finite-state automata, 2023. URL <https://arxiv.org/abs/2308.14077>.
- [8] Armando Bellante, Reinis Irmejs, Marta Florido-Llinàs, María Cea Fernández, Marianna Crupi, Matthew Kiser, and J. Ignacio Cirac. Compiling quantum regular language states, 2026. URL <https://arxiv.org/abs/2602.02698>.
- [9] L. Berman and J. Hartmanis. On isomorphisms and density of np and other complete sets. *SIAM Journal on Computing*, 6(2):305–322, 1977. DOI: [10.1137/0206023](https://doi.org/10.1137/0206023).

- [10] Jean Berstel and Christophe Reutenauer. *Non-commutative Rational Series with Applications*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2010.
- [11] Amandeep Singh Bhatia and Ajay Kumar. Quantum finite automata: survey, status and research directions, 2019. URL <https://arxiv.org/abs/1901.07992>.
- [12] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009. ISSN 1745-2481. DOI: [10.1038/nphys1157](https://doi.org/10.1038/nphys1157).
- [13] Janusz A. Brzozowski. Towards a theory of complexity of regular languages, 2017. URL <https://arxiv.org/abs/1702.05024>.
- [14] Xie Chen, Zheng-Cheng Gu, and Xiao-Gang Wen. Classification of gapped symmetric phases in one-dimensional spin systems. *Phys. Rev. B*, 83:035107, Jan 2011. DOI: [10.1103/PhysRevB.83.035107](https://doi.org/10.1103/PhysRevB.83.035107).
- [15] Noam Chomsky and George A. Miller. Finite state languages. *Information and Control*, 1(2):91–112, 1958. DOI: [10.1016/S0019-9958\(58\)90082-2](https://doi.org/10.1016/S0019-9958(58)90082-2).
- [16] J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003, Dec 2021. DOI: [10.1103/RevModPhys.93.045003](https://doi.org/10.1103/RevModPhys.93.045003).
- [17] Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. On the computation of some standard distances between probabilistic automata. In Oscar H. Ibarra and Hsu-Chun Yen, editors, *Implementation and Application of Automata*, pages 137–149. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-37214-1. URL https://link.springer.com/chapter/10.1007/11812128_14.
- [18] Gregory M. Crosswhite and Dave Bacon. Finite automata for caching in matrix product algorithms. *Phys. Rev. A*, 78:012356, Jul 2008. DOI: [10.1103/PhysRevA.78.012356](https://doi.org/10.1103/PhysRevA.78.012356).
- [19] R. H. Dicke. Coherence in spontaneous radiation processes. *Phys. Rev.*, 93:99–110, Jan 1954. DOI: [10.1103/PhysRev.93.99](https://doi.org/10.1103/PhysRev.93.99).
- [20] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Phys. Rev. A*, 62:062314, Nov 2000. DOI: [10.1103/PhysRevA.62.062314](https://doi.org/10.1103/PhysRevA.62.062314).
- [21] Flavio D’Alessandro, Benedetto Intrigila, and Stefano Varricchio. On the structure of the counting function of sparse context-free languages. *Theoretical Computer Science*, 356(1):104–117, 2006. ISSN 0304-3975. DOI: [10.1016/j.tcs.2006.01.037](https://doi.org/10.1016/j.tcs.2006.01.037).
- [22] Manuel Erhard, Mario Krenn, and Anton Zeilinger. Advances in high-dimensional quantum entanglement. 2(7):365–381. ISSN 2522-5820. DOI: [10.1038/s42254-020-0193-5](https://doi.org/10.1038/s42254-020-0193-5).
- [23] G. Evenbly and G. Vidal. Tensor network states and geometry. *Journal of Statistical Physics*, 145(4):891–918, 2011. DOI: [10.1007/s10955-011-0237-4](https://doi.org/10.1007/s10955-011-0237-4).
- [24] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Communications in Mathematical Physics*, 144(3):443–490, 1992. ISSN 1432-0916. DOI: [10.1007/BF02099178](https://doi.org/10.1007/BF02099178).
- [25] Douglas R. Farenick. *Algebras of linear transformations*. Universitext. Springer, 2001. ISBN 978-0-387-95062-4.
- [26] Henning Fernau, Stefan Hoffmann, and Michael Wehar. Finite automata intersection non-emptiness: Parameterized complexity revisited, 2021. URL <https://arxiv.org/abs/2108.05244>.
- [27] Marta Florido-Llinàs, Álvaro M. Alhambra, David Pérez-García, and J. Ignacio Cirac. Uniform matrix product states with a boundary, 2025. URL <https://arxiv.org/abs/2512.11968>.
- [28] Jeffrey EF Friedl. *Mastering regular expressions*. O’Reilly Media, Inc., 2006. ISBN 978-0596528126.
- [29] Ángel J. Gallego and Román Orús. Language design as information renormalization. *SN Computer Science*, 3(2):140. ISSN 2661-8907. DOI: [10.1007/s42979-021-01002-y](https://doi.org/10.1007/s42979-021-01002-y).
- [30] José Garre-Rubio, Alex Turzillo, and András Molnár. Mps stability and the intersection property, 2025. URL <https://arxiv.org/abs/2501.17109>.
- [31] Paweł Gawrychowski, Dalia Krieger, Narad Rampersad, and Jeffrey Shallit. Finding the growth rate of a regular or context-free language in polynomial time. *International Journal of Foundations of Computer Science*, 21(04):597–618, 2010. DOI: [10.1142/S0129054110007441](https://doi.org/10.1142/S0129054110007441).
- [32] Dora Giammarresi and Antonio Restivo. *Two-Dimensional Languages*, pages 215–267. Springer Berlin Heidelberg, 1997. ISBN 978-3-642-59126-6. DOI: [10.1007/978-3-642-59126-6_4](https://doi.org/10.1007/978-3-642-59126-6_4).
- [33] Lei Gioia, Sanjay Moudgalya, and Olexei I. Motrunich. Distinct types of parent hamiltonians for quantum states: Insights from the w state as a quantum many-body scar, 2025. URL <https://arxiv.org/abs/2510.24713>.
- [34] Ivan Glasser, Ryan Sweke, Nicola Pancotti, Jens Eisert, and J. Ignacio Cirac. *Expressive power of tensor-network factorizations for probabilistic modeling*. Curran Associates Inc., Red Hook, NY, USA, 2019. DOI: [10.5555/3454287.3454421](https://doi.org/10.5555/3454287.3454421).
- [35] Sarang Gopalakrishnan. Push-down automata as

- sequential generators of highly entangled states, jan 2025.
- [36] Daniel Gottesman. Stabilizer codes and quantum error correction, 1997. URL <https://arxiv.org/abs/quant-ph/9705052>.
 - [37] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. *Going Beyond Bell's Theorem*, pages 69–72. Springer Netherlands, Dordrecht, 1989. ISBN 978-94-017-0849-4. DOI: [10.1007/978-94-017-0849-4_10](https://doi.org/10.1007/978-94-017-0849-4_10).
 - [38] Jutho Haegeman, Bogdan Pirvu, David J. Weir, J. Ignacio Cirac, Tobias J. Osborne, Henri Verschelde, and Frank Verstraete. Variational matrix product ansatz for dispersion relations. *Phys. Rev. B*, 85:100408(R), Mar 2012. DOI: [10.1103/PhysRevB.85.100408](https://doi.org/10.1103/PhysRevB.85.100408).
 - [39] J. Hartmanis and S. R. Mahaney. An essay about research on sparse np complete sets. In P. Dembiński, editor, *Mathematical Foundations of Computer Science 1980*, pages 40–57. Springer Berlin Heidelberg, 1980. ISBN 978-3-540-38194-5. DOI: [10.1007/BFb0022494](https://doi.org/10.1007/BFb0022494).
 - [40] M. Hein, J. Eisert, and H. J. Briegel. Multi-party entanglement in graph states. *Phys. Rev. A*, 69:062311, Jun 2004. DOI: [10.1103/PhysRevA.69.062311](https://doi.org/10.1103/PhysRevA.69.062311).
 - [41] Stefan Hoffmann. Computational complexity of synchronization under sparse regular constraints. In Evripidis Bampis and Aris Pagourtzis, editors, *Fundamentals of Computation Theory*, pages 272–286, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86593-1. DOI: [10.1007/978-3-030-86593-1_19](https://doi.org/10.1007/978-3-030-86593-1_19).
 - [42] John Hopcroft. An $n \log n$ algorithm for minimizing states in a finite automaton. In Zvi Kohavi and Azaria Paz, editors, *Theory of Machines and Computations*, pages 189–196. Academic Press, 1971. ISBN 978-0-12-417750-5. DOI: [10.1016/B978-0-12-417750-5.50022-1](https://doi.org/10.1016/B978-0-12-417750-5.50022-1).
 - [43] John E. Hopcroft and R. M. Karp. A linear algorithm for testing equivalence of finite automata. Technical report, University of California, 1971. URL <https://hdl.handle.net/1813/5958>.
 - [44] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65, 2001.
 - [45] Katsushi Inoue and Akira Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13(2):95–121, 1977. ISSN 0020-0255. DOI: [10.1016/0020-0255\(77\)90023-8](https://doi.org/10.1016/0020-0255(77)90023-8).
 - [46] Galina Jirásková and Alexander Okhotin. State complexity of cyclic shift. *RAIRO - Theoretical Informatics and Applications*, 42(2):335–360, 2008. DOI: [10.1051/ita:2007038](https://doi.org/10.1051/ita:2007038).
 - [47] Sampath Kannan, Z. Sweedyk, and Steve Mahaney. Counting and random generation of strings in regular languages. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, page 551–557, USA, 1995. Society for Industrial and Applied Mathematics. ISBN 0898713498. DOI: [10.5555/313651.313803](https://doi.org/10.5555/313651.313803).
 - [48] Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. Language equivalence for probabilistic automata. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification*, pages 526–540. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-22110-1. DOI: [10.1007/978-3-642-22110-1_42](https://doi.org/10.1007/978-3-642-22110-1_42).
 - [49] S. C. Kleene. *Representation of Events in Nerve Nets and Finite Automata*, pages 3–42. Princeton University Press, Princeton, 1956. ISBN 9781400882618. DOI: [10.1515/9781400882618-002](https://doi.org/10.1515/9781400882618-002).
 - [50] B. Kraus. Local unitary equivalence of multipartite pure states. *Phys. Rev. Lett.*, 104:020504, Jan 2010. DOI: [10.1103/PhysRevLett.104.020504](https://doi.org/10.1103/PhysRevLett.104.020504).
 - [51] Kenneth Krohn and John Rhodes. Algebraic theory of machines. i. prime decomposition theorem for finite semigroups and machines. 116(0):450–464. ISSN 0002-9947, 1088-6850. DOI: [10.1090/S0002-9947-1965-0188316-1](https://doi.org/10.1090/S0002-9947-1965-0188316-1).
 - [52] Hing Leung. Descriptive complexity of nfa of different ambiguity. *International Journal of Foundations of Computer Science*, 16(05):975–984, 2005. DOI: [10.1142/S0129054105003418](https://doi.org/10.1142/S0129054105003418).
 - [53] Dafa Li. Stochastic local operations and classical communication (SLOCC) and local unitary operations (LU) classifications of n qubits via ranks and singular values of the spin-flipping matrices. *Quantum Information Processing*, 17(6):132. ISSN 1573-1332. DOI: [10.1007/s11128-018-1900-3](https://doi.org/10.1007/s11128-018-1900-3).
 - [54] Tianyu Li, Doina Precup, and Guillaume Rabusseau. Connecting weighted automata, tensor networks and recurrent neural networks through spectral learning. *Machine Learning*, 113(5):2619–2653, 2024. ISSN 1573-0565. DOI: [10.1007/s10994-022-06164-1](https://doi.org/10.1007/s10994-022-06164-1).
 - [55] Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=De4FYqjFueZ>.
 - [56] Mateusz Michałek and Yaroslav Shitov. Quantum version of wielandt's inequality revisited. *IEEE Transactions on Information Theory*, 65(8):5239–5242, 2019. DOI: [10.1109/TIT.2019.2897772](https://doi.org/10.1109/TIT.2019.2897772).
 - [57] Ashwin Nayak and Dave Touchette. Augmented index and quantum streaming algorithms for

- dyck(2), 2022. URL <https://arxiv.org/abs/1610.04937>.
- [58] Takamasa Okudono, Masaki Waga, Taro Sekiyama, and Ichiro Hasuo. Weighted automata extraction from recurrent neural networks via regression on state spaces. 34(4): 5306–5314. ISSN 2374-3468, 2159-5399. DOI: [10.1609/aaai.v34i04.5977](https://doi.org/10.1609/aaai.v34i04.5977).
- [59] Román Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, 2019. ISSN 2522-5820. DOI: [10.1038/s42254-019-0086-7](https://doi.org/10.1038/s42254-019-0086-7).
- [60] Alexandros Palioudakis, Kai Salomaa, and Selim G. Akl. State complexity and limited non-determinism. In Martin Kutrib, Nelma Moreira, and Rogério Reis, editors, *Descriptive Complexity of Formal Systems*, pages 252–265, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-31623-4. DOI: [10.1007/978-3-642-31623-4_20](https://doi.org/10.1007/978-3-642-31623-4_20).
- [61] Jian-Wei Pan, Zeng-Bing Chen, Chao-Yang Lu, Harald Weinfurter, Anton Zeilinger, and Marek Żukowski. Multiphoton entanglement and interferometry. *Rev. Mod. Phys.*, 84:777–838, May 2012. DOI: [10.1103/RevModPhys.84.777](https://doi.org/10.1103/RevModPhys.84.777).
- [62] Christopher J Pappacena. An upper bound for the length of a finite-dimensional algebra. *Journal of Algebra*, 197(2):535–545, 1997. ISSN 0021-8693. DOI: [10.1006/jabr.1997.7140](https://doi.org/10.1006/jabr.1997.7140).
- [63] D. Perez-Garcia, F. Verstraete, M.M. Wolf, and J.I. Cirac. Matrix product state representations. *Quantum Information and Computation*, 7(5): 401, 2007. DOI: [10.26421/QIC7.5-6-1](https://doi.org/10.26421/QIC7.5-6-1).
- [64] Luca Pezzè, Augusto Smerzi, Markus K. Oberthaler, Roman Schmied, and Philipp Treutlein. Quantum metrology with nonclassical states of atomic ensembles. *Rev. Mod. Phys.*, 90:035005, Sep 2018. DOI: [10.1103/RevModPhys.90.035005](https://doi.org/10.1103/RevModPhys.90.035005).
- [65] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959. DOI: [10.1147/rd.32.0114](https://doi.org/10.1147/rd.32.0114).
- [66] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001. DOI: [10.1103/PhysRevLett.86.5188](https://doi.org/10.1103/PhysRevLett.86.5188).
- [67] Bala Ravikumar and Oscar H. Ibarra. Relating the type of ambiguity of finite automata to the succinctness of their representation. *SIAM Journal on Computing*, 18(6):1263–1282, 1989. DOI: [10.1137/0218083](https://doi.org/10.1137/0218083).
- [68] Paul M. Riechers and Thomas J. Elliott. Identifiability and minimality bounds of quantum and post-quantum models of classical stochastic processes, 2025. URL <https://arxiv.org/abs/2509.03004>.
- [69] Shinsei Ryu and Tadashi Takayanagi. Holographic derivation of entanglement entropy from the anti-de sitter space/conformal field theory correspondence. *Phys. Rev. Lett.*, 96: 181602, May 2006. DOI: [10.1103/PhysRevLett.96.181602](https://doi.org/10.1103/PhysRevLett.96.181602).
- [70] Subir Sachdev. *Quantum Phase Transitions*. Cambridge University Press, 2 edition, 2011. DOI: [10.1017/CBO9780511973765](https://doi.org/10.1017/CBO9780511973765).
- [71] Jacques Sakarovitch. *Elements of automata theory*. Cambridge University Press, 2009. DOI: [10.1017/CBO9781139195218](https://doi.org/10.1017/CBO9781139195218).
- [72] David Sauerwein, Andras Molnar, J. Ignacio Cirac, and Barbara Kraus. Matrix product states: Entanglement, symmetries, and state transformations. *Phys. Rev. Lett.*, 123:170504, Oct 2019. DOI: [10.1103/PhysRevLett.123.170504](https://doi.org/10.1103/PhysRevLett.123.170504).
- [73] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011. ISSN 0003-4916. DOI: [10.1016/j.aop.2010.09.012](https://doi.org/10.1016/j.aop.2010.09.012).
- [74] Norbert Schuch, David Pérez-García, and Ignacio Cirac. Classifying quantum phases using matrix product states and projected entangled pair states. *Phys. Rev. B*, 84:165139, Oct 2011. DOI: [10.1103/PhysRevB.84.165139](https://doi.org/10.1103/PhysRevB.84.165139).
- [75] M.P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2):245–270, 1961. ISSN 0019-9958. DOI: [10.1016/S0019-9958\(61\)80020-X](https://doi.org/10.1016/S0019-9958(61)80020-X).
- [76] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995. DOI: [10.1103/PhysRevA.52.R2493](https://doi.org/10.1103/PhysRevA.52.R2493).
- [77] Michael Sipser. *Introduction to the theory of computation*. Boston : PWS Pub. Co., 1997. ISBN 978-0-534-94728-6. DOI: [10.2307/2586778](https://doi.org/10.2307/2586778).
- [78] R. E. Stearns and H. B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing*, 14(3):598–611, 1985. DOI: [10.1137/0214044](https://doi.org/10.1137/0214044).
- [79] Andrew Szilard, Sheng Yu, Kaizhong Zhang, and Jeffrey Shallit. Characterizing regular languages with polynomial densities. In Ivan M. Havel and Václav Koubek, editors, *Mathematical Foundations of Computer Science 1992*, pages 494–503. Springer Berlin Heidelberg, 1992. ISBN 978-3-540-47291-9. DOI: [10.1007/3-540-55808-X_48](https://doi.org/10.1007/3-540-55808-X_48).
- [80] Barbara M. Terhal. Quantum error correction for quantum memories. *Rev. Mod. Phys.*, 87: 307–346, Apr 2015. DOI: [10.1103/RevModPhys.87.307](https://doi.org/10.1103/RevModPhys.87.307).
- [81] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd. Gaussian quantum information. *Rev. Mod. Phys.*,

- 84:621–669, May 2012. DOI: [10.1103/RevModPhys.84.621](https://doi.org/10.1103/RevModPhys.84.621).
- [82] Gail Weiss, Yoav Goldberg, and Eran Yahav. Learning deterministic weighted automata with queries and counterexamples. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. DOI: [10.5555/3454287.3455055](https://doi.org/10.5555/3454287.3455055).
 - [83] Sheng Yu. *Regular Languages*, pages 41–110. Springer Berlin Heidelberg, 1997. ISBN 978-3-642-59136-5. DOI: [10.1007/978-3-642-59136-5_2](https://doi.org/10.1007/978-3-642-59136-5_2).
 - [84] Sheng Yu, Qingyu Zhuang, and Kai Salomaa. The state complexities of some basic operations on regular languages. *Theoretical Computer Science*, 125(2):315–328, 1994. ISSN 0304-3975. DOI: [10.1016/0304-3975\(92\)00011-F](https://doi.org/10.1016/0304-3975(92)00011-F).
 - [85] Bei Zeng, Xie Chen, Duan-Lu Zhou, and Xiao-Gang Wen. *Quantum Information Meets Quantum Matter: From Quantum Entanglement to Topological Phases of Many-Body Systems*. Quantum Science and Technology. Springer, 2019. ISBN 978-1-4939-9082-5 978-1-4939-9084-9. DOI: [10.1007/978-1-4939-9084-9](https://doi.org/10.1007/978-1-4939-9084-9).
 - [86] Wucheng Zhang. Sequential generation of two-dimensional super-area-law states with local parent hamiltonian. *PRX Quantum*, Oct 2025. DOI: [10.1103/t7py-glgv](https://doi.org/10.1103/t7py-glgv).
 - [87] Yifan Zhang and Edgar Solomonik. On stability of tensor networks and canonical forms, 2020. URL <https://arxiv.org/abs/2001.01191>.