Quantum SAT Problems with Finite Sets of Projectors are Complete for a Plethora of Classes

Ricardo Rivera Cardoso[1,*], Alex Meiburg[2,3], and Daniel Nagaj[1]

[1] *RCQI, Institute of Physics, Slovak Academy of Sciences, Bratislava, Slovakia*
[2] *Perimiter Institute for Theoretical Physics, Waterloo, Canada*
[3] *Institute for Quantum Computing, University of Waterloo*

Abstract

Prior to this work, all known variations of the *Quantum Satisfiability* (QSAT) problem—consisting of determining whether a k-local (k-body) Hamiltonian is frustration-free—could be classified as being either in P, NP-complete, MA-complete, or QMA_1-complete. Problems in the latter three classes are not known to be solvable efficiently. Here, we demonstrate new variations of this problem that are complete for BQP_1, coRP, QCMA, $\mathsf{PI}(\mathsf{coRP}, \mathsf{NP})$, $\mathsf{PI}(\mathsf{BQP}_1, \mathsf{NP})$, $\mathsf{PI}(\mathsf{BQP}_1, \mathsf{MA})$, $\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP})$, $\mathsf{SoPU}(\mathsf{BQP}_1, \mathsf{NP})$, and $\mathsf{SoPU}(\mathsf{BQP}_1, \mathsf{MA})$. Given that the first two classes refer to problems efficiently solvable with a quantum or probabilistic classical computer, we show new kinds of efficiently solvable QSAT problems. Our result also implies that a future complete classification for quantum constraint satisfaction problems (QCSPs), analogous to Zhuk's dichotomy theorem for classical CSPs, must either include these 13 classes, or otherwise show that some of them are equal. Furthermore, we present the first nontrivial problem known to be complete for BQP_1.

We prove there are QSAT problems complete for BQP_1, coRP, and QCMA by restricting the allowed sets of Hamiltonians to consist of elements similar to H_{init}, H_{prop}, and H_{out}, seen in the circuit-to-Hamiltonian transformation. Usually, these are used to demonstrate hardness of QSAT and *Local Hamiltonian* problems, and so our proofs of hardness are simple. We modify these terms to involve high-dimensional data and clock qudits, ternary logic, and either monogamy of entanglement or specific clock encodings to ensure that all Hamiltonians generated with a polynomial number of these three elements can be decided in their respective classes. The remaining six problems are obtained by considering "sums" and "products" of the QSAT problems mentioned here. Before this work, the QSAT problems generated in this way resulted in complete problems for PI and SoPU classes that were trivially equal to NP, MA, or QMA_1. We thus commence the study of these new and seemingly nontrivial classes.

While [Meiburg, 2021] first sought to prove completeness for coRP, BQP_1, and QCMA, we note that his constructions are flawed, leading to incorrect proofs of these statements. Here, we rework these constructions, as well as obtain improvements on the required qudit dimensionality.

[*] ricardo.rivera@savba.sk

1 Introduction

Many of the interesting and puzzling phenomena in many-body physics occur in the ground state. One way to study quantum systems in this state is through their ground state energy, as it can provide information about some of the physical and chemical properties of the system. It is thus of great interest to calculate or even estimate this quantity. This task is embodied by the k-Local Hamiltonian (k-LH) problem. Specifically, given a *k-local* (k-body) Hamiltonian—an operator of the form $H = \sum_i h_i$ where each h_i acts on at most k qubits—and two numbers $a, b \in \mathbb{R}$ with $b - a \geq 1/poly(n)$, this problem consists of distinguishing between the cases where H has an eigenvalue less than a or greater than b. Kitaev [1] showed that k-LH with $k \geq 5$ (and later improved to $k \geq 2$ [2]) is unlikely to be decided efficiently with a classical or quantum computer. In complexity theory terms, k-LH with $k \geq 2$ is QMA-complete.[1]

The LH problem is considered a "weak" quantum constraint satisfaction problem (QCSP) as states with energy less than a do not necessarily minimize the energy of each h_i. For this reason, LH is often compared to MAX-k-SAT instead of the "strong" CSP k-SAT. Due to the immense importance of SAT in classical complexity and other hard sciences, Bravyi [3] defined the Quantum k-SAT (k-QSAT) problem. Given a set of k-local projectors (also referred as *clauses* or *constraints*) and a number $b \in \mathbb{R}$, this problem consists of distinguishing between the cases where there exists a state that simultaneously lies in the null space of all projectors, or for all states, the penalty incurred by violations of the constraints is greater than b.[2] Bravyi showed that 2-QSAT on qubits is in P while k-QSAT with $k \geq 4$ (and later improved to $k \geq 3$ [4]) is QMA_1-complete when using the Clifford+T gate set $\mathcal{G}_8 = \{H, \mathrm{CNOT}, T\}$.[3]

Interestingly, these two problems have in common that they are in P for a certain k but appear to become much harder for $k+1$: LH is in P for $k \leq 1$ and becomes QMA-complete for $k > 1$, while QSAT is in P for $k \leq 2$ and $\mathsf{QMA}_1^{\mathcal{G}_8}$-complete for $k > 2$. This is not entirely surprising since the Hamiltonians considered in the problems have no restriction other than their locality, and perhaps the difficulty lies in deciding "unphysical" Hamiltonians. Following this line of thought, others have considered variations of these problems where the h_i are drawn from more realistic and relevant sets $\mathcal{S}$ that satisfy some property or correspond to a physical model. To name a few, these may be stoquastic [6], commuting [7], fermionic [8], bosonic [9], or from models like the Heisenberg [10] and Bose-Hubbard [11]. In addition, one might also consider placing restrictions on the geometry of the problem [12, 13, 14, 15, 16].

In a landmark result, Cubitt and Montanaro [17] showed that any LH problem where the h_i are drawn from a finite set of at most 2-local qubit Hermitian matrices can be classified as being either in P, NP-complete, StoqMA-complete, or QMA-complete.[4] As decision problems in the latter three classes are not known to be efficiently solvable in either classical or quantum computers, they showed that the only Hamiltonians of this type for which the LH problem can be solved efficiently are those with only 1-local terms. This is significant, as many relevant Hamiltonians in nature can be approximated by 2-local Hamiltonians of this type (e.g. all those supported on Pauli operators like Heisenberg and Ising spin glass models), and it is then likely that estimating their ground state energy efficiently lies outside of reach. Moreover, their result has led to a much larger repertoire of problems from which to construct reductions and potentially show the complexity of other computational problems, as well as its connection to universal quantum Hamiltonians for analogue Hamiltonian simulation [19, 20].

For QSAT, we don't have a similar classification theorem for a given interaction locality, mapping different choices of clauses/projectors to different complexities. This is perhaps because we know that 2-QSAT is easy (in P) without a restriction on the clauses, and 3-QSAT is already hard (QMA_1-complete). As the main research motivation was to find which of the simple variants of QSAT is already hard, this was de facto "mission accomplished" for QSAT. Therefore, prior to this work, we knew QSAT problems, with clauses drawn from a finite set $\mathcal{S}$ of $\mathcal{O}(1)$-local qudit (with $d = \mathcal{O}(1)$) projectors, that could classified as being either

[1] The class QMA can be thought of as the quantum analogue of NP, or more accurately MA since the class has probabilistic acceptance and rejection.

[2] Alternatively, this problem can be defined with local Hamiltonians instead of projectors, in which case, the problem is equivalent to determining whether the Hamiltonian is frustration-free.

[3] QMA_1 is the one-sided error variation of QMA with perfect completeness, i.e. instances for which the answer is "yes" (in this case frustration-free Hamiltonians) are accepted with certainty. The notation $\mathcal{G}_8$ stems from Ref. [5] and denotes the Clifford-cyclotomic gate set of degree of 8. The reason why it is necessary to specify the gate set for classes with perfect completeness is discussed in Section 2.2.1.

[4] StoqMA is the class of problems equivalent to estimating the ground state energy of the transverse-field Ising model [18].

in P, NP-complete, MA-complete, or QMA_1-complete. The representative problems are 2-SAT, 3-SAT [21, 22], Stoquastic 6-SAT* [6], and 3-QSAT* [4].[5,6] However, it was not clear whether there exist variants of QSAT with other than these 4 possible complexities.

Moreover, from a physical perspective, QSAT has gathered less attention than LH since 2-QSAT can be decided classically. However, we remark that interesting developments in the fields of quantum chemistry [23], high energy physics [24] and nuclear physics [25, 26, 27] have shown that 3- or 4-local Hamiltonians are sometimes necessary to explain emergent physics. It would thus be exciting to determine if these Hamiltonians are frustration-free, or possibly identify new sets of Hamiltonians with $k > 2$ for which the QSAT problem can still be solved efficiently.

From a complexity theory perspective, QSAT may still play a significant role and help provide answers to several open questions. For example, whether perfect error-reduction is possible for QMA (i.e. whether $\mathsf{QMA}_1 = \mathsf{QMA}$), whether we could have perturbation theory gadgets that preserve perfect completeness, and whether we can establish a complete classification theorem for QSAT problems. This could result in new relations between complexity classes and be useful for constructing reductions from other problems of interest.

In this work we make progress in both directions. Our main result shows there are 9 new k-QSAT problems with $k = \mathcal{O}(1)$ that are complete for BQP_1, coRP, and QCMA, $\mathsf{PI}(\mathsf{coRP}, \mathsf{NP})$, $\mathsf{PI}(\mathsf{BQP}_1, \mathsf{NP})$, $\mathsf{PI}(\mathsf{BQP}_1, \mathsf{MA})$, $\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP})$, $\mathsf{SoPU}(\mathsf{BQP}_1, \mathsf{NP})$, and $\mathsf{SoPU}(\mathsf{BQP}_1, \mathsf{MA})$. Together with the problems that are complete for P, NP, MA, and QMA_1-complete, our results imply that a complete classification for strong QCSPs must either include these 13 classes, or otherwise demonstrate that some of these are equal. We show that there are several interesting nontrivial relationships within these classes, and so a classification theorem demonstrating that there are fewer than these 13 classes, could present exciting results. We thus motivate the search for such a theorem. As a corollary of our result, we show that there are QSAT problems with $k > 2$ that can be solved efficiently, as well as the first nontrivial problem not known to be complete for BQP_1. Finally, as we discuss further in the text, the classes $\mathsf{PI}(\mathsf{A}, \mathsf{B})$ and $\mathsf{SoPU}(\mathsf{A}, \mathsf{B})$ are rarely mentioned in literature, and in the few occasions they have been considered, it has been for different A and B than the ones shown here [28, 29]. Our work then initiates the study of these classes, providing complete problems for each one.

1.1 Summary of results

The notation used here is given in Section 2.1. Our main result establishes that the QSAT problem SLCT-QSAT is $\mathsf{BQP}_1^{\mathcal{G}_8}$-complete. However, as the construction and analysis of this problem is contrived, we first show that the simpler and less optimized version of this problem, LCT-QSAT, is also complete for this class.

Theorem 1. *The problem* Linear-Clock-Ternary-QSAT (LCT-QSAT; Definition 3.1) *with* 4*-local clauses acting on* 17*-dimensional qudits is* $\mathsf{BQP}_1^{\mathcal{G}_8}$*-complete.*

An interesting feature of this problem, and one that may be of independent interest, is that this problem makes clever use of the principle of monogamy of entanglement to strongly constrain the structure of input instances, facilitating the task of deciding whether they are frustration-free.[7] Unfortunately, this trick comes at a price of high qudit dimensionality. Our main result shows that by relaxing the constraint on the instance's structure and instead study the instances more closely, we can obtain a similar problem with the same complexity but with reduced qudit dimensionality.

Theorem 2. *The problem* Semilinear-Clock-Ternary-QSAT (SLCT-QSAT; Definition 4.1) *with* 4*-local clauses acting on* 6*-dimensional qudits is* $\mathsf{BQP}_1^{\mathcal{G}_8}$*-complete.*

Recently, among many other interesting results, Rudolph [31] demonstrated that $\mathsf{BQP}_1^{\mathcal{G}_{2^i}} = \mathsf{BQP}_1^{\mathcal{G}_{2^j}}$ for any $i, j \in \mathbb{N}$ (Theorem 3.4). In other words, any problem in BQP_1 using a Clifford-cyclotomic gate set of degree 2^i can be perfectly simulated with one of degree 2^j for all $i, j \in \mathbb{N}$.

[5] We refer to k-SAT as a QSAT problem as all of its instances can be realized using diagonal projectors.

[6] As defined, Stoquastic 6-SAT and 3-QSAT allow an infinite set of projectors. We add an asterisk to refer to a particular version of the problem where the projectors are $\{H_{init}, H_{prop}, H_{out}\}$ for which the problem also remains complete for their respective classes.

[7] This construction is the most faithful to those considered by Meiburg in Ref. [30].

Corollary 1.1. *The problems* LCT-QSAT *and* SLCT-QSAT *are* BQP_1*-complete with any gate set* $\mathcal{G}_{2^l}$ *with* $l \in \mathbb{N}$.

Subsequently, by performing slight modifications to the clauses of SLCT-QSAT, we also obtain QCMA-complete and coRP-complete problems:

Theorem 3. *The problem* WITNESSED SLCT-QSAT (Definition 5.1) *with* 4*-local clauses acting on* 8*-dimensional qudits is* QCMA*-complete.*

Theorem 4. *The problem* CLASSICAL SLCT-QSAT (Definition 6.1) *with* 5*-local clauses acting on* 6*-dimensional qudits is* coRP*-complete.*

Which One? This:Finally, there is a notion of "sums" and "products" for both classical and quantum CSPs. We show that some sums and products of the QSAT problems above create new problems that are complete for an additional six classes.

Or: Finally, we consider the *direct product* "⊗" and *direct sum* "⊕" (see Definitions 8.1 and 8.3) of QSAT problems and show that there are six new problems that are complete for the *pairwise intersection of classes*, denoted PI(A, B), and the *star of pairwise union of classes* SoPU(A, B). Roughly, for any two classes A and B, PI(A, B) and SoPU(A, B) correspond to the sets of problems that can be expressed as the intersection and union of a problem in A and a problem in B.[8] We show:

Theorem 5. *Let* "⊗" *and* "⊕" *denote the* direct product *and* direct sum *for quantum constraint satisfaction problems (formally defined in Definitions 8.5 and 8.6). Pairwise combinations of the four* QSAT *problems—* 3-SAT*,* CLASSICAL SLCT-QSAT*,* SLCT-QSAT*, and* STOQUASTIC QSAT**—yield the following complete problems:*

1. CLASSICAL SLCT-QSAT ⊗ 3-SAT *is* PI(coRP, NP)*-complete.*
2. CLASSICAL SLCT-QSAT ⊕ 3-SAT *is* SoPU(coRP, NP)*-complete.*
3. SLCT-QSAT ⊗ 3-SAT *is* $\mathsf{PI}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{NP})$*-complete.*
4. SLCT-QSAT ⊕ 3-SAT *is* $\mathsf{SoPU}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{NP})$*-complete.*
5. SLCT-QSAT ⊗ STOQUASTIC 6-SAT* *is* $\mathsf{PI}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{MA})$*-complete.*
6. SLCT-QSAT ⊕ STOQUASTIC 6-SAT* *is* $\mathsf{SoPU}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{MA})$*-complete.*

The relationship between the 13 classes mentioned so far is shown in Fig. 1.

Given that the QSAT problems in Theorems 1–5 consist of finite sets of projects, and similarly 2-SAT, 3-SAT, STOQUASTIC 6-SAT*, and an 3-QSAT* which are respectively in P, NP-complete, MA-complete and $\mathsf{QMA}_1^{\mathcal{G}_8}$-complete our results imply that:

Corollary 1.2. *A complete classification theorem for strong QCSPs with* $\mathcal{O}(1)$*-local clauses must either include* 13 *classes or otherwise indicate that some of these are equal.*

1.2 Proof techniques

Here, we briefly discuss how we achieve the results above. We keep the discussion general since each QSAT problem differs slightly from the others, and we think that a more detailed explanation should be delayed after the preliminary section. The first subsection of each section discussing a QSAT problem provides this more detailed and complete overview.

In summary, we prove Theorems 1–4 by constructing QSAT problems restricted to consist of only three (four in the latter two problems) elements: H'_{init}, H'_{prop}, and H'_{out}, which are modifications of the projectors seen in the circuit-to-Hamiltonian transformation originating in Refs. [1, 32]. Normally, H_{init}, H_{prop}, and

[8] These classes are not to be confused with A ∩ B and A ∪ B. A ∩ B corresponds to the set of problems that are in both A and B, while A ∪ B corresponds to those that are in either A or B.

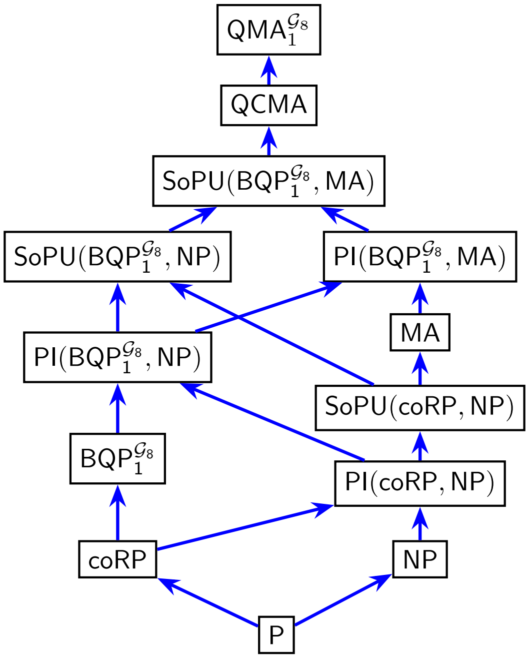

$\mathsf{QMA}_1^{\mathcal{G}_8}$
QCMA
$\mathsf{SoPU}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{MA})$
$\mathsf{SoPU}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{NP})$
$\mathsf{PI}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{MA})$
MA
$\mathsf{PI}(\mathsf{BQP}_1^{\mathcal{G}_8}, \mathsf{NP})$
$\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP})$
$\mathsf{BQP}_1^{\mathcal{G}_8}$
$\mathsf{PI}(\mathsf{coRP}, \mathsf{NP})$
coRP
NP
P

Figure 1: The classes for which we now have a complete strong QCSP, and their corresponding inclusions. In this work, we show completeness for quantum complexity classes with perfect completeness using the Clifford+T gate set $\mathcal{G}_8 = \{H, T, \text{CNOT}\}$. Rudolph's [31] result further strengthens ours by showing that $\mathsf{BQP}_1^{\mathcal{G}_8} = \mathsf{BQP}_1^{\mathcal{G}_{2^l}}$ for all $l \geq 1$. We discuss some of the inclusions in this figure in Section 2.2 and Section 8.

H_{out} are local operators used to encode the initialization, (unitary) propagation, and readout steps of a quantum circuit, respectively. In other words, one can encode the evaluation of a quantum circuit into the ground state of the local Hamiltonian $H = \sum_i (H_{init})_i + \sum_j (H_{prop})_j + \sum_k (H_{out})_k$. If the quantum circuit has perfect completeness, H is frustration-free and the zero-energy ground state is known as the (computational) *history state*. This transformation is hence useful to show that some LH and QSAT problems are hard for some complexity classes. Because our projectors, albeit slightly different, still implement these ideas, they can be used to show without much difficulty that the QSAT problems defined here are hard for coRP, BQP_1, and QCMA. The difficulty, and the reason why modifications are necessary, is to show that these problems are contained within the classes.

To achieve this, we would like that all input instances generated with a polynomial number of these three types of clauses encode the evaluation of a quantum circuit. Then, as just mentioned, verifying whether the instance is frustration-free simply depends on the readout. The algorithm that decides the instance then simply prepares the initial state, evaluates the circuit, and finally measures the readout qubits. If the circuit is a quantum circuit, this can be done in BQP_1. Similarly, if the circuit is a classical reversible circuit, this can be done in coRP. In practice, the instances do not generally resemble the encoding of a quantum circuit, but rather form a nonsensical web of constraints making it hard to tell whether all constraints can be satisfied. We therefore modify these clauses to enforce more structure in the instances, such that they may only be frustration-free if they have a "nice" structure. Some but not all might encode the evaluation of a circuit.

The two most relevant modifications of H_{init}, H_{prop}, and H_{out} are that we define them using high-dimensional data and clock qudits, which allows us to perform ternary logic on the data qudits and to enforce a linear clock structure within the input instances. For example, in the first QSAT problem mentioned, we allow the clock particles to have two additional 2-dimensional subspaces and demand that neighboring clock qudits should be maximally entangled. By the principle of monogamy of entanglement, it is evident that each clock qudit can be entangled with at most two other clock qudits. This results in the linear structure mentioned, and consequently a linear evaluation of the quantum circuit.

We prove Theorem 5 by We construct the QSAT problems that are complete for the PI and SoPU classes by considering "sums" and "products" of constraint satisfaction problems. Specifically, these are obtained by considering combinations of the 7 complete QSAT problems mentioned previously.

1.3 Discussion and open questions

In our first result, we show that Classical SLCT-QSAT and SLCT-QSAT are two QSAT problems with $k > 2$ that can be solved efficiently with a probabilistic or quantum computer, being the first problems of their kind. Unfortunately, the Hamiltonians used in these problems are artifacts built to achieve these results and do not immediately correspond to Hamiltonians of interest. It is unclear to us whether this can be achieved by modifying the constructions of these problems, however, one may still consider other tools to prove these results. We hope that having demonstrated that such problems exist, our results inspire others to search for physically relevant sets of Hamiltonians with $k > 3$ that may be efficiently solvable.

Another interesting observation about Classical SLCT-QSAT is that while the problem is defined with quantum Hamiltonians and has a highly entangled zero-energy ground state, it is complete for a classical complexity class. It would then be interesting to try and reformulate this problem in purely classical terms. Aharanov and Grilo [33] recently undertook a similar task for the MA-complete Stoquastic QSAT problem and showed that it is equivalent to a combinatorial optimization problem.

Our first result shows that there are seven complexity classes that have strong QCSPs as complete problems. Specifically, we introduce three new such problems to the already existing four. These seven classes belong to a larger set classes corresponding to polynomial-time computation and verification. Arguably, this set is sometimes considered the "most natural" or "relevant" set of complexity classes. We can differentiate the classes within this set by three defining properties: (1) type of circuit—classical or quantum; (2) type of witness state—no witness, classical witness, quantum witness; (3) type of error—no error, one-sided error, and two-sided error. Ranked in order of apparent difficulty, the seven classes mentioned here are (ignoring the gate set issue for the moment):

1. P: Classical, no witness, no error.
2. coRP: Classical, no witness, one-sided error.
3. NP: Classical, classical witness, no error.
4. MA: Classical, classical witness, two-sided error (and one-sided error).
5. BQP$_1$: Quantum, no witness, one-sided error.
6. QCMA: Quantum, classical witness, two-sided error (and one-sided error).
7. QMA$_1$: Quantum, quantum witness, one-sided error.

Are there any obvious omissions in this list for which we can expect another complete QCSP problem? By considering combinations of their three defining properties, there are a total of 18 classes. Without the seven (technically 9 as MA $=$ MA$_1$ and QCMA $=$ QCMA$_1$) included here, the remaining 9 classes without known QSAT complete problems are: (1-3) $\{c\}_{\text{circuit}} \times \{q\}_{\text{witness}} \times \{\varnothing, 1, 2\}_{\text{error}}$, (4-6) $\{q\}_{\text{circuit}} \times \{\varnothing, c, q\}_{\text{witness}} \times \{\varnothing\}_{\text{error}}$, (7-8) $\{c, q\}_{\text{circuit}} \times \{\varnothing\}_{\text{witness}} \times \{2\}_{\text{error}}$, and (9) $\{q\}_{\text{circuit}} \times \{q\}_{\text{witness}} \times \{2\}_{\text{error}}$. The first set corresponds to classical classes with a quantum witness, however, we cannot have such classes and can then be discarded. The second set corresponds to quantum classes with no error. Little is known about these classes as they appear to be extremely difficult to work with. This is mainly due to the perfect completeness requirement which implies that no matter which instance (and witness in the case of verification classes), the circuit is never fooled into accepting. This is also why in the classification above we let one-sided error classes default to those with perfect completeness and bounded soundness instead of also considering the possibility of perfect soundness and bounded completeness. While it may be possible that there are complete QCSPs for these classes, we conjecture that this is very unlikely. The remaining three classes are BPP, BQP, and QMA. Since coRP $\subseteq$ BPP, BQP$_1$ $\subseteq$ BQP, and QMA$_1$ $\subseteq$ QMA there are clearly strong QCSPs in this classes. Demonstrating that a QCSP is hard for them requires encoding a probabilistic circuit into an instance of this problem. Usually, these proofs are done via the circuit-to-Hamiltonian transformation that encodes the circuit into the ground state of a Hamiltonian. However, if the circuit is probabilistic, there is no state that satisfies all Π_{init}, Π_{prop}, and Π_{out} clauses simultaneously. Thus, other techniques are needed, and not many are known.[9] Another approach that may offer a positive answer to this question is if these classes admit a scheme that boosts their acceptance probabilities to 1. Jordan *et al.* [34] showed that this was possible for QCMA (demonstrating that $\mathsf{QCMA} = \mathsf{QCMA}_1$), but whether this is possible for the other classes remains a long-standing open question.

Our second result adds an additional 6 classes to the set of classes with strong QCSPs complete problems. As mentioned, these results are obtained by considering "sums" and "products" of pairs of QSAT problems from our previous result. The first question one might ask is: if there are 7 QSAT complete problems for the "more natural" classes, why are there only 6 and not 21 "sum" and "product" complete QSAT problems? While this is explained in more detail in Section 8, we note that this follows from a class property stating that if $\mathsf{A} \subseteq \mathsf{B}$, then $\mathsf{PI}(\mathsf{A}, \mathsf{B}) = \mathsf{SoPU}(\mathsf{A}, \mathsf{B}) = \mathsf{B}$. This is the reason why prior to our work, these classes were not considered. Indeed, the classes with complete QSAT problems were P, NP, MA, and QMA_1, but $\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{MA} \subseteq \mathsf{QMA}_1$, and so the classes PI and SoPU that resulted from considering pairs of these classes were trivial, i.e. they were equal to the largest of the two classes. With our results, there are now QSAT problems that are complete for classes not known to be contained within each other. These are: $\mathsf{coRP} \subseteq_? \mathsf{NP}$, $\mathsf{NP} \subseteq_? \mathsf{BQP}_1$, and $\mathsf{MA} \subseteq_? \mathsf{BQP}_1$. Hence, we only obtain 6 problems that are complete for new and seemingly nontrivial classes. Besides the inclusions shown in Fig. 1, there is very little we know for certain about this classes. However, another thing to note, and one that is of importance in Corollary 1.2, is that some of these classes may be related via derandomization conjectures. Among the complexity theory community, it is conjectured that $\mathsf{P} = \mathsf{BPP}$, implying that every problem solvable with a probabilistic classical computer can also be solved deterministically. If true, $\mathsf{PI}(\mathsf{coRP}, \mathsf{NP}) = \mathsf{PI}(\mathsf{P}, \mathsf{NP})$ and $\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP}) = \mathsf{SoPU}(\mathsf{P}, \mathsf{NP})$. Then, since $\mathsf{P} \subseteq \mathsf{NP}$, these two complexity classes are trivial and both are equal to NP. There is also reason to believe that a weaker version of derandomization where $\mathsf{NP} = \mathsf{MA}$ may be true. Similarly, in this case, these two classes become equal to NP. Now, observe that since it is not known whether $\mathsf{BQP}_1 \subseteq \mathsf{NP}$ or vice versa, and there is no strong evidence for either, the classes $\mathsf{PI}(\mathsf{BQP}_1, \mathsf{NP}), \mathsf{SoPU}(\mathsf{BQP}_1, \mathsf{NP}), \mathsf{PI}(\mathsf{BQP}_1, \mathsf{MA})$, and $\mathsf{SoPU}(\mathsf{BQP}_1, \mathsf{MA})$ would be expected to exhibit the most unique behavior.

Finally, let us consider the implications of Corollary 1.2. If such a theorem is proven and the classification contains fewer than 13 classes, this could have exciting implications as the majority of these classes appear to be quite distinct from one another. Even if such a theorem proves any of the derandomization conjectures, it would be a great result since such proofs have eluded us for many decades. On the other hand, a classification showing that there are more than 13 classes would be a stark contrast with classical strong CSPs, which can be completely classified as being either in P or NP-complete [35, 36]. This would highlight the more rich and complex panorama of strong QCSPs, and establish a larger repertoire of problems from which to construct reductions and potentially describe the complexity of other problems.

1.4 Organization

In Section 2 we present a brief discussion of the complexity classes relevant for this paper, as well as a brief summary of Bravyi's proof that k-QSAT is QMA_1-complete for $k \geq 4$ [3]. Elements of this proof will be relevant in the next sections. In Section 3 we present the first and simplest QSAT problem, LCT-QSAT, and prove that it is BQP_1-complete. Subsequently, in Section 4, we make several changes to the definition of this past problem and obtain SLCT-QSAT, which has better locality and particle dimensionality. We prove Theorem 2. In Sections 5 and 6 we demonstrate that by making small changes to the definition of SLCT-QSAT, we can obtain two other QSAT problems that are complete for QCMA and coRP, proving Theorems 3 and 4. Finally, in Section 8 we introduce the direct sum and product for QCSPs and demonstrate that these operations yield six problems that are complete for the classes PI and SoPU.

[9] Another technique is to reduce an already known hard problem into an instance of the target problem. For the LH problem, this is done via perturbation theory gadgets [2, 10, 17]. These gadgets however rely on approximations and hence do not preserve perfect completeness.

2 Preliminaries

In this section, we briefly review a few useful concepts for this work. In Section 2.2, we discuss some relevant classical and quantum complexity classes along with their one-sided error variation. Subsequently, in Section 2.3 we review the quantum satisfiability problem k-QSAT defined by Bravyi [3] and his use of the circuit-to-Hamiltonian transformation to prove the problem is QMA_1-hard. For readers already familiar with these topics, we simply refer them to the discussion about classes with perfect completeness in Section 2.2, and to Eq. (5) in Section 2.3 which points to the clock encoding used in the constructions of this paper.

2.1 Notation

For a bitstring x, let $|x|$ denote the number of bits in x. For $n \in \mathbb{N}^+$, let $[n] = \{0, 1, \ldots, n-1\}$.

For some complexity classes, we specify the gate set used. Here, we use the Clifford-cyclotomic gate sets $\mathcal{G}_m$ defined in Ref. [5]. Specifically, we only consider those that are a power of two. These are: $\mathcal{G}_2 := \{X, \text{CNOT}, \text{Toffoli}, H \otimes H\}$, $\mathcal{G}_4 := \{X, \text{CNOT}, \text{Toffoli}, \zeta_8 H\}$, and for $l \geq 3$, $\mathcal{G}_{2^l} := \{H, \text{CNOT}, T_{2^l}\}$. Here, $T_{2^l} = \text{diag}(1, \zeta_{2^l})$ where $\zeta_{2^l} = e^{2\pi i/2^l}$ is a primitive 2^l-th root of unity.

In all quantum circuits considered here, we let $U_0 = I$ (sometimes also written as $U_{0,0}$). The same is true for classical circuits Q and classical reversible circuits R. For circuits that decide computational problems, we let *ans* denote the qubit that when measured provides this decision. We accept the instance if the qubit is measured and yields outcome "1", and reject otherwise. Usually, *ans* is the first ancilla qubit of the circuit.

For a circuit U_n that decides an instance x with $|x| = n$, we denote U_x as the circuit where the instance x is encoded into it and the inputs are only ancilla qubits in the $|0\rangle$ state.

2.2 Complexity classes

A *promise problem* $A = (A_{yes}, A_{no})$ is a computational problem consisting of two non-intersecting sets $A_{yes}, A_{no} \subseteq \{0,1\}^*$ where given an instance $x \in \{0,1\}^*$ (promised to be in one of the two sets), one is tasked to determine if $x \in A_{yes}$ (x is a yes-instance) or $x \in A_{no}$ (x is a no-instance).[10] Let $n = |x|$ denote the size of x.

The first complexity class we consider is that composed of promise problems that can be decided probabilistically on a classical computer.

Definition 2.1 (BPP). *A promise problem A is in* BPP *iff there exists a polynomial p and a family of polynomial-time uniform classical algorithms $\{Q_n\}$ that take as input a binary string x and a random bitstring $r \in \{0,1\}^{p(n)}$, such that:*

- *(Completeness) If $x \in A_{yes}$,* $\Pr[Q_n \text{ accepts } (x,r)] \geq 2/3$.
- *(Soundness) If $x \in A_{no}$,* $\Pr[Q_n \text{ accepts } (x,r)] \leq 1/3$.

The quantum analogue of this class, where the classical algorithm becomes a quantum algorithm, is BQP.

Definition 2.2 (BQP). *A promise problem A is in* BQP *iff there exists a polynomial q and a family of polynomial-time uniform quantum circuits $\{U_n\}$ that take as input a binary string x and use at most $q(n)$ ancilla qubits, such that:*

- *(Completeness) If $x \in A_{yes}$,* $\Pr[U_n \text{ accepts } x] \geq 2/3$.
- *(Soundness) If $x \in A_{no}$,* $\Pr[U_n \text{ accepts } x] \leq 1/3$.

One can also consider a class where instead of computing the solution, one is tasked with verifying a given candidate solution (also known as *witness*).

Definition 2.3 (QMA). *A promise problem A is in* QMA *iff there exist polynomials p, q and a family of polynomial-time uniform quantum (verifier) circuits $\{U_n\}$ that take as input a binary string x and a witness state $|\psi_x\rangle$ of at most $p(n)$ qubits, and use at most $q(n)$ ancilla qubits, such that:*

[10] The asterisk over the set is known as the *Kleene star* and is used to represent strings of any finite size.

- *(Completeness) If $x \in A_{yes}$, then there exists a state $|\psi_x\rangle$ such that* $\Pr[U_n \text{ accepts } (x, \psi_x)] \geq 2/3$.
- *(Soundness) If $x \in A_{no}$, then for any witness state $|\psi_x\rangle$,* $\Pr[U_n \text{ accepts } (x, \psi_x)] \leq 1/3$.

If we receive a classical binary bitstring y_x as a witness instead of a quantum state, we obtain QCMA.

Definition 2.4 (QCMA)**.** QCMA *is defined in a similar way as* QMA*, except the quantum state $|\psi_x\rangle$ is replaced by the computational basis state $|y_x\rangle$.*

2.2.1 Perfect completeness

In this paper we are interested in a variation of the classes above where the acceptance probability of yes-instances is equal to one. These classes are said to have *perfect completeness*, and they are one of the two types of classes with *one-sided error*. Although these classes appear to be similar to their two-sided error variation, quantum complexity classes with one-sided error require a more precise treatment as they are not known to be independent of the gate set used. Indeed, the Solovay-Kitaev theorem [37] used to resolve this issue for quantum classes with two-sided error only works for approximate equivalence of universal gate sets and not perfect equivalence. Thus, for these classes (with some exceptions), one must specify the gate set used by the quantum circuits. This is not the case for classical complexity classes as it is known that every classical circuit using gate set $\mathcal{G}$ can be perfectly simulated by another circuit using a universal gate set $\mathcal{G}'$.

Given this discussion, we can then define one-sided error classes as follows:

Definition 2.5 (Classes with perfect completeness)**.** *Let $\mathcal{C}$ be a complexity class with two-sided error. The variation of this class with perfect completeness is defined in a similar way to $\mathcal{C}$ except for the following differences:*

1. *The acceptance probability must be exactly* 1 *when $x \in \mathcal{A}_{yes}$.*
2. *If $\mathcal{C}$ is a quantum complexity class, the gate set $\mathcal{G}$ used by the quantum circuits $\{U_n\}$ must be specified.*

The class with one sided error is generally denoted as $\mathcal{C}_1$, or $\mathcal{C}_1^{\mathcal{G}}$ if it is a quantum complexity class.

This sensibility to the gate set in quantum complexity classes is the reason why, in Theorems 1 and 2, we explicitly stated that LCT-QSAT and SLCT-QSAT are complete for BQP_1 with the particular choice of gate set $\mathcal{G}_8$. It also presents other complications. To see this, consider BQP. It is evident that $\mathsf{BQP}_1^{\mathcal{G}} \subseteq \mathsf{BQP}$ for any arbitrary gate set $\mathcal{G}$, and also that $\mathsf{P} \subseteq \mathsf{BQP}$. However, is it true that $\mathsf{P} \subseteq \mathsf{BQP}_1^{\mathcal{G}}$? Fortunately, for the Clifford+T gate set used in this paper, the class $\mathsf{BQP}_1^{\mathcal{G}_8}$ follows the intuitive containment of classes. Indeed, P and coRP are contained in this class. To see this, consider the fact that any classical circuit is equivalent to a reversible classical circuit, and moreover the circuit may consist only of Toffoli gates. Then, a result by Giles and Selinger [38] shows each Toffoli can be efficiently and perfectly decomposed in terms of Clifford and T gates. For any problem in P or coRP, it can also be decided by a $\mathsf{BQP}_1^{\mathsf{Cliff+T}}$ algorithm that runs the classical reversible circuit, replacing each Toffoli by its decomposition in terms of Clifford and T gates. The circuit is guaranteed to have perfect completeness. The relationships between the eight classes mentioned here is given by $\mathsf{coRP} \subseteq \mathsf{BQP}_1^{\mathcal{G}} \subseteq \mathsf{BQP} \subseteq \mathsf{QCMA}_1^{\mathcal{G}} = \mathsf{QCMA} \subseteq \mathsf{QMA}_1 \subseteq \mathsf{QMA}$.

A relevant feature of these six classes is that the gap between the acceptance and rejection probabilities can be amplified. It can be shown that as long as these probabilities are separated by the inverse of a polynomial, there is a scheme (e.g. repetition and majority vote) that makes these probabilities exponentially close to 1 and 0, respectively, requiring only a polynomial number of extra steps [1, 39]. We will make use of this statement to show that the quantum satisfiability problems mentioned in this paper meet the required soundness conditions.

Interestingly, Jordan *et al.* [34] showed that if the circuits that decide a QCMA problem consist of gates with a succinct representation (e.g. the Clifford+T gate set), the acceptance probability of yes-instances can be amplified additively to be exactly 1. In other words, they showed that $\mathsf{QCMA}_1^{\mathcal{G}} \subseteq \mathsf{QCMA}$, concluding that $\mathsf{QCMA}_1^{\mathcal{G}} = \mathsf{QCMA}$. This is the reason why in Theorem 3 we state that the problem WITNESSED SLCT-QSAT is QCMA-complete. We show that this problem is $\mathsf{QCMA}_1^{\mathcal{G}}$-complete, and by this result, it is also QCMA-complete. To this day, it remains an open question whether perfect amplification can also work for BQP and QMA. In the case of QMA, it is believed that this is not the case as one can show that there exists an oracle for which $\mathsf{QMA} \neq \mathsf{QCMA}_1$ [40]. However, a similar claim was made about QCMA and QCMA_1.

2.3 k-QSAT & the Circuit-to-Hamiltonian transformation

Here, we introduce QUANTUM k-SAT (denoted here as k-QSAT) as defined by Gosset and Nagaj in Ref. [4]. We present relevant parts of the proofs showing that k-QSAT is contained in QMA_1 for any constant k, and QMA_1-hard for $k \geq 6$. While Bravyi's [3] original work demonstrates hardness for $k \geq 4$, we choose to present this slightly weaker result for brevity, but also to introduce our clock encoding and notation useful for the rest of this paper.

As we are working to prove the inclusion and hardness of this problem for a class requiring perfect completeness, it is necessary to specify the gate set used by the quantum circuits. For now, let us simply refer to this gate set as $\mathcal{G}$, and we will later show specifically which gates should be used. In addition, we also have to be wary that all operations can be performed with perfect accuracy using gates from this set and all measurements are in the computational basis. For this purpose, Gosset and Nagaj introduce the following set of projectors.

Definition 2.6 (Perfectly measurable projectors)**.** *Let $\mathcal{P}$ be the set of projectors such that every matrix element in the computational basis has the form*

$$\frac{1}{4}(a + ib + \sqrt{2}c + i\sqrt{2}d) \tag{1}$$

for $a, b, c, d \in \mathbb{Z}$.

The (promise) problem k-QSAT can be defined as follows.

Definition 2.7 (k-QSAT)**.** *Given an integer n and an instance x consisting of a collection of projectors $\{\Pi_i\} \subset \mathcal{P}$ where each Π_i acts nontrivially on at most k qubits, the problem consists on deciding whether (1) there exists an n-qubit state $|\psi_{sat}\rangle$ such that $\Pi_i|\psi_{sat}\rangle = 0$ for all i, or (2) for every n-qubit state $|\psi\rangle$, $\Sigma_i \langle\psi|\Pi_i|\psi\rangle \geq 1/poly(n)$. We are promised that these are the only two cases. We output "YES" if (1) is true, or "NO" otherwise.*

One can think of this problem as being presented with a list of constraints or *clauses* (the projectors Π_i) and tasked with distinguishing between the following cases: (1) there exists a state a state that satisfies all constraints (a *satisfying state*), or (2) any possible state induces a violation of the constraints greater than $1/poly(n)$. The promise sets the conditions for classifying instances as either $x \in \mathcal{L}_{\text{yes}}$ or $x \in \mathcal{L}_{\text{no}}$. Without this promise, the problem becomes seemingly harder as it requires distinguishing between the case where the projectors are satisfiable, and the case where they are not but the violation induced by some states could be exponentially close to zero. Without a promise, the problem is most likely not contained in QMA_1.

2.3.1 In QMA_1

Suppose we are presented with a witness state $|\psi_{\text{wit}}\rangle$ and a k-QSAT instance composed of projectors $\{\Pi_i\}$. The quantum algorithm that decides whether this state satisfies all projectors Π_i consists of simply measuring the eigenvalues of all projectors on this state. Then, if all measured eigenvalues are 0, we conclude that all projectors are satisfied by the state and output "YES". Otherwise, we reject.

Specifically, we measure the eigenvalue of a projector Π_i by applying the unitary

$$V(\Pi_i) = \Pi_i \otimes X + (I - \Pi_i) \otimes I, \tag{2}$$

to the witness and an additional ancilla qubit in the state $|0\rangle$, followed by a measurement of the ancilla in the computational basis. Here, X denotes the Pauli-X gate. The probability that $|\psi_{\text{wit}}\rangle$ does not satisfy projector Π_i (obtain outcome "1") is given by

$$p_i = \langle\psi_{\text{wit}}|\, \Pi_i \,|\psi_{\text{wit}}\rangle\,. \tag{3}$$

By defining the acceptance probability as the probability that all measurements produce outcome "0", and assuming $V(\Pi_i)$ can be implemented perfectly with gate set $\mathcal{G}$, one can show that this algorithm meets the completeness and soundness conditions of QMA_1, and so k-QSAT is contained in this class. The proof of these statements is not shown here since a similar argument will be presented at the end of Section 3.

As mentioned, to support this claim, it is necessary to demonstrate that $V(\Pi_i)$ can be implemented perfectly with gate set $\mathcal{G}$. In Ref. [3], Bravyi argued that this was in fact possible for all k-local projectors if these were picked from a field $\mathbb{F} \subset \mathbb{C}$ such that all of their matrix elements had an exact representation. However, this result was later withdrawn [4]. Gosset and Nagaj [4] later showed that Bravyi's proof could be fixed by restricting the projectors to be from the set $\mathcal{P}$ above. Their argument is based on Giles and Selinger's theorem, showing that any $m \times m$ unitary whose matrix elements are all of the form $1/2^s(a+ib+\sqrt{2}c+i\sqrt{2}d)$ where $s \in \mathbb{N}$ and $a, b, c, d \in \mathbb{Z}$, can be decomposed into a sequence of $\mathcal{O}(3^m s)$ gates from the Clifford+T gate set using a single additional ancilla qubit. Then, since $\Pi_i \subset \mathcal{P}$ and each Π_i is a k-local projector with constant k, $V(\Pi_i)$ has the required form. Moreover, since $V(\Pi_i)$ is independent of n, i.e. a constant-sized matrix, it can be decomposed into polynomially-many gates of the Clifford+T set. This discussion then concludes that k-QSAT (as defined in Definition 2.7) is in QMA_1 if $\mathcal{G} = \mathcal{G}_8 = \{H, \mathrm{CNOT}, T\}$.

Recently, Rudolph [31] generalized this result and brought it closer to Bravyi's original definition by showing that k-QSAT is complete for a field. Specifically, the 3-QSAT problem where the projectors are drawn from the field $\mathbb{Q}(\zeta_{2^l})$ is complete for $\mathsf{QMA}_1^{\mathcal{G}_{2^l}}$ as long as $l \geq 3$.

2.3.2 QMA_1-hard

Now, we discuss elements of the proof demonstrating that k-QSAT is QMA_1-hard when $k \geq 6$ and for any gate set $\mathcal{G}$ that is universal for quantum computation.

To show this result, we have to prove that any instance x of an arbitrary promise problem in QMA_1 can be transformed or *reduced* in polynomial time into an instance x' of k-QSAT where the answer to the original problem and the transformed one is the same for all instances. Furthermore, we also need to show that all projectors of the resulting k-QSAT instance act on at most 6 qubits.

Let $U_x = U_L \dots U_1$ with $U_i \in \mathcal{G}$ and $L = poly(n)$ be the QMA_1 verification circuit where given an instance x of a problem $A = (A_{yes}, A_{no})$, U_x decides whether $x \in A_{yes}$ or $x \in A_{no}$. The input to the circuit consists of the p-qubit witness state $|\psi_{\mathrm{wit}}\rangle$, and a q-qubit ancilla register D (referred to as the *data* register) initialized to the state $|0\rangle^{\otimes q}$, where p and q are two polynomials in $n = |x|$. Additionally, let the answer be obtained by measuring one of the ancilla qubits (denoted by *ans*) in the computational basis, where outcome "1" means the instance is accepted, while outcome "0" means the instance is rejected. The goal of the reduction is to engineer a set of 6-local projectors such that they are uniquely satisfied by the state encoding the evaluation of the circuit U on $|\phi_0\rangle := |0\rangle^{\otimes q} \otimes |\psi_{\mathrm{wit}}\rangle$ at all steps of the computation. This state is appropriately known as the (computational) *history state* and is given by

$$|\psi_{hist}\rangle := \frac{1}{\sqrt{L+1}} \sum_{t=0}^{L} U_t \dots U_0 |\phi_0\rangle_D \otimes |C_t\rangle_C \,, \tag{4}$$

where $U_0 := I$ is a dummy unitary introduced for convenience. Here, we have introduced a *clock* register C acting on a new (not yet specified) Hilbert space used to keep track of the current step in the computation. Clearly, this history state can be defined in many ways depending on the implementation of the states $|C_t\rangle$. In this paper, we choose a clock encoding acting on $\mathcal{H}_{clock} = (\mathbb{C}^3)^{\otimes L+1}$ consisting of the *ready* state $|r\rangle$, the *active* state $|a\rangle$, and the *dead* state $|d\rangle$, which progresses as

$$\begin{aligned} |C_0\rangle &= |a_0 r_1 r_2 \dots r_L\rangle \,, \\ |C_1\rangle &= |d_0 a_1 r_2 \dots r_L\rangle \,, \\ |C_2\rangle &= |d_0 d_1 a_2 \dots r_L\rangle \,, \\ &\vdots \\ |C_L\rangle &= |d_0 d_1 d_2 \dots a_L\rangle \,. \end{aligned} \tag{5}$$

We refer to these basis states as the *legal* states of the clock, and all other basis states as *illegal*.

The projectors that allow us to build the required k-QSAT instance act on both of these Hilbert spaces and are given by

$$\begin{aligned} P^{(i)}_{init} &:= |1\rangle\langle 1|_i \otimes |a\rangle\langle a|_0 \, , \\ P^{(i)}_{out} &:= |0\rangle\langle 0|_i \otimes |a\rangle\langle a|_L \, , \\ P^{(i)}_{prop,U} &:= \frac{1}{2}\big[I^{\otimes 2} \otimes |ar\rangle\langle ar|_{i-1,i} + I^{\otimes 2} \otimes |da\rangle\langle da|_{i-1,i} - U \otimes |da\rangle\langle ar|_{i-1,i} - U^{\dagger} \otimes |ar\rangle\langle da|_{i-1,i} \big] , \end{aligned} \tag{6}$$

which receive an index to specify its action on another particle. Observe that P_{init} and P_{out} act on a single data and clock particle, while $P_{prop,U}$ acts on two data qubits and two clock particles. As each clock particle can be represented by two qubits, albeit a bit wastefully, it is evident that these projectors are at most 6-local (on qubits). Other clock encodings may lead to different locality.[11]

Each projector in Eq. (6) penalizes states that do not meet certain requirements. (Initialization) P_{init} requires that when clock particle 0 is in the state $|a\rangle$, data qubit i is initialized to $|0\rangle$. (Computational propagation) $P_{prop,U}$ requires that as clock particles i and $i+1$ transition from $|ar\rangle$ to $|da\rangle$, U is applied to two qubits of the data register. (Readout) Finally, P_{out} requires that when clock qudit L is in the state $|a\rangle$, data qubit i is in the state $|1\rangle$.[12] Aside from these projectors, one also has to define

$$\begin{aligned} P_{start} &:= |r\rangle\langle r|_0 \, , \\ P_{stop} &:= |d\rangle\langle d|_L \, , \\ P^{(i)}_{clock} &:= |r\rangle\langle r|_i \otimes (I - |r\rangle\langle r|)_{i+1} + |a\rangle\langle a|_i \otimes (I - |r\rangle\langle r|)_{i+1} + |d\rangle\langle d|_i \otimes |r\rangle\langle r|_{i+1} \, , \end{aligned} \tag{7}$$

which are at most 4-local projectors requiring that the clock states have the form described in Eq. (5). Furthermore, these six types of projectors are of the form given in Definition 2.6 and are hence projectors from $\mathcal{P}$ as required. Finally, using the six types of projectors of Eq. (6) and Eq. (7), the instance that encodes the verifier circuit $U = U_L \dots U_1$ is given by

$$\begin{aligned} H_{init} &:= \sum_{b \in ancilla} P^{(b)}_{init}, \\ H_{prop} &:= \sum_{t=1}^{L} P^{(t)}_{prop,U_t} \\ H_{out} &:= P^{(ans)}_{out}, \\ H_{clock} &:= P_{start} + P_{stop} + \sum_{c \in C} P^{(c)}_{clock}. \end{aligned} \tag{8}$$

We illustrate this instance in Fig. 2 (for simplicity, we assume the data register consists only of four qubits). The set of projectors that define this k-QSAT instance are the individual terms of the sum; however, we often group them into positive semi-definite terms resembling those of the Local Hamiltonian problem [1]. In Eq. (8), the H_{init} term requires that all ancilla qubits from register $\mathcal{D}$ are initialized to $|0\rangle$, leaving the data qubits for the witness state "free" or un-initialized. H_{prop} defines a clock register of $L+1$ particles and requires that as time progresses from $t-1$ to t, U_t is applied to the data qubits. H_{out} requires that at the end of the computation ans is measured to be "1". Finally, H_{clock} requires that we obtain a running clock register and that the clock progresses as shown in Eq. (5). Together, the terms H_{init}, H_{prop}, and H_{clock} require that if there exists a state satisfying all of their projectors, the state must mimic the evaluation of the quantum circuit $U = U_L \dots U_1$ on the state $|\phi_0\rangle$. This is the history state of Eq. (4) with the clock encoding of Eq. (5). Moreover, if the verification circuit U accepts with certainty, the history state also satisfies H_{out} and is thus the unique ground state of the 6-local Hamiltonian $H = H_{init} + H_{prop} + H_{out} + H_{clock}$.

This concludes the transformation of the circuit into local Hamiltonians. Completing the proof that 6-QSAT is QMA_1-hard requires showing that if $x \in A_{yes}$ then x' has a frustration-free ground state, while if $x \in A_{no}$ then the ground state energy of H is not too low. The former case is straightforward since

[11] In Ref. [3], Bravyi employs a four-state clock encoding, $2L+1$ clock basis states, and an additional propagation projector. This allows interactions between either two clock particles at a time or one clock particle and two data qubits, resulting in 4-local projectors. However, this comes at a cost of increased clock particle dimensionality.

[12] Unlike Bravyi [3] and Meiburg [30], we define P_{out} so it is satisfied when the logical qubit is in the state $|1\rangle$, and not $|0\rangle$.

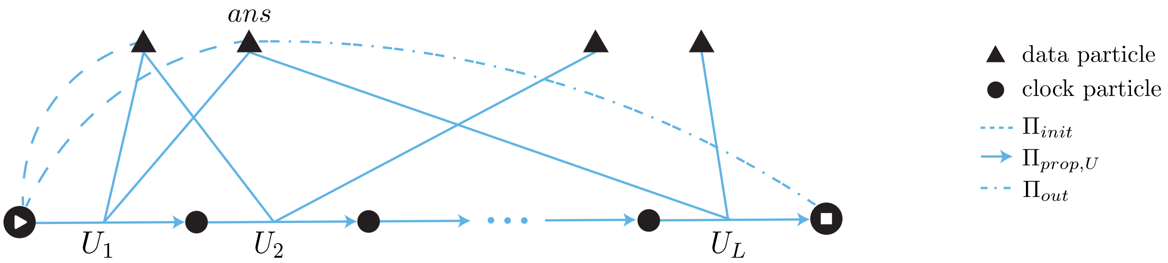


Figure 2: Representation of a k-QSAT instance which encodes a QMA_1 verification circuit $U = U_L \dots U_1$. For simplicity, we let U act on four data qubits: two ancilla qubits and two qubits for the witness state. The ancillas are those present in P_{init} clauses, and the witness state qubits those that are un-initialized. The ancilla measured at the end of the computation is labeled ans. The leftmost clock particle has a "start" icon to represent the action of the P_{start} clause, and the rightmost a "stop" icon for the P_{stop} clause. The P_{clock} clauses are represented by the arrows on top of the $P_{prop,U}$ lines that connect clock particles together. It is represented by an arrow since this clause establishes the clock progression.

the history state of Eq. (4) is, by construction, the state that satisfies all clauses of the instance. On the other hand, the latter case is significantly more challenging as it requires relating the null spaces of the four non-commuting operators $H_{init}, H_{prop}, H_{out}$, and H_{clock}. The key to accomplish this is through Kitaev's Geometric lemma:

Lemma 2.1 (Geometric lemma [1])**.** *Let $H_1 \succeq 0$ and $H_2 \succeq 0$ be two positive semi-definite operators with null spaces $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Suppose the null spaces have trivial intersection, i.e. $\mathcal{S}_1 \cap \mathcal{S}_2 = \{\mathbf{0}\}$. Let $\gamma(H)$ denote the smallest non-zero eigenvalue of H. Then,*

$$H_1 + H_2 \succeq \min\{\gamma(H_1), \gamma(H_2)\} \cdot \frac{(1-\alpha)}{2} \tag{9}$$

where

$$\alpha = \max_{|\eta\rangle \in \mathcal{S}_1} \langle \eta | \Pi_{\mathcal{S}_2} | \eta \rangle \,. \tag{10}$$

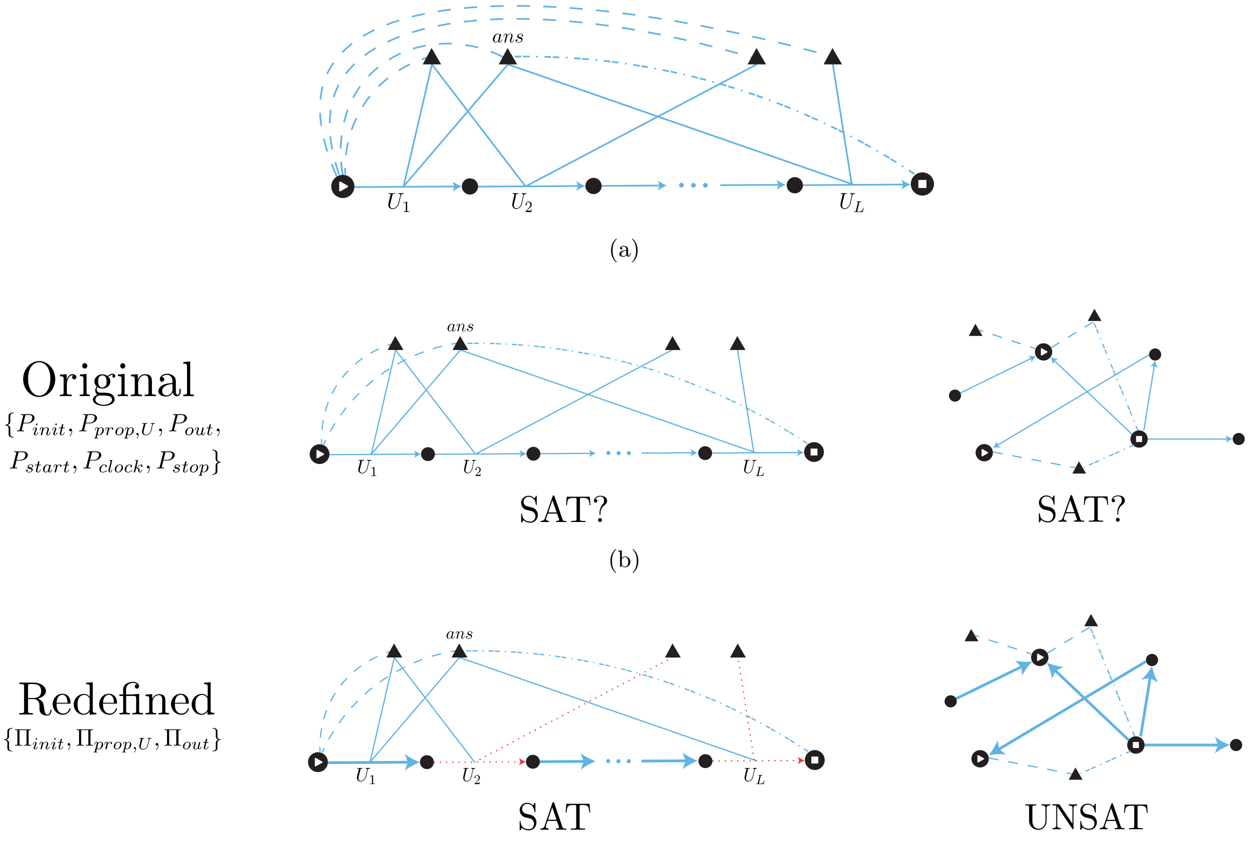
We do not present the rest of the hardness proof here as it is beyond the scope of this preliminary section, and a similar analysis is presented in Section 3.5.

3 A BQP_1-complete Problem Leveraging Monogamy

In this section we prove Theorem 1, which states that the quantum satisfiability problem Linear-Clock-Ternary-QSAT is BQP_1-complete with the Clifford+T gate set. In particular, in Section 3.1, we provide an overview on how we build upon the circuit-to-Hamiltonian transformation introduced in Section 2.3.2 to engineer a satisfiability problem that is both contained within BQP_1 and BQP_1-hard. In Section 3.2, we present the detailed definition of the problem and formally begin the proof of its containment in BQP_1. In particular, we describe the structure of input instances and introduce terminology useful for later subsections. Section 3.3 continues with an in-depth analysis on the satisfiability of instances, categorizing each one as trivially unsatisfiable, trivially satisfiable, or one requiring the assistance of a quantum algorithm. Then, based on the previous subsection, Section 3.4 presents the hybrid quantum-classical algorithm that decides the satisfiability of all instances along with an analysis of its correctness. This concludes that the problem is contained in BQP_1. Lastly, Section 3.5 shows that the problem is BQP_1-hard.

3.1 Construction overview

The goal of the construction is to design a QSAT problem that can encode the computation of any quantum circuit in BQP_1, while also being able to solve any of its instances in quantum polynomial-time with perfect


ans
U_1
U_2
U_L
Original
$\{P_{init}, P_{prop,U}, P_{out},$
$P_{start}, P_{clock}, P_{stop}\}$
SAT?
SAT?
Redefined
$\{\Pi_{init}, \Pi_{prop,U}, \Pi_{out}\}$
SAT
UNSAT

(a)

(b)

(c)

Figure 3: (a) Example of a typical instance that encodes the computation of a BQP_1 circuit $U_L \dots U_1$. The satisfiability of the instance can also be decided in BQP_1. (b) Examples of the troublesome instances whose satisfiability is not known to be decidable with a BQP_1 algorithm. (c) The above instances recast with the new set of projectors of Eqs. (12)–(14). The bold blue arrows represent the $\Pi_{prop,U}$ clauses which now also indicate the particles should be maximally entangled, and the dotted red arrows those that are connected to undefined logical qudits. With these projectors, their satisfiability can be more easily decided. The instance on the left is satisfiable due to the undefined clauses, while the one on the right is unsatisfiable as any potential satisfying state would violate monogamy of entanglement. The instance in (a) has the same meaning/satisfiability with either set of projectors.

completeness and bounded soundness. We define the problem using projectors Π_{init}, $\Pi_{prop,U}$, and Π_{out} similar to P_{init}, $P_{prop,U}$, and P_{out} defined in Eq. (6).[13] To see why our projectors must differ from the original ones, consider the QSAT problem built with $\{P_{init}, P_{prop,U}, P_{out}, P_{start}, P_{clock}, P_{end}\}$. Showing that the problem is BQP_1-hard is straightforward, as we can encode the circuit that computes the answer to a BQP_1 problem in a similar way as that shown in Section 2.3. This time however, all data particles in the instance should be initialized, preventing having free particles that can accommodate a witness state (see Fig. 3a). The difficulty lies in demonstrating that every instance generated with a polynomial number of these projectors can also be decided in BQP_1. There is a fundamental and a practical limitation for this:

- Instances which encode the computation of a QMA_1 problem, e.g. the instance in Fig. 2 and the left instance in Fig. 3b, are valid inputs. This is problematic since it is unknown how to decide these instances in BQP_1 (and doing so would show that $\mathsf{BQP}_1 = \mathsf{QMA}_1$).

[13] The projectors P_{start}, P_{clock}, and P_{stop} associated with the clock encoding remain unchanged and are integrated into the definitions of Π_{init}, Π_{prop}, and Π_{out}.

- Input instances may form intricate structures complicating the task of deciding if a satisfying state exists, e.g. the right instance in Fig. 3b.

We define the projectors Π_{init}, Π_{prop}, and Π_{out} to address these two difficulties (see Fig. 3c). Importantly, these projectors do not significantly alter the proof that the problem is BQP_1-hard and can proceed as mentioned. Now, let us briefly discuss how we overcome both difficulties.

Instances like those in Fig. 2, which have a proper structure and uninitialized data particles, are prototypical examples of QMA instances. These "free" particles give one the freedom to guess if there exists a state they can be in such that the instance can be satisfied (or equivalently be provided with such a state which we verify). To address this issue, we remove the need to guess a satisfying state by introducing a new *undefined* basis state $|?\rangle$ (making the data particles 3-dimensional), such that setting the free data particles to this state always results in a satisfiable instance. More specifically, we achieve this by defining $\Pi_{prop,U}$ so that if any data particle in the clause is in state $|?\rangle$, the clause is satisfied without needing to apply the associated unitary.[14] Then, for these instances, the satisfying state is given by a truncated version of the history state (without a witness) since the computation is no longer required to elapse past the first $\Pi_{prop,U}$ clause acting on an undefined state. We say the instance is now "trivially satisfiable" as its structure alone suffices to determine its satisfiability. The formal statement of the claims presented here is given in Lemma 3.7.

To determine the satisfiability of intricate instances, the projectors are now also defined to leverage the principle of monogamy of entanglement. Each clock particle is equipped with two 2-dimensional auxiliary subspaces *CA* and *CB* (making them 12-dimensional) and the $\Pi_{prop,U}$ clauses are then defined to require that the *CB* subspace of the predecessor clock particle forms a $|\Phi^+\rangle$ Bell pair with the *CA* subspace of its successor. Then, if a *CA* or *CB* subspace is required to form more than one Bell pair, the principle of monogamy of entanglement states that only one of these clauses can be satisfied, and so the instance is unsatisfiable. Therefore, instances that are not deemed unsatisfiable because of this reason must form one-dimensional chains with a unique "time" direction. Finally, to guarantee that Π_{init} and Π_{out} only act on the ends of the chain, these make use of a new *endpoint* particle consisting of a single two-dimensional space *EC* and require that it also forms a Bell pair with either the *CA* (for Π_{init}) or *CB* (for Π_{out}) subspace of a clock particle. Fig. 4 provides a visual summary on how these projectors use monogamy to form one-dimensional instances.

Although these modifications do not get rid off all difficulties, they are enough to determine the satisfiability of all input instances via a hybrid algorithm. Briefly, the classical part of the algorithm evaluates the structure of the clauses in the instance and concludes whether it is trivially unsatisfiable, trivially satisfiable, or is one requiring the assistance of a quantum subroutine. Trivially unsatisfiable instances are those whose clause arrangement imply one or several clauses cannot be simultaneously satisfied, like those that violate monogamy of entanglement. On the other hand, trivially satisfiable instances are those whose clauses do not create any conflicts but whose structure is simple enough that the satisfying state can be inferred, like those with proper structure and uninitialized data particles. We show that the only type of instances that are not in either one of these cases, are those like Fig. 3a which express the computation of a quantum circuit on initialized ancilla qubits. For these instances, the classical algorithm makes use of a quantum subroutine that executes the quantum circuit expressed by the instance, while simultaneously measuring the eigenvalues of relevant projectors. The measurement outcomes indicate whether the instance should be accepted or rejected.

3.2 Problem definition

The problem we claim is BQP_1-complete is the following:

Definition 3.1 (Linear-Clock-Ternary-QSAT)**.** *The problem* Linear-Clock-Ternary-QSAT *is a quantum constraint satisfaction problem defined on the* 17*-dimensional Hilbert space*

$$\begin{aligned}\mathcal{H} = \; & \operatorname{span}\{|0_L\rangle, |1_L\rangle, |?_L\rangle\} \oplus \operatorname{span}\{|0_{EC}\rangle, |1_{EC}\rangle\} \\ & \oplus (\operatorname{span}\{|r_C\rangle, |a_C\rangle, |d_C\rangle\} \otimes \operatorname{span}\{|0_{CA}\rangle, |1_{CA}\rangle\} \otimes \operatorname{span}\{|0_{CB}\rangle, |1_{CB}\rangle\}),\end{aligned} \tag{11}$$

[14] This shows that although the data particles are 3-dimensional and the unitaries are gates from a set designed to act on qubits, there is no issue because the gates will never act on undefined data particles.

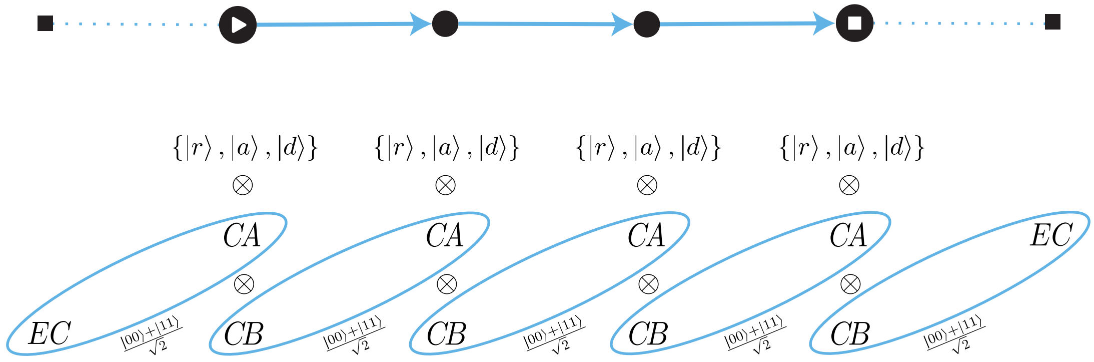


Figure 4: (a) A simplified graphical representation of a one-dimensional instance built with projectors Π_{init}, $\Pi_{prop,U}$, and Π_{out} analogous to previous figures. The data particles are not shown here. The endpoint particles are represented by the black squares, and its connections to clock particles via Π_{init} and Π_{out} clauses are given by the dotted blue lines. (b) A more descriptive and accurate representation of the figure above where the Hilbert space of both clock and endpoint particles is explicit. The blue ellipses signify that the encircled subspaces are required to be maximally entangled. Each clock particle has a free subspace spanned by $|r\rangle$, $|a\rangle$, and $|d\rangle$.

consisting of a logical, endpoint, and clock subspaces. The problem consists of 5 *types of projectors:* Π_{init}, $\Pi_{prop,U}$ *(one for every* $U \in \{H, HT, (H \otimes H)\text{CNOT}\}$*), and* Π_{out}*, each acting on at most four* 17*-dimensional qudits. The representation of these terms as projectors is too long to fit here, so instead we define the positive semi-definite operators*

$$\begin{aligned} O_{init} := \; & \Pi_{L,1} + \Pi_{C,2} + \Pi_{E,3} + \Pi_{start,2} + (I - |0\rangle\langle 0|)_1 \otimes |a\rangle\langle a|_2 \\ & + (I - |\Phi^+\rangle\langle\Phi^+|_{CA,EC})_{2,3}, \end{aligned} \tag{12}$$

$$\begin{aligned} O_{prop,U} := \; & \Pi_{L,1} + \Pi_{L,2} + \Pi_{C,3} + \Pi_{C,4} \\ & + (\Pi_{work,U} + I^{\otimes 2} \otimes \Pi_{clock,D})(\Pi_D \otimes \Pi_D \otimes I^{\otimes 2}) \\ & + (I^{\otimes 2} \otimes \Pi_{clock,?})(I - \Pi_D \otimes \Pi_D \otimes I^{\otimes 2}) \\ & + (I - |\Phi^+\rangle\langle\Phi^+|_{CA,CB})_{3,4}, \end{aligned} \tag{13}$$

and

$$\begin{aligned} O_{out} := \; & \Pi_{L,1} + \Pi_{C,2} + \Pi_{E,3} + \Pi_{stop,2} + |0\rangle\langle 0|_1 \otimes |a\rangle\langle a|_2 \\ & + (I - |\Phi^+\rangle\langle\Phi^+|_{CB,EC})_{2,3}, \end{aligned} \tag{14}$$

where it should be understood that $\Pi_{init}, \Pi_{prop,U}$*, and* Π_{out} *are actually the projectors with the same null space as the one defined by its corresponding positive semi-definite operators* O_{init}*,* $O_{prop,U}$*, and* O_{out}*. Here, we also use a slight abuse of notation to denote operators like* $0^5 \oplus (|r\rangle\langle r| \otimes I^4)$ *simply as* $|r\rangle\langle r|$*. Additionally,* O_{init}*,* $O_{prop,U}$*, and* O_{out} *are composed of the* 4*-local projector*

$$\Pi_{work,U} := \frac{1}{2}\left[I^{\otimes 2} \otimes |ar\rangle\langle ar| + I^{\otimes 2} \otimes |da\rangle\langle da| - U \otimes |da\rangle\langle ar| - U^\dagger \otimes |ar\rangle\langle da|\right], \tag{15}$$

and other 1*- and* 2*-local projectors: the role-assigning projectors*

$$\begin{aligned} \Pi_C &:= I - (|r\rangle\langle r|_C + |a\rangle\langle a|_C + |d\rangle\langle d|_C), \\ \Pi_E &:= I - (|0\rangle\langle 0|_{EC} + |1\rangle\langle 1|_{EC}), \\ \Pi_L &:= I - (|0\rangle\langle 0|_L + |1\rangle\langle 1|_L + |?\rangle\langle ?|_L), \end{aligned} \tag{16}$$

the clock start/stop projectors

$$\begin{aligned}\Pi_{start} &:= |r\rangle\langle r| \,, \\ \Pi_{stop} &:= |d\rangle\langle d| \,,\end{aligned} \tag{17}$$

the projector onto the data subspace

$$\Pi_D := |0\rangle\langle 0|_L + |1\rangle\langle 1|_L \,, \tag{18}$$

and the clock-keeping projectors

$$\begin{aligned}\Pi_{clock,D} &:= |r\rangle\langle r| \otimes (I - |r\rangle\langle r|) + |a\rangle\langle a| \otimes (I - |r\rangle\langle r|) + |d\rangle\langle d| \otimes |r\rangle\langle r| \,, \\ \Pi_{clock,?} &:= |r\rangle\langle r| \otimes (I - |r\rangle\langle r|) + |a\rangle\langle a| \otimes (I - |r\rangle\langle r|) + |d\rangle\langle d| \otimes I.\end{aligned} \tag{19}$$

Evidently, Π_{init} and Π_{out} are 3-local, while $\Pi_{prop,U}$ is 4-local (on high-dimensional qudits).

There is also a promise. We are assured that for every instance considered either (1) there exists a state $|\psi_{sat}\rangle$ on n 17-dimensional qudits such that $\Pi_i|\psi_{sat}\rangle = 0$ for all i, or (2) $\Sigma_i \langle\psi|\Pi_i|\psi\rangle \geq 1/poly(n)$ for all $|\psi\rangle$.

The goal is to output "YES" if (1) is true, or output "NO" otherwise.

This definition can be summarized as follows:

Linear-Clock-Ternary-QSAT

Input:	An integer n, and a set of projectors $\{\Pi_i\} \subseteq \{\Pi_{\text{init}}, \Pi_{\text{out}}, \Pi_{\text{prop},\mathcal{G}}\}^{poly(n)}$, where $\mathcal{G} \in \{H, HT, (H \otimes H)\text{CNOT}\}$ and each projector acts nontrivially on at most four 17-dimensional qudits.
Promise:	Either (1) there exists an n-qudit state $|\psi_{sat}\rangle$ such that $\Pi_i|\psi_{sat}\rangle = 0$ for all i, or (2) $\Sigma_i \langle\psi|\Pi_i|\psi\rangle \geq 1/poly(n)$ for all $|\psi\rangle$.
Goal:	Output "YES" if (1) is true, or output "NO" otherwise.

Let us note a few things about Definition 2.2. First, observe that each of the individual projectors that compose O_{init}, $O_{prop,U}$, and O_{out} belongs to the set $\mathcal{P}$ of Definition 2.6. As we will discuss in Section 3.4, this suffices for the quantum algorithm and it is not necessary to demonstrate that Π_{init}, $\Pi_{prop,U}$, and Π_{out} are also elements from this set. Second, the projectors Π_{start}, Π_{clock}, and Π_{stop} that in Section 2.3.2 were independent, have now been incorporated into the definitions of the projectors Π_{init}, Π_{prop} and Π_{out} (Π_{clock} has been renamed to $\Pi_{clock,D}$). Additionally, we include a new term $\Pi_{clock,?}$ to restrict the allowed states of the clock register whenever a logical qudit is undefined. Lastly, let us point out the differences between this definition and the one of Meiburg [30]. Here, we use a three-state clock instead of the two-state *unary clock* encoding in order to fix some loose ends.[15] This increases the qudit dimensionality from 13 to 17 when compared to the unary clock encoding. This repairs an error in Ref. [30], which lacks the projectors that require the clock register to only represent legal clock states.

Another observation is that the unitaries U within the $\Pi_{prop,U}$ clauses are not exactly from the Clifford+T gate set like in Section 2.3.2, but from a slight variation where T and CNOT are also accompanied by a Hadamard gate. We use this set for its property that each unitary necessarily changes computational basis states. We remark that while the potentially satisfiable instances encode circuits using this set, the algorithm that decides these instances by executing the circuits can be actually thought of as using the Clifford+T gate set. Thus, our claim that the problem is in BQP_1 with the Clifford+T gate set is accurate.

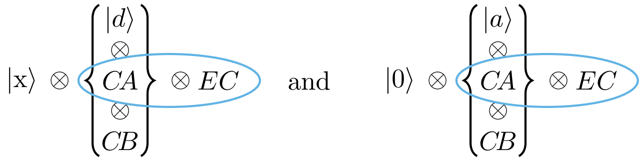
Now, let us analyze these projectors in greater detail and determine which states lie in their null spaces.

3.2.1 Initialization and termination

The Π_{init} term of Eq. (12) is a projector acting on three qudits. In the first line, the Π_L, Π_C and Π_E projectors demand that these qudits serve the roles of logical, clock, and endpoint qudits respectively. Then, the term Π_{start} demands that the clock qudit cannot be $|r\rangle$. Finally, the last term of the line, $(I - |0\rangle\langle 0|)_1 \otimes |a\rangle\langle a|_2$, corresponds to the initialization of the logical qudit (similar to P_{init} in Eq. (6)), requiring that when the

[15] Preliminary results suggest that Bravyi's 4-dimensional clock encoding could also be used to achieve smaller locality at the cost of greater qudit dimensionality. We decided to use the three-state clock encoding as it simplifies the proofs of later sections.

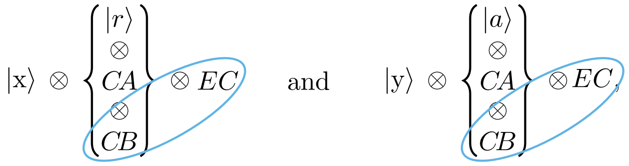
clock qudit is $|a\rangle$, the logical qudit is $|0\rangle$. The projector in the second line demands that the EC subspace of the endpoint qudit forms a $|\Phi^+\rangle$ Bell pair with the CA endpoint of the clock qudit. Considering the demands of all these projectors, one can show that the states


$|\mathrm{x}\rangle \otimes \left\{ |d\rangle \otimes CA \otimes CB \right\} \otimes EC$
and
$|0\rangle \otimes \left\{ |a\rangle \otimes CA \otimes CB \right\} \otimes EC$

(20)

are the only states that satisfy all clauses of Π_{init}. Here, $\mathrm{x} \in \{0, 1, ?\}$ and the blue ellipse represents that the subspaces are maximally entangled. Observe that in the first state, the state of the clock qudit alone suffices to satisfy the clause and the constraint on the logical qudit is not enforced.

Π_{out} is defined similarly as Π_{init}, except that it incorporates a projector similar to P_{out} instead of P_{init} and the CA and CB subspaces swap roles. One can show that the satisfying states are


$|\mathrm{x}\rangle \otimes \left\{ |r\rangle \otimes CA \otimes CB \right\} \otimes EC$
and
$|\mathrm{y}\rangle \otimes \left\{ |a\rangle \otimes CA \otimes CB \right\} \otimes EC,$

(21)

for $\mathrm{x} \in \{0, 1, ?\}$ and $\mathrm{y} \in \{1, ?\}$. Note that in this construction, we allow the Π_{out} clause to be satisfied when the logical qudit is in the state $|1\rangle$ or $|?\rangle$.

As we will see shortly, Π_{init} and Π_{out} will still serve the same primary role as in the proof that k-QSAT is hard for QMA_1: to initialize data qudits at the start of the computation, and to verify the state of data qudits at its conclusion. In addition, the Π_{start} and Π_{stop} projectors give Π_{init} and Π_{out} another purpose: to obtain a running clock. We will observe that when both Π_{start} and Π_{stop} are present in an instance, the state where all clock qudits are either $|r\rangle$ or $|d\rangle$ cannot be a satisfying state.

Writing the satisfying states of clauses as in Eq. (20) and Eq. (21) will become cumbersome when considering instances with multiple clauses and satisfying states that are superpositions of qudits. For this reason, from now on, we only write the state of the logical qudit and the primary subspace of the clock qudit—the one that actually represents a state of the clock—and forget about the auxiliary subspaces. We can do so because satisfying clauses does not require mixing the auxiliary subspaces with the actual clock and logical subspaces, and any state whose auxiliary subspaces are not of the form shown in Fig. 4 must violate one of the Π_{init}, Π_{prop} or Π_{out} clauses.

3.2.2 Propagation and clock

The first line of $\Pi_{prop,U}$ demands that two of the four multi-purpose qudits serve as logical qudits and the other two as clock qudits. The fourth line demands that both clock qudits form a Bell pair in the CB subspace of the predecessor and the CA subspace of the successor. The second and third line express the conditions for propagation, where each line specifies different requirements depending on the state of the logical qudits.

First, suppose that none of logical qudits are undefined, i.e. they are in the joint state $|\phi\rangle$ such that $(\langle ?| \otimes I) |\phi\rangle = (I \otimes \langle ?|) |\phi\rangle = 0$. Under these conditions, the term $(\Pi_D \otimes \Pi_D \otimes I^{\otimes 2})$ in the second line is not satisfied, while $(I - \Pi_D \otimes \Pi_D \otimes I^{\otimes 2})$ in the third line is. Then, to satisfy all four lines of the $\Pi_{prop,U}$ clause, the qudits must be in a state that lies in the null space of the term $(\Pi_{work,U} + I^{\otimes 2} \otimes \Pi_{clock,D})$. Since $\Pi_{work,U}$ is identical to $P_{prop,U}$ of Eq. (6), this term requires that there is a usual propagation of the computation, while maintaining a correct form of the clock qudits. The only states that satisfy all terms are

$$|\phi\rangle \otimes |rr\rangle, \qquad |\phi\rangle \otimes |dd\rangle \qquad \text{and} \qquad \frac{|\phi\rangle \otimes |ar\rangle + U|\phi\rangle \otimes |da\rangle}{\sqrt{2}}. \tag{22}$$

On the other hand, if one of the logical qudits in the clause is undefined, we obtain the opposite behavior: $(\Pi_D \otimes \Pi_D \otimes I^{\otimes 2})$ is satisfied, while $(I - \Pi_D \otimes \Pi_D \otimes I^{\otimes 2})$ is not. Then, the satisfying state of the clause

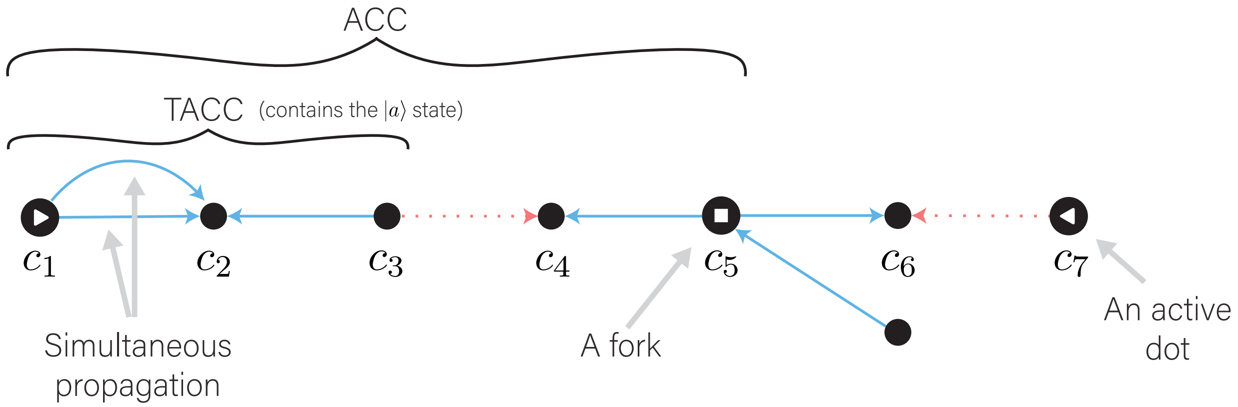


Figure 5: A clock component containing examples of relevant events that may occur in the clock component. The clock component has two active clock chains: one given by c_1 to c_5, and the other by c_7, c_6, c_5. The first chain is truncated due to an undefined clause, and hence c_1 to c_4 forms a TACC. Similarly, the second ACC is also truncated, making c_7 an active dot. For clarity, the endpoint qudits are not drawn.

is that which lies in the null space of $(I^{\otimes 2} \otimes \Pi_{clock,?})$, where $\Pi_{clock,?}$ is similar to $\Pi_{clock,D}$ except that the successor clock qudit must be $|r\rangle$. Then, the only states that satisfy all terms are

$$\{|?0\rangle, |?1\rangle, |0?\rangle, |1?\rangle, |??\rangle\} \otimes \{|rr\rangle, |ar\rangle\}. \tag{23}$$

In contrast to the states that satisfy the clauses with well-defined logical qudits, the satisfying state of clauses with undefined logical qudits consists only of the term previous to the application of the unitary. For ease, we will refer to the clauses with these two different types of logical qudit values as *well-defined* and *undefined* clauses. See also Fig. 3b, where we draw undefined $\Pi_{prop,U}$ clauses as red dotted arrows.

Finally, observe that the state where the clock qudits are set to $|rr\rangle$ is enough to satisfy the $\Pi_{prop,U}$ clause independently from the state of the logical qudits.

3.2.3 Instances

As will be discussed in more detail in the following subsections, showing that LCT-QSAT can be solved in BQP_1 requires us to determine the satisfiability of any possible instance created using a polynomial amount of $\Pi_{init}, \Pi_{prop,U}$, and Π_{out} clauses. Let us briefly present the types of instances that a collection of such clauses may form, and establish several definitions and notation useful for the rest of the text. The analysis of the satisfiability of these instances will, for the most part, be postponed until the next subsection. We note that from now on, we will refer to $\Pi_{prop,U}$ simply as Π_{prop}, as we will generally not be concerned with the associated unitary.

The most general instance we can receive as input is one with no structure whatsoever: one where the qudits of the instance are required to serve multiple roles. In other words, the qudits of the instance may be acted on by multiple Π_C, Π_E, and Π_L clauses. While proved formally in Lemma 3.1, these clauses cannot be satisfied simultaneously as the subspaces are orthogonal to each other and so the instance should be rejected. For instances where all qudits serve a single role, the clock qudits and the Π_{prop} clauses connecting them make an important criterion for their satisfiability. Focusing on these two elements only, an instance can then be thought of as a collection of disjoint directed sub-graphs. We refer to each one of these sub-graphs as a *clock component*, and refer to both the clock component and its logical qudits as a *sub-instance*. We now list some particular arrangements of the clauses within a clock component that are of interest and which affect the satisfiability of the instance. These are also illustrated in Fig. 5.

Definition 3.2 (Forks)**.** *A clock component is said to have a* fork *if there exists a clock qudit that is connected via Π_{prop} clauses to more than two distinct clock qudits.*

Definition 3.3 (Simultaneous propagation)**.** *A clock component is said to have* simultaneous propagation *if there are two or more Π_{prop} clauses acting on the same pair of clock qudits.*

In a clock component, pairs of Π_{init} and Π_{out} clauses and the clauses connecting them define a crucial part of the instance called the *active clock chain*, abbreviated as ACC.

Definition 3.4 (Active clock chain)**.** *In a clock component, an* active clock chain *is a set of clock qudits and Π_{prop} clauses in the one-dimensional path connecting a clock qudit present in a Π_{init} clause to a clock qudit present in a Π_{out} clause, such that no other Π_{init} or Π_{out} clause acts on the clock qudits of this chain.*

An active clock chain receives this name because setting the clock qudits in the chain to either one of the inactive states violates the Π_{start} and Π_{stop} projectors within the Π_{init} and Π_{out} clauses. Therefore, to satisfy the clauses of the chain, at least one of the clock qudits must be in an active state. However, the possible location of the active state in the satisfying state also depends on the existence of undefined clauses within the ACC. This leads us to the definition of *truly active clock chains* (TACCs)—the part of the chain that necessarily contains an active state.

Definition 3.5 (Truly active clock chain)**.** *A* truly active clock chain *is a subset of clock qudits and Π_{prop} clauses within an active clock chain consisting of the clock qudit present in the Π_{init} clause up to (and including) the first clock qudit present in an undefined Π_{prop} clause. If there is no such clause, the truly active clock chain is identical to the active clock chain.*

We define the length L of an ACC and TACC as the number of Π_{prop} clauses within the chain. A chain of length L then involves $L + 1$ clock qudits. It is worth noting that some chains may also have $L = 0$ which may arise in an instance where both a Π_{init} and Π_{out} clause act on the same clock qudit, or when an undefined clause acts on the clock qudit with the starting Π_{init} clause. We will call these chains *active dots*. In this section, active dots and chains of non-zero length behave quite similarly, however, the distinction between the two becomes more relevant in the construction of Section 4.

For the final pieces of notation, we use T instead of L to the refer to the length of a TACC when it is a strict subset of an ACC. Finally, we let q denote the number of logical qudits involved in clauses within a TACC.

Now, let us discuss the satisfiability of the possible input instances.

3.3 Instance satisfiability

In this subsection we formally begin the proof that Linear-Clock-Ternary-QSAT is contained in BQP_1. Here, we focus on determining the satisfiability conditions of all input instances, which will serve as the back-bone of the algorithm presented in the following subsection. We present our findings as a series of lemmas, and remark that we work under the assumption that the instance considered in a lemma has not been decided by any of the previous lemmas. For clarity and continuity, we omit the proofs of the lemmas here, and instead collect them in Appendix A.

In summary, we decide the satisfiability of an instance by evaluating all of its sub-instances and the logical qudits that they may share. An instance is unsatisfiable if at least one of its sub-instances is also unsatisfiable, and conversely, the instance is satisfiable iff all sub-instances are satisfiable. We will show that most sub-instances can be decided based on the arrangement of their clauses. However, there are three types of sub-instances which we do not know how to decide classically and hence make use of a quantum algorithm. The first type of sub-instance is that with a single one-dimensional TACC spanning the whole ACC, properly initialized logical qudits, and a coherent flow of time like that shown in Fig. 3a. These are the sub-instances that can be identified with a quantum circuit acting on a "data" register and a "clock" register finalized by measurements of some of the data qubits, just like in previous uses of the circuit-to-Hamiltonian construction.[16] The other types of instances are those with a single TACC and a coherent flow of time but contain simultaneous propagation clauses. There are two variations as the TACC may either span the whole ACC, or may be truncated due to undefined Π_{prop} clauses (the undefined clauses may only occur after the first simultaneous propagation clause, as the sub-instance would be trivial otherwise). These three sub-instances are shown in Fig. 6.

To begin, when presented with a general instance, the first criterion regarding its satisfiability is that each qudit must serve a single role, e.g. data, clock, or endpoint:

[16] As we will also show at a later time, these instances can encode the computation of a BQP_1 problem. Deciding them classically (either deterministically or probabilistically) would show that this model of classical computation is equivalent to quantum computation with perfect completeness.

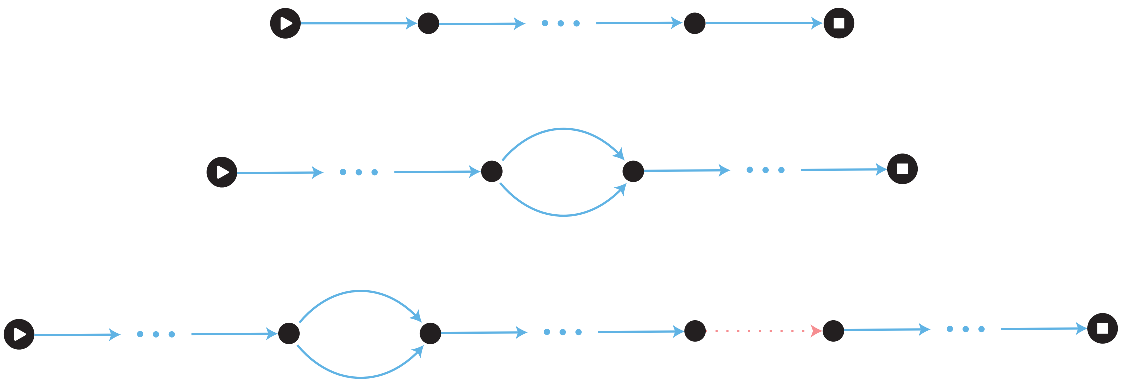


Figure 6: Examples of sub-instances whose satisfiability is determined with a quantum algorithm. Top: Instance with an active clock chain containing no undefined or simultaneous propagation clauses. The length of the chain may be $0 \leq L < \mathcal{O}(n)$. Middle: Instance with simultaneous propagation clauses followed by well-defined clauses. Bottom: Instance with simultaneous propagation clauses followed by both well-defined and undefined Π_{prop} clauses.

Lemma 3.1 (Single-type qudits)**.** *If there is a* 17*-dimensional qudit in the instance acted on by at least two projectors* Π_α *and* $\Pi_{\alpha'}$*, where* $\alpha, \alpha' \in \{C, E, L\}$ *and* $\alpha \neq \alpha'$*, the instance is unsatisfiable.*

Therefore, instances that may have a satisfying state must have qudits that serve a single role at all times. By construction, the next criterion for determining the satisfiability of instances comes from the monogamy of entanglement conditions between clock and endpoint qudits. To properly analyze this structure, we separate the instance into smaller sub-instances by considering the sets of connected clock qudits (the clock components mentioned previously).[17] Clearly, partitioning an instance by considering only its clock qudits may not generate completely disjoint sub-instances. These could still share logical qudits among them. We proceed by first evaluating the satisfiability of individual sub-instances and only then evaluate how these relate to each other. This is reasonable because the satisfying state of a sub-instance determines which of its logical qudits are actually constrained and should therefore be considered "shared".

3.3.1 A clock component and its logical qudits

As mentioned previously and as illustrated by the following four lemmas, the principle of monogamy of entanglement allows us to reject a large number of clock components. Assuming that the clock components have at least one Π_{prop} clause, we can show that the only kind of sub-instances not deemed unsatisfiable by the monogamy condition are those that form a single one-dimensional chain of clock qudits with a unique direction. Besides, if they contain Π_{init} and Π_{out} clauses, these must be positioned at the ends of the chain, with all arrows pointing away from Π_{init} and into Π_{out}. We note that these instances may contain simultaneous propagation clauses, as long as they point in the correct direction. Let us state the lemmas corresponding to these claims, starting with the lemmas regarding the connections of clock qudits for clock components with multiple clock qudits. Afterwards, we briefly discuss the clock components formed by a single clock qudit.

Lemma 3.2 (Clock qudit two neighbor maximum)**.** *If a clock qudit in a clock component is connected to more than two other clock or endpoint qudits, the instance is unsatisfiable.*

Lemma 3.3 (Unique direction of a clock chain)**.** *If a clock qudit in the chain has two successors or two predecessors, the instance is unsatisfiable.*

[17] Although endpoint qudits are also able to connect clock qudits together via Π_{init} and Π_{out} clauses, we do not consider such clock components at the moment to avoid possible confusion. Besides, as will be shown shortly in Lemma 3.4, satisfying these clauses inevitably leads to a violation of monogamy of entanglement.

As promised, these two lemmas reject all clock components that do not form a one-dimensional chain of clock qudits where all Π_{prop} clauses point in the same direction. These clock components may be either in the form of a cycle or a one-dimensional line. However, neither lemma can reject the case where a clock qudit in the chain is connected to its successor and/or predecessor via multiple Π_{prop} clauses (which may have different unitaries and act on different logical qudits). The principle of monogamy of entanglement is incapable of ruling out these instances since each additional Π_{prop} (with the correct direction) does not require entangling new subspaces of the clock qudits. As a result, the same already-entangled subspaces can be reused. Although this behavior is a required feature of the construction for Π_{init} clauses, we cannot prevent the same from happening for Π_{prop} clauses.[18] The stacking of these clauses presents a great downside and an extra difficulty in deciding the satisfiability of an instance. In Ref. [30], Meiburg crucially missed this important event, which as we will see shortly, requires a more comprehensive quantum algorithm than the one presented there. We discuss these instances in more detail in Section 3.3.3, for now, let us present two lemmas regarding the Π_{init} and Π_{out} clauses.

Lemma 3.4 (Unique endpoint qudit for Π_{init}/Π_{out} clauses)**.** *If an endpoint qudit is connected to more than one clock qudit, or connected to a single clock qudit via a Π_{init} and Π_{out} clause, the instance is unsatisfiable.*

Lemma 3.5 (Unique clock qudit for Π_{init}/Π_{out} clauses)**.** *Let c_0 refer to the clock qudit that all Π_{prop} clauses point away from, and c_L the one at the other end of the chain. If a Π_{init} clause acts on a clock qudit that is not c_0, the instance is unsatisfiable. Similarly, if a Π_{out} clause acts on a clock qudit that is not c_L, the instance is unsatisfiable.*

The first lemma shows that satisfiable instances cannot have clock components joined together by endpoint qudits, which is why we ignored these cases in the first place. Furthermore, this lemma also rejects instances where an endpoint qudit is present in both Π_{init} and Π_{out} clauses even if they connect to the same clock qudit. The second lemma rejects instances where Π_{init} or Π_{out} clauses act on the wrong clock qudits. Finally, observe that these lemmas also do not disallow stacking Π_{init} (or Π_{out}) clauses as long as they act on the same clock and endpoint qudit.

To summarize, the sub-instances not rejected by any of the four lemmas above must be those with one-dimensional clock chains where all Π_{prop} clauses point in the same direction. Moreover, if the sub-instance contains Π_{init} or Π_{out} clauses, all Π_{init} clauses must act on c_0 and a single endpoint qudit, and all Π_{out} on c_L and a single endpoint qudit (the endpoint qudits must also be different). In other words, Π_{init} and Π_{out} mark the endpoints of the chain and no other Π_{prop} clause can extend past them. Also, these show that cyclic instances that may be satisfiable cannot have any Π_{init} or Π_{out} clauses.

Now, let us show that clock components without both Π_{init} and Π_{out} clauses, regardless of whether they have undefined or simultaneous propagation clauses, are trivially satisfiable.

Lemma 3.6 (Lack of an ACC)**.** *If a clock component does not have at least one Π_{init} and one Π_{out} clause, the sub-instance is satisfiable.*

Now it should be apparent why the order in which we evaluate the clock component matters. For example, we cannot conclude that a clock component with only a Π_{init} clause has a satisfying state without first checking that the instance is linear, the direction of its Π_{prop} clauses, and the arrangement of the endpoint qudits.

The sub-instances not decided by any of the lemmas above must be those consisting entirely of an ACC which also displays a clear computational flow of time, i.e. they have a start, an end, and all Π_{prop} clauses in between point away from the starting qudit. The ACC however may contain simultaneous propagation clauses. At the moment, we can gather that if these sub-instances are satisfiable, at least one of the clock qudits of the chain must be in the state $|a\rangle$. This is because setting all clock qudits of the chain to $|r\rangle$ violates the Π_{init} clause, setting them to $|d\rangle$ violates the Π_{out} clause, and the Π_{clock} (both $\Pi_{clock,D}$ and $\Pi_{clock,?}$) clauses penalize any direct clock transitions from $|d\rangle$ to $|r\rangle$ and vice versa.

For a moment, consider one of these sub-instances without simultaneous propagation clauses and imagine we had defined the behavior of Π_{prop} clauses in a single way (as in Eq. (6)) instead of conditioning on the state

[18] To prove BQP_1-hardness it is necessary to stack multiple Π_{init} clauses on the same clock qudit in order to initialize all of the required logical qudits; see Fig. 3a.

of the logical qudit. The presence of the $|a\rangle$ state would trigger the propagation of computation (in order to satisfy the Π_{prop} clauses) through the whole chain, and the only satisfying state would be a history state acting on both initialized and free logical qudits. As mentioned previously, the free logical qudits give one the opportunity to guess a solution and the sub-instance therefore results in a difficulty likely greater than BQP_1. Conditioning on the state of the logical qudits thus offers us other possible satisfying assignments. At this moment however, there is ambiguity on which logical qudits must be initialized or may remain free. The following proposition offers a way forward:[19]

Proposition 3.1 (Initialization of logical qudits)**.** *Let $c_0, \ldots, c_L$ with $L \geq 0$ be the clock qudits of an ACC. Let c_0 be the qudit present in the Π_{init} clauses and c_L the one in the Π_{out} clauses. A state of the qudits of the sub-instance where c_0 is not $|a\rangle$ in any basis state of the superposition is not a satisfying state.*

This proposition implies that if a satisfying state of the sub-instance exists, c_0 must be $|a\rangle$ at some point in "time" (must be $|a\rangle$ in at least one of the basis states of the satisfying superposition). Consequently, at this "time", all logical qudits present in the Π_{init} clauses attached to c_0 are required to be in the state $|0\rangle$. The other logical qudits remain free. This then allows us to tell apart the Π_{prop} clauses that must be well-defined from those that may not.

Proposition 3.1 also requires us to make a decision. Recall that since sub-instances may share logical qudits, it is possible that a sub-instance constrains a logical qudit to be $|0\rangle$, but in a different sub-instance this logical qudit is not initialized. Should we think of the logical qudit as initialized to $|0\rangle$ in both sub-instances? Or should the constraint be considered locally and let the qudit be $|0\rangle$ in the first sub-instance and free in the second? As is discussed in more detail in Appendix C, while both ways yield the same result, we choose to perform this operation globally as it simplifies the remaining parts of the analysis. This way, when we encounter an undefined clause while analyzing a clock component, it ensures us that it is not initialized by any clock component. As a result, the TACCs are fixed which simplifies the analysis of instances. This will become even clearer when considering the classical algorithm of Section 3.4, in particular steps (3) and (4).

Importantly, Proposition 3.1 also shows that it is not possible to abuse the undefined states and claim that all instances are satisfied by setting all logical qudits to $|?\rangle$. Simply, this assignment would not be consistent with the state of the clock qudits. Finally, while this proposition demonstrates that TACCs must have at least one active state, we now show that they must have exactly one such state.

Proposition 3.2 (Unique active state in a TACC)**.** *Let $c_0, \ldots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. A state in which at any given time there are two clock qudits c_a and c_b with $0 \leq a < b \leq T$ in the state $|a\rangle$ is not a satisfying state.*

These two propositions show that satisfying the clauses within TACCs require that there is exactly one active clock qudit, which is in agreement with the desired encoding of the clock in Eq. (5). As a consequence of this statement, satisfying the Π_{prop} clauses within the chain (which are all well-defined) implies that the satisfying state must demonstrate the evolution of a state under the circuit $U = U_T \ldots U_1$ (see Eq. (22)). Furthermore, if the TACC terminates with a Π_{out} clause, then these are the same conditions that apply on the clock chain of instances like Fig. 2, and we can conclude that, if satisfiable, it must be uniquely satisfied by a history state. However, it is not entirely clear if the same holds for TACCs that terminate because of an undefined clause. We now show that these chains are always satisfied by a history state that is truncated at the time the computation reaches the undefined clause.

Lemma 3.7 (Satisfying state of a truncated ACC)**.** *Let $c_0, \ldots, c_L$ be the clock qudits of an ACC, and suppose $\Pi_{prop}^{(c_T, c_{T+1})}$ with $0 \leq T < L$ is the first propagation clause that acts on a free logical qudit l_u. The clock qudits $c_0, \ldots, c_T$ then form a TACC. Furthermore, assume the TACC has no simultaneous propagation clauses, and let $U_1, \ldots, U_T$ be the unitaries associated with the Π_{prop} clauses of this chain. Finally, let S denote the set of all logical qudits of the sub-instance, and D with $|D| = q$ the subset of these qudits that are initialized by the Π_{init} clause on c_0. Under these conditions, the sub-instance is trivially satisfiable and the satisfying state is given by*

[19] We state this as a proposition rather than a lemma as it concerns the form of a potential satisfying state, and not a condition about the satisfiability of the instance.

$$|\psi_{thist}\rangle := \left[\frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t \dots U_0 \left|0\right\rangle^{\otimes q} \otimes |?\rangle_{l_u} \otimes |\underbrace{d\dots d}_{t} a_t \underbrace{r\dots r}_{T\text{-}t}\rangle \right] \otimes |\phi\rangle_{S\setminus D} \otimes |\underbrace{r\dots r}_{L\text{-}T}\rangle \,, \tag{24}$$

where $|\phi\rangle$ is an arbitrary state of the logical qudits of the sub-instance not present in clauses within the TACC.

The satisfying state shows that the clauses within the TACC do enforce constraints on their corresponding logical qudits by first requiring that they are initialized to $|0\rangle$ and then acted on by unitary operations. Conversely, the Π_{prop} and Π_{out} clauses outside the truly active clock chain do not constrain their logical qudits. Hence, when considering logical qudits present across multiple clock components, the ones that should actually be considered as "shared" are those belonging to clauses stemming from truly active clock chains. This also suggests that the classical algorithm presented in the following section should first evaluate the satisfiability of the clauses that share logical qudits before deciding that the sub-instances with truncated ACCs (and no simultaneous propagation clauses within the TACC) are trivially satisfiable.

The previous lemma is the last statement we can make about deciding the satisfiability of a sub-instance based on the structure of its clock component. This is because if the sub-instance has not been decided by the lemmas above and its logical qudits do not conflict with other sub-instances, they must be one of the three types of "quantum" sub-instances shown in Fig. 6. Indeed, if not rejected by the lemmas above, the clock component of the sub-instance must form a one-dimensional chain with the following properties:

- There is at least one Π_{init} clause, which must act on a clock qudit at one of the ends of the chain. If there are multiple such clauses, they must all act on this same qudit, as well as use the same endpoint qudit.
- There is at least one Π_{out} clause, which must act on the qudit at the other end of the chain. If there are multiple such clauses, they must all act on this same qudit, as well as use the same endpoint qudit. The endpoint qudit must be different than the one used by the Π_{init} clauses.
- There is at least one Π_{prop} clause, which must point away from the clock qudit with the Π_{init} clauses and towards the qudit with the Π_{out} clauses. If there are multiple π_{prop} clauses, they must all have this same direction.

Additionally, it is possible for these sub-instances to have multiple Π_{prop} clauses involving the same pair of clock qudits, and if this is true, they might also have undefined clauses after the simultaneous propagation. These are the three types of "quantum" sub-instances just mentioned.

Remark 1. *Among these "quantum" sub-instances, there are two that are actually trivially decidable. These are the sub-instances with no undefined or simultaneous propagation clauses, where the logical qudits present in Π_{out} clauses are (1) initialized but not acted by any Π_{prop} clause, or (2) not initialized.*[20] *It is not difficult to see that the first type is trivially unsatisfiable, while the second type is trivially satisfiable. Despite their triviality, they will be decided by the quantum algorithm. This aims to simplify the classical algorithm and minimize the number of cases it needs to handle.*

Single clock qudit components

Before moving forward, let us provide some assurance that clock components consisting of a single clock qudit can also be decided using the lemmas above or must have a structure resembling that of longer chains that are decided by the quantum subroutine.

By definition, this clock component can only be acted on by either Π_{init} and/or Π_{out} clauses. Lemma 3.2 states that the clock qudit must be connected to at most two endpoint qudits. Then, due to the lack of Π_{prop} clauses, Lemmas 3.3 and 3.5 do not apply. If the sub-instance is not decided by the other lemmas above, then it must be acted on by both Π_{init} and Π_{out} clauses, where there is a unique endpoint qudit for the Π_{init} clauses and a separate endpoint qudit for Π_{out} clauses. For such a sub-instance, the only valid state of the clock qudit must be $|a\rangle$, implying that the logical qudits present in the Π_{init} clauses must be

[20] By construction, the second type of sub-instance cannot be acted by a Π_{prop} clause as otherwise it would create an undefined clause.

initialized to $|0\rangle$ and the logical qudits present in the Π_{out} clauses to $|1\rangle$. The instance then falls under two categories: either there is a logical qudit present in both Π_{init} and Π_{out} clauses, or there is none. These are the two sub-instances mentioned in Remark 1. As such, after verifying that their shared logical qudits (if any) do not create any conflicts, they will be decided by the quantum algorithm.

3.3.2 Shared logical qudits

As mentioned previously, partitioning the instance into sub-instances based on connected clock qudits may not result in completely disjoint sub-instances as some of the logical qudits may overlap. We now determine the joint satisfiability of these sub-instances.

To this end, we consider one of the shared logical qudits and determine if there exists a state of the sub-instances such that all clauses in which it appears can be satisfied simultaneously. The observation at the end of the previous subsection allows us to make a simplification. There, we noted that the logical qudits of a sub-instance that are actually constrained are those present in the clauses of the TACC. Thus, the relevant shared logical qudits are exclusively those that are present in clauses from multiple TACCs.

The satisfiability of the joint sub-instances depends on the clauses applied to the logical qudit. Let us now present the lemmas about the satisfiability of these cases, starting with the lemma where the shared qudit is acted by a unitary gate from a clock component.

Lemma 3.8 (No Π_{prop} clause on shared qudit)**.** *If a logical qudit is present in a Π_{prop} clause of a TACC and this qudit is also acted on by a clause from a different TACC, the instance is unsatisfiable.*

This lemma shows the unsatisfiability of shared logical qudits that are acted on by a single or multiple unitaries. The remaining cases are those were the logical qudit is acted on only by Π_{init} clauses, Π_{out} clauses, or a mix of both types of clauses.[21] The satisfiability of these cases is given by the following lemmas.

Lemma 3.9 (No Π_{init} and Π_{out} clauses on shared qudit)**.** *If a logical qudit is acted on by a Π_{init} clause from a clock component and a Π_{out} clause from another clock component, the instance is unsatisfiable.*

Lemma 3.10 (Either all Π_{init} or all Π_{out} clauses)**.** *If a logical qudit is present in clauses from multiple clock components but the clauses are either all Π_{init} or all Π_{out} clauses, there exists a state of the logical qudit that simultaneously satisfies these clauses.*

These lemmas conclude the analysis of the shared logical qudits. Now, let us finally discuss the unresolved issue of simultaneous propagation clauses.

3.3.3 Simultaneous propagation clauses

None of the lemmas in Section 3.3.1 which analyze the clock component in the sub-instance are able to decide if simultaneous propagation clauses are trivially satisfiable or unsatisfiable. At best, Lemma 3.3 establishes that simultaneous propagation clauses create unsatisfiable instances if the direction of one of the simultaneous Π_{prop} clauses is incorrect. The same can be said about the lemmas in Section 3.3.2. Even in the case where the unitaries associated with simultaneous Π_{prop} clauses act on the same logical qudits, Lemma 3.8 is unable to determine the unsatisfiability of the instance since the clauses belong to the same clock component. We now show the necessary and sufficient conditions for which these instances may admit a satisfying state.

Proposition 3.3. *Let $c_0, \ldots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length, and suppose k distinct Π_{prop} clauses act on c_t and c_{t+1} with $0 \leq t < T$ and an arbitrary pair of logical qudits. The logical qudits may or may not be repeated. Let $U_{t+1,1}, \ldots, U_{t+1,k}$ denote the unitaries associated with these k clauses, and $|\phi_t\rangle$ be the state of the logical register at time t, i.e. $|\phi_t\rangle = U_{t,1} \ldots U_{0,1} |0\rangle^{\otimes q}$ with $U_{0,1} = I$. All k Π_{prop} clauses can be satisfied simultaneously if and only if $U_{t+1,i} |\phi_t\rangle = U_{t+1,j} |\phi_t\rangle$ for all $i, j \in [k]$.*

[21] Observe that the case where the logical qudit is only acted by Π_{out} clauses can occur if the ACCs of the clock components do not have undefined clauses.

This proposition shows that to determine the satisfiability of simultaneous propagation clauses, we can compare the action of each unitary on the state $|\phi_t\rangle$. However, since $t \in \mathcal{O}(n)$ and the set of gates used is universal for quantum computation, it is unlikely that we can evaluate this classically.

Another point to consider is whether there are actually sub-instances where two or more simultaneous propagation clauses can be satisfied given that these can only be of three types: H, HT, or $(H \otimes H)$CNOT. If the gates act on different qudits, then this is not too difficult to imagine. In the case where the unitaries act on at least one common qudit, observe that $H_0 \otimes I_1 |\phi_t\rangle = (H_0 \otimes H_1)\text{CNOT}_{0,1} |\phi_t\rangle$ if $|\phi_t\rangle = |0\rangle \otimes |H_+\rangle$ and $|H_+\rangle$ is the $+1$-eigenstate of the Hadamard gate.

3.4 Algorithm and correctness

In this subsection we present the hybrid algorithm showing that LCT-QSAT can be decided in polynomial time using both a classical and quantum computer while adhering to the soundness and completeness conditions mentioned in Definition 2.2. In other words, we show that LCT-QSAT is contained within BQP_1.

In summary, the first part of the algorithm determines the satisfiability of instances by following the lemmas of Section 3.3. As mentioned there, the satisfiability of an instance is first (and most of the time) dictated by the arrangement of its clauses. Since evaluating the clauses of an instance consist of simple lookups in a graph of polynomial size, these steps can be efficiently performed with a classical computer. If the algorithm does not reject (meaning it doesn't encounter any unsatisfiable clauses), then we know for certain that the sub-instances within have proper structure and the shared logical qudits create no conflicts. These sub-instances then fall under two categories: either they are trivially satisfiable (those that lack an ACC or have a truncated ACC) or those we have identified that use a quantum algorithm. The algorithm analyzes each sub-instance, and makes use of a quantum subroutine if it happens to be of the second kind. Lastly, if the classical and quantum algorithm do not reject, the algorithm accepts the instance.

Before describing the algorithm in detail, let us make some observations. To meet the completeness and soundness conditions of BQP_1, we have to make sure that the algorithm does not reject satisfiable instances and does not accept unsatisfiable instances with high probability. We will prove this in more detail later, but this will partly follow from the correctness of the lemmas and propositions discussed in Section 3.3. Furthermore, perfect completeness requires that the algorithm consistently performs operations with perfect accuracy. This is of most concern in the quantum algorithm, which we keep in mind in its design. Finally, we note that to demonstrate that the problem is in BQP_1, the algorithm only needs to decide if the instance is satisfiable—it is not required to produce the satisfying state.

3.4.1 Classical algorithm

Our algorithm is the following:

(1) *Verify that all qudits in the instance serve a unique role.* For every qudit in the instance, check whether it is acted on by Π_C, Π_E, or Π_L clauses. If a qudit is acted on by more than one type of clause, reject. Otherwise, label the qudit as either C, E, or L depending on the type of clause acting on it. Correctness is given by Lemma 3.1.

(2) *Consider the sub-instances that make up the instance. Each sub-instance is composed by a clock component (but now also considering all connections made by endpoint qudits) and its associated logical qudits.* For every sub-instance, perform the following checks:

(2.1) If a clock qudit is connected to more than two other distinct clock or endpoint qudits, reject. Correctness is given by Lemma 3.2.

(2.2) If a clock qudit in the chain has two successors or two predecessors, reject. Correctness is given by Lemma 3.3.

(2.3) If an endpoint qudit is connected to more than one clock qudit, reject. Similarly, if an endpoint qudit is connected to the same qudit via both a Π_{init} and Π_{out} clause, reject. Correctness is given by Lemma 3.4.

(2.4) If a Π_{init} clause acts on a qudit other than the one at the beginning of the chain, reject. Similarly, if a Π_{out} clause acts on a qudit other than the one at the end of the chain, reject. Correctness is given by Lemma 3.5.

(2.5) If there are no Π_{init} and Π_{out} clauses, ignore all clauses of the sub-instance for the rest of the algorithm. Correctness is given by Lemma 3.6.

(3) *Identify the undefined Π_{prop} clauses.* For every logical qudit, check whether it is acted on by at least one Π_{init} clause. If it is not, mark all Π_{prop} clauses this qudit is part of as undefined.

(4) *Now, the satisfiability of the instance depends on the shared logical qudits of clock components. Importantly, only those within TACCs are considered.* Identify the logical qudits involved in truly active clock chains of multiple clock components. For every one of these qudits perform the following checks:[22]

(4.1) If the qudit is acted on by a Π_{prop} clause from a clock component, and is also present in another clause stemming from a different clock component, reject. Correctness is given by Lemma 3.8.

(4.2) If the qudit is acted on by both Π_{init} and Π_{out} clauses, reject. Correctness is given by Lemma 3.9.

(5) *Now that we have verified the logical qudits of truncated active clock chains, we can safely ignore them.* For every sub-instance, check whether it contains an undefined propagation clause. If it does and there are no simultaneous propagation before this clause, ignore the clauses of the sub-instance. Correctness is given by Lemma 3.7.

(6) *By this point in the algorithm, all trivial sub-instances have been decided, and the logical qudits of those that remain do not create conflicts.* For every remaining sub-instance, run the quantum algorithm to determine its satisfiability.

(7) Accept the instance. *If the algorithm reaches this step, then none of the sub-instances were rejected in the previous steps so we accept the instance.*

3.4.2 Quantum algorithm

As mentioned previously, the classical algorithm tasks the quantum subroutine to decide three types of sub-instances: those with unique active clock chain that have (1) no simultaneous propagation clauses and well-defined Π_{prop} clauses, (2) simultaneous propagation clauses followed by well-defined Π_{prop} clauses, or (3) simultaneous propagation clauses followed by one or multiple undefined Π_{prop} clauses, shown in Fig. 6. As the clauses outside the TACC can be satisfied by choosing the state of the clock qudits appropriately, the goal of the quantum algorithm is to decide if there exists a state that satisfies all constraints of the TACC. For the remainder of this discussion, we let the TACC be of length T, act on q logical qudits, and let $k_{t,t+1}$ denote the number of simultaneous propagation clauses involving clock qudits c_t and c_{t+1} with $t \in [T]$. We note that since the logical qudits of the TACC cannot be $|?\rangle$, in the actual quantum circuits they are simply referred to as logical *qubits*.

Let us consider the first type of sub-instance which does not contain simultaneous propagation clauses, i.e. $k_{t,t+1} = 1$ for all $t \in [T]$. For these sub-instances, we know that the Π_{init} and Π_{prop} clauses of the chain are uniquely satisfied by the history state corresponding to the circuit expressed by the TACC (mentioned below Eq. (8)). Therefore, the algorithm's only task is to determine if this state also satisfies the Π_{out} clauses.

One verification method, analogous to Bravyi's algorithm in Section 2.3.1, involves constructing the history state and measuring the eigenvalue of the Π_{out} clauses to verify if they can be satisfied. A quick inspection shows that this method is wasteful as the Π_{out} clauses are only concerned with the state of the qudits at the end of the computation, i.e. only one out of the $T + 1$ basis states in the history state's superposition is relevant. A simpler approach is to create the quantum circuit $U = U_T \dots U_1$ expressed by the TACC, execute it on the initial state $|0\rangle^{\otimes q}$, measure the state of the logical qubits present in the Π_{out} clauses at the end of the computation, and flip the measurement outcomes. If any of these result in "1", we reject. To see why these methods are equivalent, consider a Π_{out} clause that acts

[22] A step to include Lemma 3.10 is not necessary as it requires no action.

on logical qudit l_{out}. The probability of measuring eigenvalue 1 with Bravyi's method (see Eq. (2)) is $\Pr(\text{outcome } 1) = \langle\psi_{hist}|\,\Pi_{out}\,|\psi_{hist}\rangle = (T+1)^{-1}\,\langle 0|^{\otimes q}\,U_{0,1}^{\dagger}\ldots U_{T,1}^{\dagger}\Pi^{(0)}U_{T,1}\ldots U_{0,1}\,|0\rangle^{\otimes q}$, where we observed that all other terms of Π_{out} have no contribution, and defined $\Pi^{(0)} := |0\rangle\langle 0|_{l_{out}} \otimes I_{rest}$. It is then apparent that this quantity is proportional to the probability that l_{out} is measured at the end of the circuit and yields outcome "0". We flip the measurement outcomes in the alternative method so in both cases, observing "1" signifies a clause violation.

Unfortunately, for the second and third type of sub-instance, these procedures cannot apply directly since there is no longer a unique history state and the simultaneous Π_{prop} clauses should also be verified. To resolve this, let us first develop a method similar to Bravyi's that works for these cases, and then determine a more efficient way of implementing it that avoids creating the full history state. As we will soon see, establishing the equivalence between the two methods is quite useful, as proving the correctness of the algorithm (completeness and soundness) is easier in Bravyi's approach.

Bravyi's Approach

Although sub-instances may now contain multiple history states, Proposition 3.3 shows that if the sub-instance is satisfiable, there is a single history state independent of the propagation "path" chosen. We can thus define a "privileged" history state

$$|\psi_{phist}\rangle := \frac{1}{\sqrt{T+1}}\sum_{t=0}^{T} U_{t,1}\ldots U_{0,1}\,|0\rangle^{\otimes q} \otimes |\underbrace{d\ldots d}_{t}\,a_t\,\underbrace{r\ldots r}_{T-t}\rangle\,, \tag{25}$$

where the computation is performed with unitaries $U_{t,1}$ for all $t \in [T]$ and use this state to test whether the other simultaneous Π_{prop} clauses are also satisfied by this state. We can then design the quantum algorithm to measure the eigenvalue of these clauses on $|\psi_{phist}\rangle$, and at the end, measure the eigenvalues of the Π_{out} clauses that form part of the TACC (if any).

Our Approach

With the intention of building a simpler approach that does not require building a history state, consider the following claim.

Claim 3.1. *Let $\Pi_{prop}^{(c_t,c_{t+1})}$ be a clause acting on clock qudits c_t and c_{t+1} with associated unitary $U_{t+1,j}$, where $j \in [k_{t,t+1}]$. Measuring the eigenvalue of this clause on the state $|\psi_{phist}\rangle$, is equivalent to applying circuit $\mathcal{C}$ of Fig. 7 to the state $|\phi_t\rangle$ and an ancilla qubit in the state $|0\rangle$. Here, $|\phi_t\rangle$ is the state of the data register at time t, i.e. $|\phi_t\rangle = U_{t,1}\ldots U_{0,1}\,|0\rangle^{\otimes q}$ with $U_{0,1} = I$. Furthermore, this circuit can be simulated perfectly by a circuit $\mathcal{C}_{\mathcal{G}}$ that uses a constant number of gates from the set $\mathcal{G} = \{H, T, \mathrm{CNOT}\}$ and an additional ancilla qubit.*

Thus, we can build a quantum circuit that consists of evolving the state according to the circuit expressed by the TACC, but if it encounters simultaneous propagation clauses ahead, it applies circuit $\mathcal{C}_{\mathcal{G}}$ instead. If the measurement of the ancilla yields outcome "1", we reject the instance. We would like the algorithm to continue repeating this pattern of updating the current state and performing checks when necessary, however, there is one nuisance that we must address. If the instance is unsatisfiable and one of the checks mistakenly passes (as could be the case if the simultaneous unitaries produce similar states), the post-measurement state becomes $(I - \Pi_{prop})\,|\psi_{phist}\rangle$ (in Bravyi's description). If the algorithm is allowed to continue, the following checks would not use the privileged history state, but rather this perturbed state. Although this might still be permissible by the soundness condition, for ease of analysis, we design the algorithm to avoid it altogether. Therefore, after applying a check and measuring the ancilla qubit, we simply restart the algorithm, with the condition that in this new iteration, the algorithm skips all previously performed checks.

Algorithm

Explicitly, our quantum algorithm is the following:c

(6.1) Initialize a register containing $q+2$ qubits, each initialized to $|0\rangle$. *The first q qubits correspond to the logical qudits of the TACC, while the other two are the ancillas used in $\mathcal{C}_{\mathcal{G}}$.*

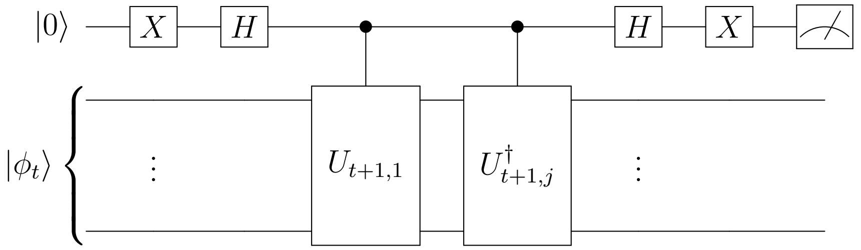


Figure 7: Circuit $\mathcal{C}$ that evaluates whether propagating $|\phi_t\rangle$ with $U_{t+1,1}$ is the same as propagating with $U_{t+1,j}$. Here, $U_{t+1,1}$ and $U_{t+1,j}$ act nontrivially on at most two qubits of the data register. This circuit is equivalent to measuring the eigenvalue of projector Π_{prop} (with unitary $U_{t+1,j}$) on $|\psi_{phist}\rangle$. If $U_{t+1,1}$ and $U^\dagger_{t+1,j}$ are unitaries from the set $\mathcal{G}_8$, the circuit can be perfectly simulated with this set.

(6.2) For every $t \in [T]$, do the following:

(6.2.1) If $k_{t,t+1} > 1$:

- Choose an index $1 < j \leq k_{t,t+1}$. Evaluate circuit $\mathcal{C}_\mathcal{G}$ using $U_{t+1,j}$.
- Remove the Π_{prop} clause with associated unitary $U_{t+1,j}$ from the sub-instance.
- If the measurement outcome is "1", reject. Otherwise, go back to step (6.1).

(6.2.2) Otherwise, apply $U_{t+1,1}$ to the data register to obtain $|\phi_{t+1}\rangle$.

(6.3) *At this step, the algorithm has produced the state* $|\phi_T\rangle$. If there's at least one Π_{out} clause involving c_T, do the following:

(6.3.1) Choose one of the Π_{out} clauses and measure the corresponding data qubit. Flip the resulting outcome.

(6.3.2) Remove this Π_{out} projector from the sub-instance.

(6.3.3) If the outcome is "1", reject. Otherwise, go back to step (6.1).

(6.4) Return to the classical algorithm.

3.4.3 Runtime

For an instance with n qudits and $\mathcal{O}(poly(n))$ clauses, the algorithm reaches a decision in $\mathcal{O}(poly(n))$ time.

In detail, steps (1), (3), and (4) require iterating over the qudits of the instance (either all of them or some particular subset). For each qudit, the algorithm performs checks based on the clauses acting on this qudit. Given that there are n qudits and $\mathcal{O}(poly(n))$ clauses acting on each qudit, each step takes $n \cdot \mathcal{O}(poly(n)) = \mathcal{O}(poly(n))$ time. Step (2) iterates over all sub-instances. Observe that because each sub-instance must contain at least one qudit, the total number of sub-instances is $\mathcal{O}(n)$, each composed of $\mathcal{O}(n)$ qudits. For each sub-instance, in steps (2.1)–(2.3), the algorithm iterates over all clock and endpoint qudits and searches over the clauses acting on these qudits. These steps also take $\mathcal{O}(n) \cdot \mathcal{O}(poly(n)) = \mathcal{O}(poly(n))$ time. Steps (2.4) and (2.5) simply iterate over the clauses of a sub-instance, thus requiring $\mathcal{O}(poly(n))$ time. Then, it is apparent that step (2) takes $\mathcal{O}(poly(n))$ time. In step (5), the algorithm once again iterates over all sub-instances and searches over the clauses present in the sub-instance. If we suppose that removing a clause takes $\mathcal{O}(1)$ time, step (5) can be completed in $\mathcal{O}(poly(n))$ time.

Now, let us consider the runtime of the quantum subroutine used in step (6). The input to this quantum subroutine is a TACC of length $T \in \mathcal{O}(n)$ and possibly $\mathcal{O}(poly(n))$ simultaneous propagation clauses acting on qudits c_t and c_{t+1} for any $t \in [T]$. Step (6.1) takes $\mathcal{O}(n)$ time, since $d \in \mathcal{O}(n)$. In (6.2), to progress from time t to $t+1$, the algorithm applies circuit $\mathcal{C}_\mathcal{G}$ to the state $|0\rangle \otimes |\phi_t\rangle$ $\mathcal{O}(poly(n))$ times. Moreover, after each application, the algorithm restarts (with one less clause) and must again produce $|\phi_t\rangle$ by applying $\mathcal{O}(n)$ unitaries to the initial state $|0\rangle^{\otimes q}$. Thus, the step $t \to t+1$ takes $\mathcal{O}(poly(n)) \cdot \mathcal{O}(n) = \mathcal{O}(poly(n))$ time.

Then, since $T \in \mathcal{O}(n)$, step (6.2) takes $\mathcal{O}(poly(n))$ time. Finally, note that in step (6.3), the algorithm must perform $\mathcal{O}(poly(n))$ measurements. Similar to the previous step, after each measurement, the algorithm restarts (with one less Π_{out} clause) and must again create $|\phi_T\rangle$ by applying $T \in \mathcal{O}(n)$ unitaries. This step then takes $\mathcal{O}(poly(n))$ time. This shows that the quantum algorithm then takes $\mathcal{O}(poly(n))$ time, and considering that it may be used $\mathcal{O}(poly(n))$ times, all of step (6) takes $\mathcal{O}(poly(n)) \cdot \mathcal{O}(poly(n))$ time. This concludes that the algorithm is efficient.

3.4.4 Completeness and soundness

To prove the correctness of the algorithm above, we have to show that for any input instance x of LCT-QSAT, the algorithm presented above can distinguish between the cases where x is a yes-instance or a no-instance. To reiterate, x is a yes-instance if there exists a state $|\psi_{sat}\rangle$ such that for all projectors $\{\Pi_i\}$ that compose the instance, $\Pi_i |\psi_{sat}\rangle = 0$ for all i. On the other hand, x is a no-instance if for all possible states, the sum of the constraint violations is not too small, i.e. $\sum_i \langle\psi| \Pi_i |\psi\rangle \geq 1/poly(n)$ for any state $|\psi\rangle$. Furthermore, to meet the completeness and soundness conditions associated with BQP_1, we have to show that all yes-instances are always labeled as "satisfiable" by the hybrid algorithm, and no-instances are labeled as "unsatisfiable" most of the time.

In Section 3.3, we provided an exhaustive list of instances that can decided based on their structure. As the classical algorithm is based on this list, the algorithm never rejects instances with a valid structure and always rejects instances whose structure implies two or more clauses cannot be satisfied simultaneously by any state. If the classical algorithm is able to determine the instance, then both the completeness and soundness conditions are clearly satisfied. However, if the instance is not decided by the classical algorithm, there must be at least one sub-instance that requires the quantum subroutine.

Quantum sub-instances

Importantly, the classical algorithm ensures that none of the logical qudits involved in the TACCs of these sub-instances interfere with other sub-instances, and so we can consider them independently. Let $\{S_j^{(1,2)}\}$ denote the set of sub-instances of the first and second type, and $\{S_j^{(3)}\}$ the set of sub-instances of the third type. Here, each S_j is itself a set of clauses that define the sub-instance.

To determine whether the instance is a yes-instance or no-instance, we present the following claim:

Claim 3.2. *If the quantum sub-instances $\{S_j\}$ are satisfiable, the* candidate state

$$|\psi_{cand}\rangle = \bigotimes_j |\psi_{phist}\rangle_{S_j^{(1,2)}} \bigotimes_k \left(|\psi_{phist}\rangle \otimes |?\rangle_{l_u} \otimes \underbrace{|r\ldots r\rangle}_{L-T} \right)_{S_k^{(3)}}, \tag{26}$$

is always a satisfying state. Here, $|\psi_{phist}\rangle_{S_j}$ is the privileged history state defined in Eq. (25) where the circuit corresponds to the TACC associated with sub-instance S_j and l_u is the first undefined logical qudit of the TACC.

Proof. Assume that the quantum sub-instances $\{S_j\}$ are satisfiable. Consequently, by Proposition 3.3, the sub-instances with simultaneous propagation clauses have a single computational history. Then, because the TACC encompasses the whole ACC in the first and second type of sub-instances, they must be uniquely satisfied by a history state. These are the $|\psi_{phist}\rangle_{S_j^{(1,2)}}$ terms. This is not the case for the third type of sub-instance as it may be satisfied either by a truncated history state that leverages the undefined Π_{prop} clause or by a history state that encompasses the whole ACC. However, it is not difficult to see that if the latter history state is a satisfying state, then so must the truncated state.[23] These are the $(|\psi_{phist}\rangle \otimes |?\rangle_{l_u} \otimes |r\ldots r\rangle)_{S_k^{(3)}}$ terms above. □

The claim allows us to show that if the quantum algorithm can determine or verify that this candidate state is not a satisfying state, we can conclude that the instance must be unsatisfiable as no other state can be a

[23] The reverse implication is not necessarily true because even though the truncated ACC may be satisfiable, considering a longer chain with more constraints can only make the sub-instance harder and possibly unsatisfiable.

satisfying state.

The quantum algorithm presented above makes use of this claim and determines the satisfiability of the instance by indirectly measuring the violation of constraints incurred by the state $|\psi_{cand}\rangle$. In practice however, the algorithm only measures the constraints related to the TACC of the sub-instances, and even then it does not measure the Π_{init} or lone Π_{prop} projectors. This is because, first, the satisfying state of the clauses beyond the TACC is trivial, and second, because the privileged history states clearly satisfy the Π_{init} and lone Π_{prop} projectors. This is why upon receiving the TACC of a sub-instance S_j, the quantum algorithm only measures the eigenvalue of the simultaneous Π_{prop} projectors (if any) and the Π_{out} projectors within (if any) on the state $|\psi_{phist}\rangle_{S_j}$. Let $K_j \subset S_j$ and $M_j \subset S_j$ denote these two sets of projectors, respectively. Then, for a projector $\Pi_i \in K_j \cup M_j$, the probability that the algorithm determines the constraint is not satisfied (observes outcome "1"; equivalent to measuring eigenvalue 1) and rejects the instance is given by

$$p_{i,j} = \langle\psi_{phist}|\,\Pi_i\,|\psi_{phist}\rangle_{S_j}\,. \tag{27}$$

Then, the algorithm's acceptance probability AP, the probability that none of the measured ancilla qubits yield outcome "1", can be bounded as

$$1 - \sum_{i,j} p_{i,j} \leq AP \leq 1 - \max_{i,j} p_{i,j}. \tag{28}$$

Completeness

If the input instance x is a yes-instance, then all of the quantum sub-instances $\{S_j\}$ can be satisfied. For each S_j, the quantum algorithm measures the eigenvalues of projectors K_j and M_j on the state $|\psi_{phist}\rangle_{S_j}$. Since these privileged history states satisfy the sub-instances, it follows that $p_{i,j} = 0$ for all i and j. Then, from Eq. (28), we can conclude that $AP = 1$.

The state of all qubits that satisfies all clauses of the instance can be written as

$$|\psi_*\rangle = |\psi_{else}\rangle \otimes |\psi_{cand}\rangle\,, \tag{29}$$

where $|\psi_{else}\rangle$ is the joint satisfying state of the sub-instances that are trivially satisfiable. The algorithm does not explicitly compute $|\psi_{else}\rangle$, but determines that such a state exists, which is sufficient. In other words, $|\psi_*\rangle = |\psi_{sat}\rangle$.

Soundness

As mentioned above, the quantum algorithm can conclude that the instance is unsatisfiable if the state $|\psi_{cand}\rangle$ receives a penalty greater than $1/poly(n)$. To prove the soundness condition, it is necessary that the algorithm detects this case with probability $\geq 2/3$ (the probability that the circuit accepts the instance is $\leq 1/3$). We now show that this is the case.

The key idea is to leverage the problem's promise. Recall that for no-instances, we are promised that for any n-qudit state $|\psi\rangle$, the equation $\sum_{\Pi_i \in x} \langle\psi|\,\Pi_i\,|\psi\rangle \geq 1/poly(n)$ must hold. It then follows that for the state $|\psi_*\rangle$ which satisfies the instance in the yes-case, we now have

$$\begin{aligned}
\sum_{\Pi_i \in x} \langle\psi_*|\,\Pi_i\,|\psi_*\rangle &= \sum_{\Pi_i \in \{S_j\}} \langle\psi_{cand}|\,\Pi_i\,|\psi_{cand}\rangle \\
&= \sum_j \sum_{\Pi_i \in S_j} \langle\psi_{phist}|\,\Pi_i\,|\psi_{phist}\rangle_{S_j} \\
&= \sum_j \sum_{\Pi_i \in K_j \cup M_j} \langle\psi_{phist}|\,\Pi_i\,|\psi_{phist}\rangle_{S_j} \\
&\geq 1/poly(n),
\end{aligned} \tag{30}$$

where in the first line we made use of the fact that the projectors of the trivially decidable instances were all determined to be satisfiable by the classical algorithm. In the second line we simply expanded the expression and grouped some terms together, and in the third line we noted that $|\psi_{phist}\rangle_{S_j}$ always satisfies the Π_{init} and lone Π_{prop} projectors within. Then, the maximum violation of a projector in the instance can be bounded as

$$\max_{i,j} p_{i,j} \geq \frac{\sum_j \sum_{\Pi_i \in K_j \cup M_j} \langle \psi_{phist} | \Pi_i | \psi_{phist} \rangle_{S_j}}{\text{total \# of projectors}} \geq \frac{1/poly(n)}{\text{total \# of projectors}} \geq 1/poly(n), \tag{31}$$

where the second inequality follows from Eq. (30), and last inequality follows from the fact that there are at most a polynomial number of projectors in an instance. Combining this with Eq. (28), we get that $AP \leq 1 - 1/poly(n)$. Finally, as explained below Definition 2.5, since the gap between the acceptance probabilities of yes- and no-instances is given by an inverse polynomial, it is possible to reduce this acceptance probability below 1/3 and satisfy the soundness condition of BQP_1.

3.5 Hardness

Here, we show that the problem Linear-Clock-Ternary-QSAT is BQP_1-hard, thus completing the proof of Theorem 1. In other words, we show that we can encode any instance x of a BQP_1 promise problem $A = (A_{yes}, A_{no})$, into an instance x' of LCT-QSAT, such that any yes-instance (no-instance) of A also becomes a yes-instance (no-instance) of our problem.

Let $U_x = U_L \ldots U_1$ with $U_i \in \mathcal{G}$ and $L = poly(n)$ be the BQP_1 circuit where given an instance x of problem A, decides whether $x \in A_{yes}$ or $x \in A_{no}$. The input to the circuit is the q-qubit ancilla register $|0\rangle^{\otimes q}$, where $q = poly(n)$. Let ans be the qubit that when measured provides this decision. The reduction $x \mapsto x'$ requires $q + (L+1) + 2$ qudits, each of dimension 17, and is the following:

1. Choose q qudits to serve as the logical qudits of the computation, $L+1$ qudits as the clock qudits, and 2 qudits as the endpoint qudits. Define an ordering for each type of qudit.
2. For all $i \in [q]$, create a Π_{init} clause acting on c_0, e_0 and l_i.
3. For all $j \in [L]$, create a Π_{prop} clause with associated unitary U_{j+1}, which acts on clock qudits c_j and c_{j+1} (with c_j being the predecessor of c_{j+1}) and the two logical qudits that the unitary originally acts on.
4. Create one Π_{out} clause acting on c_L, e_1 and l_{ans}.

The resulting instance x' consists of a one-dimensional clock chain of length L ($L+1$ clock qudits) with a unique direction, no simultaneous propagation clauses, and fully initialized logical qudits (see Fig. 3a).

To prove the correctness of the reduction, let us first consider the case where $x \in A_{yes}$. Here, x' is satisfied by the history state

$$|\psi_{hist}\rangle = \frac{1}{\sqrt{L+1}} \sum_{t=0}^{L} U_t \ldots U_0 |0\rangle^{\otimes q} \otimes |\underbrace{d \ldots d}_{t} a_t \underbrace{r \ldots r}_{L-t}\rangle , \tag{32}$$

where $U_0 = I$.[24] As mentioned previously, this history state immediately satisfies the Π_{init} and Π_{prop} clauses of the instance, while the satisfiability of the Π_{out} clause depends on whether the evaluation of U on the initial state results in qudit l_{ans} being in the state $|1\rangle$. This is true since, by assumption, U accepts x with certainty. In other words, in the original instance, ans always produces outcome "1", implying that in our instance l_{ans} always results in the state $|1\rangle$.

When $x \in A_{no}$, we have to demonstrate that any state of the qudits in the instance incurs a penalty greater than an inverse polynomial. Unfortunately, this is more complex that in the previous case, and so it requires a more in-depth analysis.

3.5.1 No small eigenvalues for no-instances

In this section it will be more convenient to think of the O_{init}, O_{prop}, O_{out} operators as positive semi-definite operators (so as defined in Eqs. (12)–(14)) instead of projectors. This does not change in any way the task we have to prove. In fact, this makes the problem "harder" as the eigenvalues of excited states may now be

[24] More explicitly, the satisfying state is given by $|\psi_{hist}\rangle \otimes |\psi_{aux}\rangle$ where the second state accounts for the Bell pairs formed in the auxiliary subspaces of qudits.

closer to 0 instead of all of them being +1-eigenvalues.

As in Section 2.3.2, it is useful to group the similar types of operators in the LCT-QSAT instance in terms of Hamiltonians. The instance can then be expressed as

$$H = H_{roles} + H_{aux} + H_{clock} + H_{prop} + H_{init} + H_{out}, \tag{33}$$

where H_{roles} collects all of the Π_E, Π_C and Π_L terms, H_{prop} the $\Pi_{work,U}(\Pi_D \otimes \Pi_D \otimes I^{\otimes 2})_{j,j+1}$ terms (L of them), H_{init} the $(I - |0\rangle\langle 0|)_{l_i} \otimes |a\rangle\langle a|_{c_0}$ terms (q of them), and H_{out} the single $|0\rangle\langle 0|_{l_{ans}} \otimes |a\rangle\langle a|_{c_L}$ term. Additionally,

$$H_{aux} := (I - |\Phi^+\rangle\langle\Phi^+|_{EC,CA})_{e_0,c_0} + (I - |\Phi^+\rangle\langle\Phi^+|_{CB,EC})_{c_L,e_1} + \sum_{j=0}^{L}(I - |\Phi^+\rangle\langle\Phi^+|_{CA,CB})_{c_j,c_{j+1}} \tag{34}$$

collects the terms that require the auxiliary subspaces to form $|\Phi^+\rangle$ Bell pairs, and

$$\begin{aligned} H_{clock} := &\sum_{j=0}^{L}\left[\Pi_D \otimes \Pi_D \otimes \Pi_{clock,D} + (I^{\otimes 2} \otimes \Pi_{clock,?})(I - \Pi_D \otimes \Pi_D \otimes I^{\otimes 2})\right]_{i,i+1} \\ &+ (\underbrace{\Pi_{start} + \ldots + \Pi_{start}}_{n})_0 + (\Pi_{stop})_L, \end{aligned} \tag{35}$$

collects the L terms that constrain the states of the clock based on the state of the logical qudits, the Π_{start} projectors, and the single Π_{stop} projector. Since H_{clock} only acts on the clock qudits of the instance, we omit the c in the indices to avoid clutter. The Hamiltonian of Eq. (33) acts on the Hilbert space $\mathcal{H} = (\mathbb{C}^{17})^{\otimes q+L+3}$.

To show that the Hamiltonian of Eq. (33) has no small eigenvalues, we will partition the Hilbert space into "good" and "bad" subspaces and demonstrate that in either subspace, the smallest non-zero eigenvalue is always greater than an inverse polynomial. We will begin by considering $\mathcal{S}_{roles}$ and $\mathcal{S}_{roles}^{\perp}$ which represent the null space of H_{roles} and its orthogonal subspace, respectively. Clearly, $\mathcal{S}_{roles} \subset \mathcal{H}$, $\mathcal{S}_{roles}^{\perp} \subset \mathcal{H}$, and $\mathcal{S}_{roles} \oplus \mathcal{S}_{roles}^{\perp} = \mathcal{H}$. We will show that states in the orthogonal subspace (where qudits are not labeled properly) receive a constant "energy" penalty, whereas for the actual null space, we will apply the same logic and partition it into two subspaces, S_{aux} and $S_{aux}^{\perp}$, which we analyze in a similar way. We will repeat this process a third time (for H_{clock}), after which we will observe that when restricted to the "good" subspaces, the Hamiltonian of Eq. (33) is effectively identical to the one of Ref. [1] and simply rely on their result.[25] Observe that the order in which we consider these subspaces is relevant since, for example, we cannot conclude anything about the clock register before first ensuring that this takes place within the subspace where all arrows between clock qudits point in a unique direction. Fig. 8 summarizes these steps.

To begin, observe that $\mathcal{S}_{roles}$ is the product of subspaces where each subspace corresponds to the correct assignment of the particle. For the instance above, we expect

$$\mathcal{S}_{roles} = (\mathbb{C}^3)^{\otimes q} \otimes (\mathbb{C}^{12})^{\otimes L+1} \otimes (\mathbb{C}^2)^{\otimes 2}. \tag{36}$$

The Hamiltonian of Eq. (33) (and the projectors within) does not convert particles of one type to a different type, so both $\mathcal{S}_{roles}$ and $S_{roles}^{\perp}$ are invariant under its action. Any state in $S_{roles}^{\perp}$ has at least one particle that does not respect its assigned role and hence violates one of the Π_E, Π_L, or Π_C clauses within H_{roles} and receives a penalty ≥ 1.

Next, we consider H_{aux} restricted to the null space $\mathcal{S}_{roles}$, which we denote by $H_{aux}|_{\mathcal{S}_{roles}}$. The null space of this Hamiltonian is

$$\mathcal{S}_{aux} = (\mathbb{C}^3)^{\otimes q} \otimes (\mathbb{C}^3)^{\otimes L+1} \otimes |\psi_{aux}\rangle\,, \tag{37}$$

where $|\psi_{aux}\rangle$ is the state giving the linear structure to the clock qudits. To reiterate, here, the CB subspace of each clock qudit c_1 to c_{L-1} forms a $|\Phi^+\rangle$ Bell pair with the CA subspace of its successor. The clock qudits at the endpoints behave differently. The CA subspace of c_0 forms a Bell pair with the EC subspace of the first endpoint particle, while the CB subspace of c_L forms a Bell pair with the EC subspace of the second

[25] The clock encoding is different, but it does not affect the analysis whatsoever.

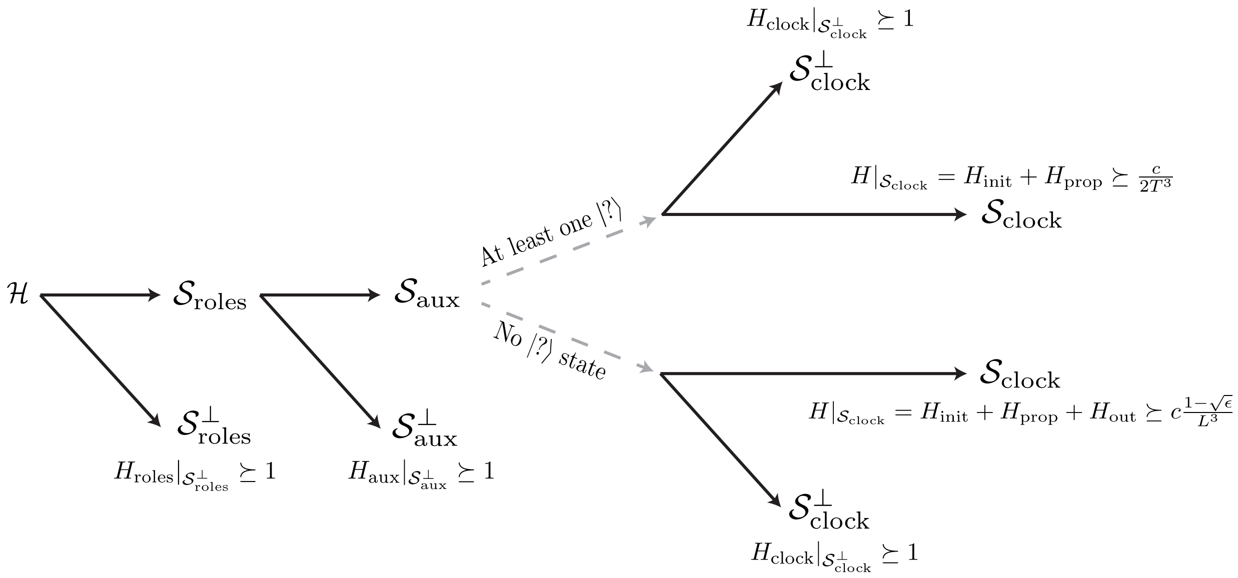

$H_{\text{clock}}|_{\mathcal{S}_{\text{clock}}^{\perp}} \succeq 1$
$\mathcal{S}_{\text{clock}}^{\perp}$
$H|_{\mathcal{S}_{\text{clock}}} = H_{\text{init}} + H_{\text{prop}} \succeq \frac{c}{2T^3}$
$\mathcal{S}_{\text{clock}}$
At least one $|?\rangle$
$\mathcal{H}$
$\mathcal{S}_{\text{roles}}$
$\mathcal{S}_{\text{aux}}$
No $|?\rangle$ state
$\mathcal{S}_{\text{clock}}$
$H|_{\mathcal{S}_{\text{clock}}} = H_{\text{init}} + H_{\text{prop}} + H_{\text{out}} \succeq c\frac{1-\sqrt{\epsilon}}{L^3}$
$\mathcal{S}_{\text{roles}}^{\perp}$
$\mathcal{S}_{\text{aux}}^{\perp}$
$H_{\text{roles}}|_{\mathcal{S}_{\text{roles}}^{\perp}} \succeq 1$
$H_{\text{aux}}|_{\mathcal{S}_{\text{aux}}^{\perp}} \succeq 1$
$\mathcal{S}_{\text{clock}}^{\perp}$
$H_{\text{clock}}|_{\mathcal{S}_{\text{clock}}^{\perp}} \succeq 1$

Figure 8: Diagram summarizing the steps of the proof showing that the Hamiltonian of Eq. (33) has no small eigenvalues. Starting by considering the full Hilbert space $\mathcal{H}$, we partition the space into two (represented by the forking arrows): the null space corresponding to a term in the Hamiltonian, and the orthogonal subspace. Underneath each subspace, we write the smallest eigenvalue of the Hamiltonian term restricted to that subspace. The diagram shows that in all subspaces, there can be no state that receives a small energy penalty. Another important property is that all subspaces are invariant under the action of the Hamiltonian.

endpoint particle. This subspace and the orthogonal subspace are invariant under the action of H. Then, observe that since the four Bell states form a basis of the 2-qubit space, the orthogonal subspace is spanned by the tensor product of any combination of the other three states. It is then clear that any state in $\mathcal{S}_{aux}^{\perp}$ violates at least one of the projectors within H_{aux}, and thus receives a penalty ≥ 1.

We can now consider $H_{clock}|_{\mathcal{S}_{aux}}$. In its current form, this Hamiltonian is difficult to analyze since the clock qudits are intertwined with the logical qudits. Indeed, depending on the state of the logical qudits, the terms within might result in either $\Pi_{clock,D}$ or $\Pi_{clock,?}$, which enforce slightly different constraints. To simplify the analysis, we partition the space into two subspaces: one where there are no logical qudits in the undefined state, and the other subspace where at least one of them is in this state. Importantly, these subspaces are invariant under the action of H since there is no transformation that takes a logical qudit in the state $|?\rangle$ to the state $|0\rangle$ or $|1\rangle$, and vice versa. Consequently, any logical qudit in the state $|?\rangle$, remains in this state.

At least one undefined logical qudit

Let $O_{prop}^{(c_T,c_{T+1})}$ with $0 \leq T < L$ be the first O_{prop} clause of the instance that acts on an undefined logical qudit. Let us refer to this qudit as l_u with $0 \leq u < q$, and for simplicity, let us assume that this is the only undefined qudit of the instance as any additional such qudits can only increase the energy further. In this case, the clock Hamiltonian of Eq. (35) becomes

$$\begin{aligned} H_{clock}(T,u) = \underbrace{\Pi_{start} + \ldots + \Pi_{start}}_{q} + \Pi_{stop} + \sum_{i=0}^{T-1} (\Pi_{clock,D})_{i,i+1} + \ (\Pi_{clock,?})_{T,T+1} \\ + \sum_{j=T+1}^{L} \left[\Pi_D \otimes \Pi_D \otimes \Pi_{clock,D} + (I^{\otimes 2} \otimes \Pi_{clock,?})(I - \Pi_D \otimes \Pi_D \otimes I^{\otimes 2})\right]_{j,j+1}, \end{aligned} \tag{38}$$

where we have not assumed a specific form of the clock projectors that act on clock qudits $T+1$ to L. In fact, the exact form of these clauses is not of much relevance. Indeed, observe that as shown in Eq. (23), the clock qudit c_{T+1} can only be in the state $|r\rangle$ in order to satisfy $\Pi_{clock,?}$. Then, regardless of whether the next clause is $\Pi_{clock,D}$ or $\Pi_{clock,?}$, both clauses require that c_{T+2} must also be fixed to $|r\rangle$ in order not to incur a penalty. The same is true for all remaining clauses.

The null space and orthogonal subspace of this Hamiltonian are

$$\begin{aligned} \mathcal{S}_{clock}(T,u) &= \mathcal{D}_u \otimes \mathrm{span}\{|C_0\rangle, \ldots, |C_T\rangle\}, \\ \mathcal{S}^{\perp}_{clock}(T,u) &= \mathcal{D}_u \otimes \mathrm{span}(\mathcal{I} \cup \{|C_{T+1}\rangle, \ldots, |C_L\rangle\}), \end{aligned} \tag{39}$$

where $\mathcal{I}$ represents the set of illegal clock states and $\mathcal{D}_u := (\mathbb{C}^2)^{\otimes u} \otimes |?\rangle_u \otimes (\mathbb{C}^2)^{\otimes q-u-1}$ is the space of logical qudits considering that the undefined logical qudit is at position u. Both null spaces are invariant under the action of H as there are no terms that transform legal states to illegal ones, or terms that force the clock to progress past $|C_T\rangle$ (as shown below in Eq. (40), there is no $(T+1)$th Π_{work} clause within H_{prop}). We can now conclude that any state in $\mathcal{S}^{\perp}_{clock}$ violates at least one of the $\Pi_{clock,D}$ or $\Pi_{clock,?}$ terms and thus receives a penalty ≥ 1.

Now, we can consider H restricted to the null space of $S_{clock}(T,u)$. Within this subspace, H simplifies to

$$H|_{\mathcal{S}_{clock}(T,u)} = H_{init} + H_{prop}(T,u); \qquad H_{prop}(T,u) := \sum_{i=0}^{T-1} (\Pi_{work})_{i,i+1}, \tag{40}$$

where we again made use of the fact that H_{prop} also depends on the undefined logical qudit, and the fact that H_{out} is projected out regardless of the position of the first undefined logical qudit. Here, $H_{prop}(T,u)$ is the Hamiltonian often seen in literature [1, 3, 41], which demands the application of the unitary circuit $U_T \ldots U_1$ on a data register.

We can now bound the smallest eigenvalue of $H_{init} + H_{prop}(T,u)$ by using Kitaev's Geometric Lemma stated in Lemma 2.1. As done in Ref. [1], the smallest non-zero eigenvalues of these operators are found by performing a change of basis to the "rotating" frame. Under the transformation, H_{init} remains unchanged, while H_{prop} takes the shape of a $(T+1) \times (T+1)$ tridiagonal matrix. The smallest non-zero eigenvalues are $\gamma(H_{init}) = 1$ and $\gamma(H_{prop}) = c/T^2$, with

$$\mathcal{S}_{init} = \mathcal{D}_u \otimes \mathrm{span}\{|C_1\rangle, \ldots, |C_T\rangle\} \qquad \text{and} \qquad \mathcal{S}_{prop} = \mathcal{D}_u \otimes \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |C_t\rangle \tag{41}$$

as their respective null spaces.[26] Given the simple form of these null spaces, it is straightforward to show that in the Geometric Lemma of Eq. (9),

$$\alpha = \max_{|\eta\rangle \in \mathcal{S}_{prop}} \langle\eta| \Pi_{\mathcal{S}_{init}} |\eta\rangle = \frac{T}{T+1} = 1 - \frac{1}{T+1}, \tag{42}$$

and so

$$H|_{\mathcal{S}_{clock}(T,u)} \succeq \min\{\gamma(H_{init}), \gamma(H_{prop})\} \frac{(1-\alpha)}{2} \succeq \frac{c}{2T^3}. \tag{43}$$

This concludes that smallest eigenvalue of H scales as $\Omega(1/T^3)$.

No undefined logical qudit

In this case, the clock Hamiltonian becomes

$$H_{clock} = \underbrace{\Pi_{start} + \ldots + \Pi_{start}}_{q} + \Pi_{stop} + \sum_{i=0}^{L} (\Pi_{clock,D})_{i,i+1}, \tag{44}$$

[26] Note that the term $|0\rangle^{\otimes q} \otimes |C_0\rangle\langle C_0|$, which is normally in the null space of H_{init}, is no longer in this space since one of the logical qudits is assumed to be in the state $|?\rangle$.

with null space and orthogonal subspace given by

$$\mathcal{S}_{clock} = (\mathbb{C}^2)^{\otimes q} \otimes \text{span}\{|C_0\rangle, \dots, |C_L\rangle\} \qquad \text{and} \qquad \mathcal{S}_{clock}^{\perp} = (\mathbb{C}^2)^{\otimes q} \otimes \text{span}(\mathcal{I}). \tag{45}$$

As mentioned previously, these subspaces are invariant under H as there is no term that transforms a legal clock state into an illegal state, or vice versa. Any state in $\mathcal{S}_{clock}^{\perp}$ violates at least one of the clauses within H_{clock} and thus receives a penalty ≥ 1.

By restricting the Hamiltonian of Eq. (33) to $\mathcal{S}_{clock}$, we obtain that $H|_{\mathcal{S}_{clock}} = H_{init} + H_{prop} + H_{out}$ where $H_{prop} = \sum_{i=0}^{L} (\Pi_{work})_{i,i+1}$. Letting $H_1 := H_{init} + H_{prop}$ and $H_2 := H_{prop}$ and again moving to the rotating frame, we can write the null spaces of these terms as

$$\mathcal{S}_1 = \Big(|0\rangle^{\otimes q} \otimes |C_0\rangle \Big) \oplus \Big((\mathbb{C}^2)^{\otimes q} \otimes \text{span}\{|C_1\rangle, \dots |C_{L-1}\rangle\} \Big) \oplus \Big(U^{\dagger}(|1\rangle_{ans} \otimes (\mathbb{C}^2)^{\otimes q-1}) \otimes |C_L\rangle \Big) \tag{46}$$

and

$$\mathcal{S}_2 = (\mathbb{C}^2)^{\otimes q} \otimes \frac{1}{\sqrt{L+1}} \sum_{t=0}^{L} |C_t\rangle \,. \tag{47}$$

Except for the different clock encoding and the absence of the subspace for witness qudits, these Hamiltonians and their corresponding null spaces are similar to those of Ref. [1]. By following the same steps as in that proof (or alternatively pages 236-239 in Ref. [42]), we can obtain that the lowest energy penalty a state may receive is $\geq c(1-\sqrt{\epsilon})L^{-3}$ where c is a positive constant and $\epsilon \leq 1/3$ is the probability that the verifier of problem A wrongfully accepts.

The soundness analysis concludes that in the case where there is an undefined logical qudit, $H \succeq c(2T^3)^{-1}$, and $H \succeq c(1-\sqrt{\epsilon})L^{-3}$ when there are none. Since $T \leq L = poly(n)$, we have that $\sum_i \langle\psi| \Pi_i |\psi\rangle \geq 1/poly(n)$ for any state $|\psi\rangle$ on $q + L + 3$ 17-dimensional qudits, and so x' is a no-instance of LCT-QSAT.

4 Reducing the Qudit Dimensionality

In this section, we remove the use of monogamy of entanglement and show that the resulting problem, now referred to as Semilinear-Clock-Ternary-QSAT, is BQP_1-complete and uses lower-dimensional qudits.

We demonstrate that even without the monogamy condition, we can obtain similar results about the satisfiability of instances. We show that the rules of propagation, the choice of clock encoding, and the requirement to keep a consistent state of the clock register at all times suffice to demonstrate that TACCs with forks or Π_{prop} clauses pointing in different directions are unsatisfiable. Additionally, we also show that Π_{init} and Π_{out} still serve as endpoints, albeit as a weaker version, and thus there is no need for endpoint qudits. Together, these results show that while the use of monogamy in the construction does facilitate some proofs, it is not crucial for the construction. Removing this condition and the use of endpoint qudits allows us to reduce the local dimension from 17 down to 6. The main difference in this construction is that sub-instances with multiple ACCs may now have a satisfying state. These sub-instances take the form of one-dimensional TACCs joined together by complex webs of connections between clock qudits, thus explaining the term *semilinear* in the name of the problem (see Fig. 9). We show that if these sub-instances are potentially satisfiable, they can be further separated into smaller instances composed only of the TACCs, and the clauses connecting them can be satisfied trivially.

The problem is defined similarly to Definition 3.1, except that the terms Π_E and $I - |\Phi^+\rangle\langle\Phi^+|$ are omitted in the definitions of Π_{init}, Π_{out}, and $\Pi_{prop,U}$.

Definition 4.1 (Semilinear-Clock-Ternary-QSAT)**.** *The problem* Semilinear-Clock-Ternary-QSAT *is a quantum constraint satisfaction problem defined on the* 6*-dimensional Hilbert space*

$$\mathcal{H} = \text{span}\{|0\rangle, |1\rangle, |?\rangle\} \oplus \text{span}\{|r\rangle, |a\rangle, |d\rangle\}, \tag{48}$$

consisting of a logical and a clock subspace. The problem consists of 5 *types of clauses: the initialization clause* Π_{init}*, the propagation clauses* $\Pi_{prop,U}$ *(one for every* $U \in \mathcal{G}$*), and a termination clause* Π_{out}*, each acting on at most four* 6*-dimensional qudits. These are defined as*

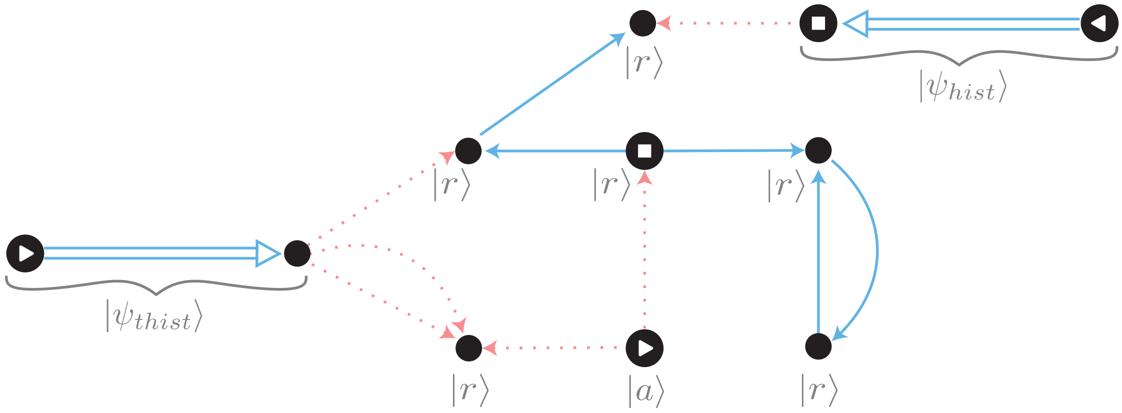


Figure 9: A clock component of a SLCT-QSAT sub-instance that may be satisfiable. The long double-lined arrows represent non-zero length TACCs where all Π_{prop} clauses within are well-defined and have the indicated direction. They may still contain simultaneous propagation clauses. The clock component hence has two TACCs plus an additional active dot at the bottom of the image (the qudit with the "start" icon present in two undefined Π_{prop} clauses). The TACC on the left of the image is truncated and so the clauses within this chain are trivially satisfiable by a truncated history state $|\psi_{thist}\rangle$, while the TACC on the right requires a quantum computer to determine whether a history state $|\psi_{hist}\rangle$ satisfies the Π_{out} clause. The clauses involving the active dot can be satisfied by fixing this clock qudit to $|a\rangle$, the adjacent clock qudits to $|r\rangle$, and the logical qudits present in the Π_{init} clauses to $|0\rangle$. The remaining clauses can be satisfied by setting the remaining clock qudits to $|r\rangle$, leaving their logical qudits unconstrained. To conclude that the instance is satisfiable, it is also necessary to verify that the logical qudits involved in the three TACCs do not conflict with each other.

$$O_{init} := \Pi_{L,1} + \Pi_{C,2} + \Pi_{start,2} + (I - |0\rangle\langle 0|)_1 \otimes |a\rangle\langle a|_2 \,, \tag{49}$$

$$\begin{aligned} O_{prop,U} := \; & \Pi_{L_1} + \Pi_{L,2} + \Pi_{C,3} + \Pi_{C,4} \\ & + (\Pi_{work} + I^{\otimes 2} \otimes \Pi_{clock,D})(\Pi_D \otimes \Pi_D \otimes I^{\otimes 2}) \\ & + (I^{\otimes 2} \otimes \Pi_{clock,?})(I - \Pi_D \otimes \Pi_D \otimes I^{\otimes 2}), \end{aligned} \tag{50}$$

and

$$O_{out} := \Pi_{L,1} + \Pi_{C,2} + \Pi_{stop,2} + |0\rangle\langle 0|_1 \otimes |a\rangle\langle a|_2 \,, \tag{51}$$

where again $\Pi_{init}, \Pi_{prop,U}$, and Π_{out} are the projectors with the same nullspace as the one defined by the positive semi-definite operators O_{init}, $O_{prop,U}$, and O_{out}. The projectors found in these clauses and the rest of the definition is the same as shown in Definition 3.1. Evidently, Π_{init}, Π_{out} and $\Pi_{prop,U}$ are still at most 4-local clauses.

The null spaces of the operators in the definition above are similar to those of Eqs. (20)–(22), with the exception that there is no longer a $|\psi_{aux}\rangle$ to represent the Bell pairs formed in the auxiliary subspaces.

4.1 Instance satisfiability

Since the definition of the clauses is similar to those of Definition 3.1, and the locality is also identical, there is no new phenomena or new structures of instances to consider. Therefore, let us begin with the analysis of instances. The proofs of the lemmas and propositions shown here can be found in Appendix B.

The first lemma dictating the satisfiability of the instance is the same as before, except now there are no clock endpoints.

Lemma 4.1 (Single-type qudits)**.** *If a 6-dimensional qudit in the instance is acted on by both a Π_C and Π_L clause, the instance is unsatisfiable.*

As before, we now consider the clock components of the sub-instance. This time however, the clock components are simply the connected components defined solely by clock qudits, and not both clock qudits and endpoint qudits.

4.1.1 A clock component and its logical qudits

In Section 3.3, the need to adhere to the principle of monogamy of entanglement provided the first criteria for determining the satisfiability of instances and established the unsatisfiability of a large number of instances. Indeed, Lemmas 3.2–3.5 determined that clock components with forks, badly oriented Π_{prop} clauses, or misplaced Π_{init} and Π_{prop} were unsatisfiable. In this construction, we can no longer make such statements. However, we find that these components can still be decided, and are in fact satisfiable provided the instance does not have an active clock chain.

Lemma 4.2 (Lack of an ACC)**.** *If a clock component does not have at least one Π_{init} and one Π_{out} clause, the sub-instance is satisfiable.*

Essentially, because there are no constraints demanding that monogamy of entanglement is preserved, we have skipped Lemmas 3.2–3.5 and jumped directly to the analogue of Lemma 3.6. We remark that although the statement here is the same as in Lemma 3.6, this lemma decides a much larger class of sub-instances.

Sub-instances that are not decided by the previous lemma must have at least one Π_{init} and one Π_{out} clause, forming an ACC of some length L with $0 \leq L < \mathcal{O}(n)$. In the previous construction, the monogamy of entanglement condition and endpoint qudits straightforwardly dictated that potentially satisfiable sub-instances must have exactly one ACC. Here, we do not have such a luxury. Instead, we have to evaluate the satisfiability of more complex clock components with potentially multiple ACCs. Ultimately, we will see that the satisfiability of the instance reduces to determining the satisfiability of TACCs.

At the moment, all we can gather is that a satisfying state of the clauses of the chain requires one of the clock qudits to be in the state $|a\rangle$. However, it is unknown whether this is enough to trigger the propagation of computation as it is also unknown which clauses can be well-defined or undefined. To determine this, we show a more powerful generalization of Proposition 3.1.

Proposition 4.1 (Initialization of logical qudits)**.** *Let $c_0, \ldots c_L$ with $L \geq 0$ be a chain of clock qudits where c_0 is a qudit present in a Π_{init} clause, and c_L is a qudit present in a Π_{out} clause. Suppose the Π_{prop} clauses of the chain point in arbitrary directions and suppose that there might be other Π_{init} clauses acting on other clock qudits within the chain (i.e. $c_0, \ldots, c_L$ need not be an ACC). A state where c_0 is not $|a\rangle$ in any basis state of the superposition is not a satisfying state.*

This proposition allows us to show that in any possible satisfying state, all of the clock qudits of the component acted on by Π_{init} must be allowed to be $|a\rangle$ at some point in time. This is because for any such clock qudit (c_0), there exists a path that connects it to another clock qudit with a Π_{out} clause. And while this path may contain other clock qudits present in Π_{init} clauses, the proposition assures us that that c_0 must be active at some point in time. Consequently, since this proposition applies to all clock components of the instance, we can identify for the whole instance which Π_{prop} clauses may be well-defined or undefined. In the upcoming algorithm, this reflected in step (1.2).

Then, we can consider the following two lemmas:

Lemma 4.3 (Π_{prop} clauses acting on c_0 point outward)**.** *Consider a clock qudit present in a Π_{init} clause of a clock component. If this qudit is also present in a Π_{prop} clause (either well-defined or undefined) that points towards it, the instance is unsatisfiable.*

Lemma 4.4 (One well-defined Π_{prop} clause maximum)**.** *Consider a clock qudit present in a Π_{init} clause and a well-defined Π_{prop} clause. If this qudit is also present in another Π_{prop} clause (either well-defined or undefined) that connects to a new clock qudit, the instance is unsatisfiable.*

These lemmas show that if the sub-instance has a potential satisfying state, the Π_{prop} clauses that act on this clock qudit must fall into one of the following two scenarios: either all of the Π_{prop} clauses are undefined and point away from this qudit, or there is a single well-defined Π_{prop} clause that points away from it. These

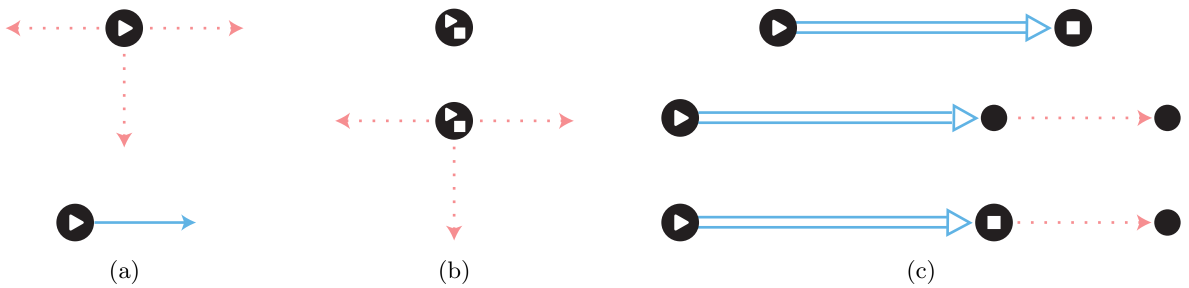


Figure 10: (a) Examples of the only types of Π_{prop} clauses acting on a clock qudit with a Π_{init} clause that may be satisfiable. We assume that the clock qudit is not present in a Π_{out} clause. Top: one or many undefined Π_{prop} clauses that point away from the qudit. Bottom: at most one well-defined Π_{prop} clause that points away from the qudit. (b) Example of the only type of active dots that may be satisfiable. Top: an active dot with no Π_{prop} clauses. Bottom: an active dot with one or many undefined Π_{prop} clauses that point away from the qudit. (c) Examples of the only types of non-zero length TACCs that may be satisfiable. Top: a TACC that encompasses the whole ACC with no clauses past the Π_{out} clause. Middle: a truncated TACC. Bottom: a TACC that encompasses the whole ACC but has clauses past the Π_{out} clause. For the last two cases, the qudit outside the TACC is allowed to be present in Π_{out} clauses.

two cases are illustrated in Fig. 10a. Now observe that since the clock component is fully connected, at least one of these paths leads to a clock qudit with a Π_{out} clause. Therefore, in the first scenario, the clock qudit forms an active dot. In the second scenario, the qudit forms part of a single chain of well-defined clauses that either terminates at some time T due to an undefined Π_{prop} clause, or reaches the clock qudit with the Π_{out} clause, in which case $T = L$. Technically, this is not a TACC as some clock qudits within the chain might contain other Π_{init} clauses; however, as will be discussed in Lemma 4.7, this event makes the instance unsatisfiable. For now, let us assume that the chain is an actual TACC and come back to this issue at a later time.

The case where the clock qudit forms an active dot can be satisfied easily. We simply set its state to $|a\rangle$ and initialize all logical qudits present in its Π_{prop} clause to $|0\rangle$. Arguing about the satisfiability of the (more general) TACCs requires more work.

Satisfiability of a TACC and its neighboring clauses

Proposition 4.1 shows that for active clock chains (and their generalization that includes Π_{init} clauses in between the chain) c_0 must be $|a\rangle$ at some point in time. Since this qudit is always part of a TACC, we have that the TACC must contain at least one active spot. We now show that these chains must contain exactly one such state at any given time, and use this to show that potentially satisfiable TACCs must be one-dimensional chains with a unique direction.

Proposition 4.2 (Unique active state in a TACC)**.** *Let $c_0, \ldots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. A state in which at any given time there are two clock qudits c_a and c_b with $0 \leq a < b \leq T$ in the state $|a\rangle$ is not a satisfying state.*

We now demonstrate that potentially satisfiable TACCs must be one-dimensional chains where all Π_{prop} clauses of the chain point away from the clock qudit with the Π_{init} clause.

Lemma 4.5 (No forks in a TACC)**.** *Let $c_0, \ldots, c_T$ with $1 < T \leq L$ be the clock qudits of a TACC with length at least two, and let c_j be a clock qudit that is not at the ends of the chain, i.e. $0 < j < T$. If c_j is connected to more than two other clock qudits, the instance is unsatisfiable.*

Lemma 4.6 (Unique direction of a TACC)**.** *Let $c_0, \ldots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. If there exists a Π_{prop} clause within the TACC that points towards c_0, the instance is unsatisfiable.*

Note that unlike in the previous construction where these two lemmas (Lemmas 3.2 and 3.3) applied to the whole active clock chain, here they only apply within the TACC.

As Lemma 4.5 only referred to the clock qudits in the interior of the chain, it remains to answer whether the clock qudits at the ends of the TACC may take part in other Π_{prop} clauses. In the previous construction, this question was trivial since the auxiliary subspaces and the principle of monogamy of entanglement ensured that (for potentially satisfiable instances) the clock qudit with the Π_{init} clauses and the one with the Π_{out} clauses marked the beginning and end of the chain, respectively. The only case where there could be clauses beyond the TACC, was when the it was a strict subset of the ACC, i.e. when an undefined clause truncated the propagation within an ACC. However, even in this case, the clock qudit with the Π_{out} clauses marked the end of the component. We now show that we can achieve a similar behavior in this construction.

Lemma 4.7 (Neighboring clauses of a non-zero length TACC)**.** *Let $c_0, \ldots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. Π_{prop} clauses that lie outside the TACC with the following properties result in an unsatisfiable instance:*

1. *If they are well-defined or undefined and include c_0.*

2. *If they are well-defined and include c_T.*

3. *If they are undefined, include c_T, and point towards it.*

In this lemma, it is important that T should be able to equal L, since without endpoint qudits to guarantee that the chain ends at the clock qudit with the Π_{out} clause, it is possible that the chain extends beyond this point. This lemma does not reject the only other type of Π_{prop} clause available: that where the clause is undefined, includes c_T, and points away from it. This is because these sub-instances may be satisfiable. Even in the previous construction this was a possibility if the TACC was a strict subset of the ACC ($T < L$), as seen in Lemma 3.7. The difference now is that the clock chain may continue past the clock qudit with the Π_{out} clause. Moreover, since the clauses beyond this TACC may not be part of other TACCs, lemmas Lemmas 4.5 and 4.6 may not apply and the clauses could form forks or cycles, with clauses pointing in any direction. This behavior can be seen in Fig. 9.

Now, let us briefly discuss the behavior of active dots. Here, they may occur when the clock qudit is present in both Π_{init} and Π_{out} clauses, or the clock qudit is present in at least one Π_{init} clause and the ACC is immediately truncated by an undefined clause. Since these TACCs do not contain Π_{prop} clauses, all there is to do is to analyze the Π_{prop} clauses external to the active dot. The next lemma shows that the clock qudit of the TACC behaves like c_T in Lemma 4.7.

Lemma 4.8 (Neighboring clauses of an active dot)**.** *Let c_d be an active dot. Π_{prop} clauses that act on c_d with the following properties result in an unsatisfiable instance:*

1. *If they are well-defined or undefined, and point towards c_d.*

2. *If they are well-defined and point away from c_d.*

Thus, the only type of Π_{prop} clauses that can act on this qudit are those that are undefined and point away from the qudit. This is shown in Fig. 10b.

This lemma and Lemma 4.7 show that a TACC (regardless of whether it has zero length, is truncated, or encompasses the whole ACC) is allowed to continue only if the next Π_{prop} clause is undefined and points away from the last clock qudit of the TACC.

The observations that the clock component is able to continue past the clock qudit with the Π_{out} clause and the clauses outside TACCs may fork or have any direction make the analysis of the satisfiability of these clock components seem more daunting; however, as we will now show, the lemmas above suffice to determine the satisfiability of all instances.

Putting the pieces together

Now we can put the previous lemmas into the bigger picture and show how they can allow us to decide the satisfiability of an instance or reach the point where the quantum algorithm can take over.

To begin, Lemma 4.1 ensures that if the instance is potentially satisfiable, all qudits in the instance must

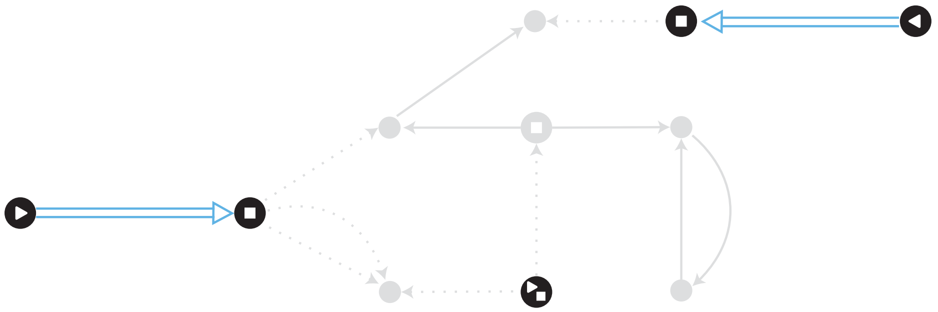


Figure 11: The clock component of Fig. 9 where the trivially satisfiable clauses are now in gray and the TACCs are highlighted. The TACC on the left and the TACC on the bottom receive an artificial Π_{stop} clause acting on their last clock qudit. These TACCs can now be treated as separate sub-instances.

serve a single role. Afterwards, we focus on the sub-instances, and use Lemma 4.2 to ignore those that do not have both Π_{init} and Π_{out} clauses. For each of the remaining sub-instances, Lemma 4.3 states that clock qudits present in Π_{init} clauses must have one of two possible structures: either the clock qudit is present only in undefined clauses that point away from it, or is part of well-defined Π_{prop} clauses that involve at most one other clock qudit (simultaneous propagation clauses are allowed). If the clock qudit has the first of these structures then this clock qudit forms an active dot. On the other hand, if the clock qudit has the second structure, then it must be part of a non-zero active clock chain. Lemmas 4.5 and 4.6 then dictate that this non-zero length TACC must be a linear chain and all Π_{prop} clauses in the chain must point away from the clock qudit with the Π_{init} clause. The TACC may either finish because of an undefined clause or end when it reaches the clock qudit with the Π_{out} clause. In either case, Lemma 4.7 shows that any adjacent clauses to the end of the TACC must be undefined and point away from the clock qudit that culminates the TACC. Unfortunately, we are unable to make any statement about the shape, direction, or definition of any Π_{prop} clauses beyond this one.

In summary, these lemmas show that potentially satisfiable sub-instances must have a clock component that is a collection of TACCs and active dots joined by a complex web of clock qudits consisting only of Π_{prop} and Π_{out} clauses (see Fig. 9). Crucially, as there are no Π_{init} clauses in this complex web of clauses to demand that there is an active clock qudit, the clauses within the web can be satisfied trivially by setting their clock qudits to $|r\rangle$. Moreover, since the clauses adjacent to the TACCs point away from them, this assignment does not conflict with the possible satisfying history states or truncated history states of the TACCs. In other words, the complex web of Π_{prop} and Π_{out} that link the TACCs can be thought of as being the joint successor of all TACCs.

As mentioned, since the clauses connecting the TACCs can be satisfied without constraining any of their logical qudits, we can effectively think of the TACCs of the sub-instance as smaller disconnected sub-instances. For TACCs that are truncated (including active dots), we can treat them as separate sub-instances by placing an artificial Π_{stop} clause on the last clock qudit of the TACC and ignore all clauses beyond. This does not alter the TACC since this is the exact same effect that an undefined clause pointing away from the qudit has on the clock component. This idea is illustrated in Fig. 11. We remark that the chains with artificial Π_{stop} clause at one end should not be confused with chains that end with Π_{out} clauses which constrain both clock and logical qudits. All of the sub-instances now have a simpler clock component where they begin with a clock qudit present in a Π_{init} clause, a series of well-defined Π_{prop} clauses forming a one-dimensional chain and pointing away from the clock qudit with the Π_{init} clause, and end with a clock qudit present in a Π_{stop} clause (some which may be artificially placed and others due to a Π_{out} clause).

4.1.2 Shared logical qudits and simultaneous propagation

The discussion about shared logical qudits and simultaneous propagation clauses is the same as in the previous construction. There, the auxiliary subspaces helped determine that the active clock chains should be, as the name implies, one-dimensional chains of qudits where all Π_{prop} clauses point in a unique direction. Previous to this, we argued that the relevant logical qudits should be those involved in clauses stemming from the TACC. We have obtained these same results in this construction above.

Similarly, the statements about simultaneous propagation clauses still hold, as they did not rely on auxiliary subspaces or endpoint qudits.

4.2 Algorithm and correctness

In this subsection we discuss the algorithm showing that SLCT-QSAT can be decided in polynomial-time with a quantum computer while demonstrating perfect completeness and bounded soundness, i.e. SLCT-QSAT is in BQP_1. This is the first part of the proof of Theorem 2.

4.2.1 Classical and quantum algorithm

The first two steps in the algorithm change slightly compared to the previous one, while the third step remains unchanged. The first step no longer mentions Π_E, and in the second step, only (2.5) remains. To be concrete, these are:

(1) *Verify that all qudits in the instance serve a single role.* For every qudit in the instance, check whether it is acted on by Π_C or Π_L clauses. If a qudit is acted on by both types of clauses, reject. Otherwise, label the qudit as either C or L depending on the type of clause acting on it. Correctness is given by Lemma 4.1.

(2) *Ignore trivial sub-instances.* For every sub-instance, check whether it contains both Π_{init} and Π_{out} clauses. If it does not, ignore all clauses of the sub-instance for the rest of the algorithm. Correctness is given by Lemma 4.2.

(3) *Identify the undefined Π_{prop} clauses.* For every logical qudit, check whether it is acted on by at least one Π_{init} clause. If it is not, mark all Π_{prop} clauses this qudit is part of as undefined.

Next, we add a new step to account for the lemmas regarding the satisfiability of clock components and TACCs of this construction. To preserve the numbering of the algorithm, we label this step as (CC), referring to the steps related to the analysis of clock components. The step is the following:

(CC) For every sub-instance, do the following:

(CC.1) Identify all undefined Π_{prop} clauses within the sub-instance and place an artificial Π_{stop} clause on the predecessor qudit.

(CC.2) *Reject if the clock qudits with Π_{init} clauses do not look like those of Fig. 10a.* For every clock qudit present in a Π_{init} clause, but not acted on by any Π_{stop} clause, perform the following checks:

(CC.2.1) If the qudit participates in a Π_{prop} clause that points towards it, reject. Correctness is given by Lemma 4.3.

(CC.2.2) If the qudit participates in two or more Π_{prop} clauses involving different clock qudits and one of these clauses is well-defined, reject. Correctness is given by Lemma 4.4

(CC.3) *Evaluate the segments of the chain that resemble the bottom drawing of Fig. 10a (these must be part of a non-zero length TACC). We reject the TACC if it does not look like those of Fig. 10c.* For each clock qudit present in a Π_{init} clause and a well-defined Π_{prop} clause, but not in any Π_{stop} clause, do the following:

(CC.3.1) Let c_j with $j = 1$ be the successor clock qudit.

(CC.3.2) If c_j is present in a Π_{stop} clause, perform the following checks:

- If there is a Π_{prop} clause (other than the one connecting it to c_{j-1}) that points towards it or is well-defined, reject. Correctness is given by Lemma 4.7.
- *Stop condition.* Stop the analysis of this sub-instance and continue with the next sub-instance.

(CC.3.3) *Reject if there's a fork.* If c_j is connected to more than three distinct clock qudits, reject. Correctness is given by Lemma 4.5.

(CC.3.4) If there's a Π_{prop} clause between c_j and c_{j+1} that points towards c_j, reject. Correctness is given by Lemma 4.6.

(CC.3.5) *The previous two steps guarantee that c_j has a single successor.* Replace $j \to j+1$ and go back to step (CC.3.2).

(CC.4) *Evaluate the active dots.* For every clock qudit present in both a Π_{init} and Π_{stop} clause, check whether it participates in a Π_{prop} clause that points towards it, or participates in a well-defined clause. If any of these are true, reject. Correctness is given by Lemma 4.8.

(CC.5) *Separate the TACCs found in the previous steps.* Ignore the clauses that do not form part of TACCs.

After this step, the algorithm has separated the TACCs of the sub-instances into new, smaller sub-instances, each with the same structure as in the previous construction. Consequently, the algorithm resumes with step (4) of the previous construction (see Section 3.4). While steps (4), (6), and (7) proceed in the same way as before, step (5) needs a small modification:

(5) *Ignore truncated active clock chains.* For every sub-instance, check whether it ends with an artificial Π_{stop} clause (*not* an actual Π_{out} clause). If it does, and there are is no simultaneous propagation before this clause, ignore the clauses of the sub-instance. Correctness is given by Lemma 3.7.

4.2.2 Runtime

Compared to the previous algorithm, the only significant difference in this algorithm that may impact the runtime is the addition of step (CC). However, this step is also efficient.

In detail, step (CC.1) can be easily seen to take $\mathcal{O}(poly(n))$ time. Step (CC.2) iterates over the $\mathcal{O}(n)$ clock qudits of the sub-instance and verifies whether each qudit is present in a Π_{init} clause and a Π_{prop} clause with some property. This takes $\mathcal{O}(n) \cdot \mathcal{O}(poly(n)) = \mathcal{O}(poly(n))$ time. In step (CC.3), the algorithm traverses the TACC and for each clock qudit, it checks the clauses that act on it. Similarly to step (CC.2), this step also takes $\mathcal{O}(poly(n))$ time. Step (CC.4) takes $\mathcal{O}(poly(n))$ time as the algorithm simply looks over all of the clauses that may act on a qudit. Finally, for step (CC.5), note that there are at most $\mathcal{O}(poly(n))$ clauses that are not part of a TACC. Thus, removing them from the instance also takes $\mathcal{O}(poly(n))$ time.

4.2.3 Completeness and soundness

This section is no different than in the previous construction. Section 4.1 covers an exhaustive list of instances that can be decided based on their structure. Since the algorithm above is based on this list, it consistently rejects instances whose structure implies two or more clauses cannot be satisfied simultaneously by any state. Conversely, it never rejects instances whose structure could be consistent with a satisfying state. If the classical algorithm alone determines the instance, then both the completeness and soundness conditions are clearly satisfied. However, if the instance is not decided by the classical algorithm, then there must be some sub-instances that require a quantum approach.

As mentioned, the quantum sub-instances in this construction are identical to those of the previous construction. The quantum algorithm that decides them is also the same here, and so all of the arguments presented in the previous construction also apply here.

4.3 Hardness

Here, we show that SLCT-QSAT is BQP_1-hard, thus completing the proof of Theorem 2.

Let $U_x = U_L \dots U_1$ with $U_i \in \mathcal{G}$ and $L = poly(n)$ be the BQP_1 circuit where given an instance x of an arbitrary problem A, decides whether $x \in A_{yes}$ or $x \in A_{no}$. The input to the circuit is the q-qubit ancilla register $|0\rangle^{\otimes q}$, where $q = poly(n)$. The reduction from x to an instance x' of SLCT-QSAT requires $q+(L+1)$ qudits (2 qudits less than in the previous construction), each of dimension 6, and is the following:

1. Choose n qudits to serve as the logical qudits of the computation and $L+1$ qudits to serve as the clock qudits. Define an ordering for each type of qudit.

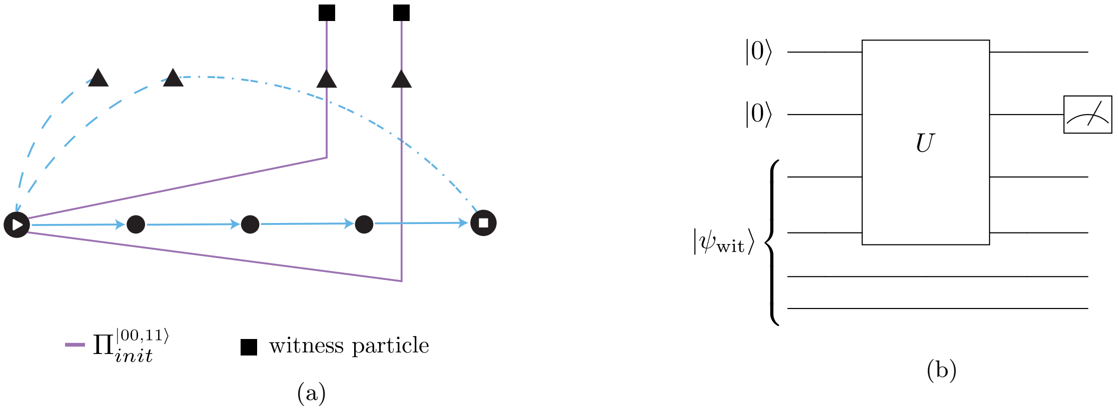


Figure 12: (a) Toy example of an input “quantum” instance with a TACC of length $L = 4$, acting on four logical qudits and two witness qudits. Although not illustrated, the Π_{prop} clauses are assumed to have unitaries $U_1, \ldots, U_4$ which act only on the logical qudits of the instance. These unitaries define a circuit $U = U_4 U_3 U_2 U_1$. (b) Quantum circuit representing the instance on the left.

2. For all $i \in [q]$, create a Π_{init} clause acting on c_0 and l_i.
3. For all $j \in [L]$, create a Π_{prop} clause with associated unitary U_{j+1}, which acts on clock qudits c_j and c_{j+1} (with c_j being the predecessor of c_{j+1}) and the two logical qudits that the unitary originally acts on.
4. Create one Π_{out} clause acting on c_L and l_{ans}.

The resulting instance x' consists of a one-dimensional clock chain of length L ($L + 1$ clock qudits) with a unique direction, no simultaneous propagation clauses, and fully initialized logical qudits.

If x is a yes-instance of A, x' is satisfiable and is satisfied entirely by the history state of Eq. (32).

On the other hand, when x is a no-instance of A, we again show that x' has no small eigenvalues by expressing the instance in terms of Hamiltonians as

$$H = H_{roles} + H_{clock} + H_{prop} + H_{init} + H_{out}. \tag{52}$$

The definitions of these Hamiltonians remains mostly the same as in the previous construction, except that now H_{roles} does not contain any Π_E term. Moreover, observe that in contrast with Eq. (33), the equation above has no H_{aux} term. We now show that despite these differences, the analysis within the null space of H_{clock} from the previous section still applies, and we can again conclude that the smallest eigenvalue of H scales with the inverse of a polynomial.

Recall that in the construction of Section 3, the constraints on the auxiliary subspaces determined that the states within the null space of H_{aux} were those with clock qudits with a unique successor and predecessor, and were all preceded by the clock qudit with the Π_{init} clauses and succeeded by the one with the Π_{out} clauses.[27] As shown in Lemmas 4.5 and 4.6, the satisfying states of SLCT-QSAT instances must also satisfy these properties. A quick inspection of the proofs of these statements shows that this is due to the clock constraints. Therefore, states within the null space of H_{clock} must also satisfy these two properties, bypassing the need of an H_{aux} term.

5 QCMA-complete Problem

The constructions from Sections 3 and 4 can be modified to generate QCMA_1-complete problems. However, since the second problem we presented is an improvement over the first, we only elaborate on the latter.

[27] When demonstrating that the problem was in BQP_1, the constraints also helped determine that satisfiable sub-instances could not have more clauses that extended past the ACC. This is not relevant when proving hardness as the instance we construct already satisfies this property.

Moreover, since $\mathsf{QCMA}_1^{\mathcal{G}} = \mathsf{QCMA}$ [34], this results in a QCMA-complete problem. While there are already many problems known to be complete for this class [43, 44, 45, 46], none of them are strong QCSPs.

5.1 Construction overview

At the beginning of Section 3.3, we argued that the unconstrained or "free" logical qudits of an instance allowed one to guess what state of these qudits (the witness state) might satisfy the instance. This freedom made the problem more difficult and thus not likely contained in BQP_1. In fact, one could show that this problem is QMA_1-complete.[28] For this reason, we introduced the undefined state $|?\rangle$, which simplified these instances and made them decidable in BQP_1. In this construction, we seek to construct a problem that sits in between these two classes so it is QCMA-complete. To accomplish this, we desire to have "free" logical qudits to accommodate a witness state that helps verify whether the instance is satisfiable, but have some sort of constraint to demand that the state is classical.[29]

In practice, creating these constraints is a difficult task because if there are two states that satisfy a clause, any superposition of the two will do so as well. Instead, we set the constraints such that if there exists a quantum witness state that is part of a satisfying state, there is also a classical witness state. Loosely, we accomplish this by defining new *witness qudits* and create a new constraint $\Pi_{init}^{|00,11\rangle}$ that connects a witness qudit with a logical one, and require that they are both either $|00\rangle$, $|11\rangle$, or in a superposition of the two. In this way, the two qudits are partially "free" as there is some freedom to their state yet posses some structure. Importantly, we ensure that the witness qudits do not form part of the computation after this initial point.

To see why this leads to the desired effect, consider the toy instance of Fig. 12 and suppose there exists a state $|\psi_{\text{wit}}\rangle$ of the four "free" qudits that leads to a satisfying state. Observe that to satisfy the $\Pi_{init}^{|00,11\rangle}$ clauses, this state must be of the form $|\psi_{\text{wit}}\rangle = (\alpha_{00}|0000\rangle + \alpha_{01}|0011\rangle + \alpha_{10}|1100\rangle + \alpha_{11}|1111\rangle)_{L_1,W_1,L_2,W_2}$ with $\sum_{b\in\{0,1\}^2}|\alpha_b|^2 = 1$, which we can rewrite in a more convenient form as $|\psi_{\text{wit}}\rangle = \sum_{b\in\{0,1\}^2}\alpha_b\,|b\rangle_L\otimes|b\rangle_W$. Then, the state that satisfies all clauses of the instance is the history state

$$
\begin{aligned}
|\psi_{hist}\rangle &= \frac{1}{\sqrt{5}}\sum_{t=0}^{4}\left[U_t\dots U_0\,|00\rangle\otimes|\psi_{\text{wit}}\rangle\right]\otimes|\underbrace{d\dots d}_{t}\,a_t\,\underbrace{r\dots r}_{4-t}\rangle \\
&= \frac{1}{\sqrt{5}}\sum_{t=0}^{4}\sum_{b\in\{0,1\}^2}\alpha_b\,|\xi_b^t\rangle\otimes|b\rangle_W\otimes|\underbrace{d\dots d}_{t}\,a_t\,\underbrace{r\dots r}_{4-t}\rangle\,,
\end{aligned}
\tag{53}
$$

where $|\xi_b^t\rangle := U_t\dots U_0\,|00\rangle\otimes|b\rangle_L$. Let us show that there is also a classical witness that leads to a satisfying history state. First, observe that any basis state $|b\rangle_L\otimes|b\rangle_W$ with $|\alpha_b| > 0$ from the decomposition of the witness satisfies the $\Pi_{init}^{|00,11\rangle}$ clauses. Consequently, the history state above but with initial state $|00\rangle\otimes|b\rangle_L\otimes|b\rangle_W$ satisfies the $\Pi_{init}^{|0\rangle}$, $\Pi_{init}^{|00,11\rangle}$, and Π_{prop} clauses of the instance. To show that this state also satisfies the Π_{out} clause, recall that the clause is satisfied if at time $t = 4$, the probability that the second qubit yields outcome "1" when measured is 1. This probability can be written as

$$
\begin{aligned}
\Pr(\text{outcome } 1) &= \sum_{b,b'\in\{0,1\}^2}\alpha_b\alpha_{b'}^*\,\langle\xi_{b'}^4|\otimes\langle b'|\,\Pi^{(1)}\,|\xi_b^4\rangle\otimes|b\rangle \\
&= \sum_{b\in\{0,1\}^2}|\alpha_b|^2\,\langle\xi_b^4|\,\Pi^{(1)}\,|\xi_b^4\rangle
\end{aligned}
\tag{54}
$$

where $\Pi^{(1)} := |1\rangle\langle 1|_2\otimes I_{rest}$, and in the second line we observed that $\langle b'|b\rangle = \delta_{b,b'}$. Then, by the assumption that the instance is satisfiable, it must be that $\langle\xi_b^4|\,\Pi^{(1)}\,|\xi_b^4\rangle = 1$ for all basis states $|b\rangle$ with $|\alpha_b| > 0$. This can also be understood as the probability that at the end of the circuit the second qubit yields outcome "1" when the witness is the basis state $|b\rangle\otimes|b\rangle$. Therefore, the history state

[28] We do not prove this in this paper since the much simpler k-QSAT problem is already known to be QMA_1-complete for $k \geq 3$ [3, 4].

[29] We continue using the undefined state for logical qudits whose initial state is not constrained so the difficulty of the problem does not become QMA.

$$|\psi_{hist}\rangle = \frac{1}{\sqrt{5}} \sum_{t=0}^{4} \left[U_t \dots U_0 \left|00\right\rangle \otimes \left|b\right\rangle_L\right] \otimes \left|b\right\rangle_W \otimes |\underbrace{d \dots d}_{t}\, a_t\, \underbrace{r \dots r}_{4-t}\rangle \tag{55}$$

with classical witness $|\psi_{\text{wit}}\rangle = |b\rangle_L \otimes |b\rangle_W$ also satisfies all clauses of the instance.

While this toy example accurately conveys the idea behind the construction, there are more details we need to discuss to prove Theorem 3. For this, let us define the problem and the new clause.

5.2 Problem definition

The problem we claim is QCMA-complete is is the following:

Definition 5.1 (Witnessed SLCT-QSAT). *The problem* Witnessed SLCT-QSAT *is a QCSP defined on the 8-dimensional Hilbert space*

$$\mathcal{H} = \text{span}\{|0_L\rangle, |1_L\rangle, |?_L\rangle\} \oplus \text{span}\{|0_W\rangle, |1_W\rangle\} \oplus \text{span}\{|r_C\rangle, |a_C\rangle, |d_C\rangle\}, \tag{56}$$

which now consists of a logical, a witness, and a clock subspace. Besides the operators O_{init}, O_{prop}, and O_{out} defined previously in Eqs. (49)–(51), the problem contains a new 3-local operator

$$O_{init}^{|00,11\rangle} := \Pi_{L,1} + \Pi_{W,2} + \Pi_{C,3} + \Pi_{start,3} + |?\rangle\langle ?|_1 \otimes |a\rangle\langle a|_3 + (I - |00\rangle\langle 00| - |11\rangle\langle 11|)_{1,2} \otimes |a\rangle\langle a|_3\,, \tag{57}$$

where $\Pi_W := I - (|0\rangle\langle 0|_W + |1\rangle\langle 1|_W)$ is the projector to the witness subspace. As before, $\Pi_{init}^{|00,11\rangle}$ refers to the projector with the same null space as $O_{init}^{|00,11\rangle}$.

The promise remains the same as before. We are promised that for every instance considered either (1) there exists a quantum *state $|\psi_{sat}\rangle$ on n 8-dimensional qudits such that $\Pi_i|\psi_{sat}\rangle = 0$ for all i, or (2) $\Sigma_i \langle\psi|\Pi_i|\psi\rangle \geq 1/poly(n)$ for all* quantum *states $|\psi\rangle$.*

The goal is to output "YES" if (1) is true, or output "NO" otherwise.

Before showing that this problem is contained within QCMA and is also QCMA-hard, let us make some observations. Given that there are now multiple initialization clauses, we will refer to the original Π_{init} clause used in Sections 3 and 4 as $\Pi_{init}^{|0\rangle}$, and use $\Pi_{init}^{|\cdot\rangle}$ when we do not care whether the clause is $\Pi_{init}^{|0\rangle}$ or $\Pi_{init}^{|00,11\rangle}$. Observe that $|?\rangle\langle ?| \otimes |a\rangle\langle a|$ prevents the logical qudit from being $|?\rangle$ when the clock qudit is $|a\rangle$. Then, together with the term $(I - |00\rangle\langle 00| - |11\rangle\langle 11|) \otimes |a\rangle\langle a|$, this implies that the logical and witness qudits must be $|00\rangle$, $|11\rangle$, or any superposition of the two. Clearly, $\Pi_{init}^{|00,11\rangle}$ sets a "weaker" constraint than $\Pi_{init}^{|0\rangle}$ as the logical qudit is not constrained to a single state, but it is also a "stronger" constraint than having completely free witness and logical qudits.

Now, let us show that this problem is in QCMA.

5.3 In QCMA

Recall that to demonstrate that the problem is in QCMA, we have to show that when x is a yes-instance, there is a basis state $|y_x\rangle$ that convinces us with certainty that the instance is satisfiable. On the other hand, if x is a no-instance, we have to show that no classical basis state is accepted with high probability.

We demonstrate this result by following the same proof structure as in the previous constructions, starting with the satisfiability of clock components and shared logical (and now witness) qudits. To begin, observe that the introduction of the $\Pi_{init}^{|00,11\rangle}$ clause does not create new conflicts in the clock components. However, for the shared logical and witness qudits, there are six new cases to consider: three where the clauses stem from the same clock component, and three where they stem from different clock components. Here, we only state and prove the first three. The proofs of the remaining three cases follow easily by first recalling Lemma 3.8, which states that the logical qudits cannot be present in Π_{prop} clauses and so their state must be fixed, and then the proof can be completed by slightly adapting the proofs shown here. We also refer the reader to Section 6.3 of the following coRP construction where there is only one case and we prove both versions of the statement. The first three cases are the following.

Lemma 5.1. *Consider a clock component that has at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. Suppose there are two $\Pi_{init}^{|00,11\rangle}$ clauses acting on the same clock and logical qudit, but distinct witness qudits. The only states of the logical and two witness qudits that satisfy the $\Pi_{init}^{|00,11\rangle}$ clauses are of the form $\alpha_0 |000\rangle + \alpha_1 |111\rangle$ for any $\alpha_0, \alpha_1 \in \mathbb{C}$ that satisfies $|\alpha_0|^2 + |\alpha_1|^2 = 1$.*

Proof. Recall from Proposition 4.1 that if the instance is satisfiable, both clauses must enforce their constraints on the shared logical qudit. Each $\Pi_{init}^{|00,11\rangle}$ then requires its logical and witness qudit to be in the same basis state and also forbids the logical qudit from being $|?\rangle$. Hence, these three qudits must be either $|000\rangle$ or $|111\rangle$. Furthermore, as mentioned in the overview, any valid superposition of the two also satisfies these clauses, and so the satisfying state is in general $\alpha_0 |000\rangle + \alpha_1 |111\rangle$. □

Lemma 5.2. *Consider a clock component that has at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. Suppose there are two $\Pi_{init}^{|00,11\rangle}$ clauses acting on the same clock and witness qudit, but distinct logical qudits. The only states of the logical and two witness qudits that satisfy the $\Pi_{init}^{|00,11\rangle}$ clauses are of the form $\alpha_0 |000\rangle + \alpha_1 |111\rangle$ for any $\alpha_0, \alpha_1 \in \mathbb{C}$ that satisfies $|\alpha_0|^2 + |\alpha_1|^2 = 1$.*

Proof. The setup is similar to that in Lemma 5.1, except now there are two witness qudits and one logical qudit. Despite this difference, it is easy to see that the proof of that statement also applies here. □

Lemma 5.3. *Consider a clock component that has at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. Suppose a $\Pi_{init}^{|0\rangle}$ and a $\Pi_{init}^{|00,11\rangle}$ clause act on the same clock and logical qudits. If the witness qudit is in the state $|1\rangle$, the instance is unsatisfiable.*

Proof. The first $\Pi_{init}^{|0\rangle}$ clause requires the logical qudit to be in the state $|0\rangle$. The only way to satisfy $\Pi_{init}^{|00,11\rangle}$ is if both the logical and witness qudits are in the same state. Hence, if the witness qudit is in the state $|1\rangle$, the clauses cannot be satisfied simultaneously. □

In a sense, these lemmas show that it is possible for a witness qudit to determine the state of multiple logical qudits, or vice versa. In a general input instance where the witness and logical qudits may be intricately connected by $\Pi_{init}^{|00,11\rangle}$ clauses, there is a general form of the state of these qudits that satisfies these clauses.

Proposition 5.1. *Consider a clock component that has at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. Suppose the sub-instance contains p witness qudits connecting to r logical qudits via $\Pi_{init}^{|00,11\rangle}$ clauses. Depending on the instance, r may be smaller or larger than p. Furthermore, suppose that a subset of the logical qudits are also acted on by $\Pi_{init}^{|0\rangle}$ clauses. The state of the witness and logical qudits that satisfies the $\Pi_{init}^{|00,11\rangle}$ and $\Pi_{init}^{|0\rangle}$ clauses of the sub-instance (the witness state) is of the form*

$$|\psi_{\mathrm{wit}}\rangle = \sum_{b\in\{0,1\}^p} \alpha_b \, |f(b)\rangle_L \otimes |b\rangle_W \,, \tag{58}$$

where $\sum_{b\in\{0,1\}^p} |\alpha_b|^2 = 1$ and $f : \{0,1\}^p \mapsto \{0,1\}^r$ is a function that depends on the specific arrangement of the $\Pi_{init}^{|00,11\rangle}$ clauses within the component. Moreover, there is the additional restriction that the basis states where the logical qudits in Z are in state $|1\rangle$ must have $\alpha_b = 0$.

Proof. Observe that, similar to clock components, the logical and witness qudits can be seen as either forming joint or disjoint sets judging by their connections via $\Pi_{init}^{|00,11\rangle}$ clauses. Suppose that there there are m such disjoint sets, and let r_i and p_i with $i \in [m]$ be the number of logical and witness qudits of the i-th set. Clearly, $r = \sum_{i\in[m]} r_i$ and $p = \sum_{i\in[m]} p_i$. By generalizing the statements of Lemma 5.1 and Lemma 5.2, it is evident that the state of any connected set must be of the form $\alpha_0 |0\rangle^{\otimes(r_i+p_i)} + \alpha_1 |1\rangle^{\otimes(r_i+p_i)}$, which we can rewrite as $\sum_{b'_i\in\{0,1\}} \alpha_{b'_i} |b'_i\rangle_L^{\otimes r_i} \otimes |b'_i\rangle_W^{\otimes p_i}$ where we have separated the logical and witness qudits. The overall state is then the m-fold tensor product of such states, given by $\sum_{b'\in\{0,1\}^m} \alpha_{b'} |b'_0\rangle^{\otimes r_0} \dots |b'_{m-1}\rangle^{\otimes r_{m-1}} \otimes |b'_0\rangle^{\otimes p_0} \dots |b'_{m-1}\rangle^{\otimes p_{m-1}}$ where we let $\alpha_{b'} := \alpha_{b'_0} \dots \alpha_{b'_{m-1}}$. This is almost Eq. (58), except now b' is a bitstring of length m, referring to the number of connected components instead of the number of witness qudits. We can expand the bitstring to be of size p, if we let $\alpha_b = 0$ for all $b \in \{0,1\}^p$ when the state $|b\rangle \neq |b'_0\rangle^{\otimes p_0} \dots |b'_{m-1}\rangle^{\otimes p_{m-1}}$, and $\alpha_b = \alpha_{b'}$ if $|b\rangle = |b'_0\rangle^{\otimes p_0} \dots |b'_{m-1}\rangle^{\otimes p_{m-1}}$. Finish. What about f? □

This concludes the discussion about the satisfiability of clock components. The subsequent discussions about simultaneous propagation clauses and active dots remains the same as before. We now present the algorithm that decides all instances of this problem.

5.3.1 Algorithm

The input to the algorithm is the instance x and a classical witness state $|y_x\rangle$ of size equal to the number of witness qudits in the instance. We do not consider a state of the "free" logical qudits, since as just shown, their state can be determined by their connections to witness qudits. Even if such a state is provided, the algorithm below will overwrite it, which spares us from having to add additional checks to verify that the states of witness and logical qudits match. The algorithm that decides all instances of the problem is the following.

The algorithm has the same first two steps as in Section 4.2.1. We now add two new steps in between (2) and (3) to account for the new lemmas above:[30]

(QCMAa) For every logical qudit in the instance, check whether it is connected to several witness qudits. If the states of the witness qudits are not the same, reject. Correctness is given by Lemma 5.1.

(QCMAb) For every logical qudit in the instance, check whether it is present in both $\Pi_{init}^{|0\rangle}$ and $\Pi_{init}^{|00,11\rangle}$ clauses. If there is such a qudit and one of the connected witness qudits is in the state $|1\rangle$, reject. Correctness is given by Lemma 5.3.

We note that these two instructions account for both scenarios: cases where the clauses stem from the same clock component and those from different components. Normally, the latter cases would be evaluated in step (4); however, in this particular construction, they should be evaluated at this time since the next step initializes the state of all logical qudits to their corresponding values. This slight adjustment makes no difference in the correctness of the algorithm as no sub-instances are accepted in steps (3) or (4) and so it is safe to permute these around.

Step (3) also requires a small modification:

(3) *Identify the undefined* Π_{prop} *clauses.* For every logical qudit, check whether it is acted on by at least one $\Pi_{init}^{|0\rangle}$ or $\Pi_{init}^{|00,11\rangle}$ clause. If it is not, mark all Π_{prop} clauses this qudit is part of as undefined.

The rest of the algorithm proceeds mostly in the same way as in Section 4.2.1, the only other remaining difference is on the input of the quantum algorithm in step (6.1). If we assume the TACC has q logical qudits initialized only by $\Pi_{init}^{|0\rangle}$ clauses, and p logical qudits initialized by $\Pi_{init}^{|\Phi^+\rangle}$ clauses, the step in the algorithm is then:

(6.1) Initialize a register containing $q+2$ qubits, each initialized to $|0\rangle$ (the two extra qubits are the ancillas used in $\mathcal{C}_{\mathcal{G}}$). Initialize a second register of p qubits where the state of each qubit is chosen to match the state of the witness qudits it connects to.

Finally, we point out that as promised, the witness qudits are not acted on by any of the unitaries of the circuit.

5.3.2 Runtime

The runtime of the algorithm can be easily seen to be in polynomial time since the new steps (QCMAa) and (QCMAb) simply iterate over $\mathcal{O}(n)$ logical qudits and for each qudit the algorithm searches over a list of $\mathcal{O}(poly(n))$ clauses.

[30] A step to include Lemma 5.2 is not necessary with the given form of the witness.

5.3.3 Completeness

Let us demonstrate that if x is a yes-instance of Witnessed SLCT-QSAT, there is a classical basis state $|y_x\rangle$ that makes the algorithm accept with certainty. First, let us address the question about the existence of the classical basis state, and then show that the instance is accepted with certainty if this state is provided.

The key to answering this first question relies on analyzing each sub-instance separately. We will show that for each sub-instance whose TACC has $\Pi_{init}^{|00,11\rangle}$ clauses connecting p witness and r logical qudits, there exists a $(p+r)$-qubit basis state that helps determine the sub-instance is satisfiable. Then, $|y_x\rangle$ will be the union of these basis states for all sub-instances. Although some sub-instances may share logical and witness qudits, the assumption that the instance is satisfiable and Lemmas 5.1–5.3 assure us that there is a state of these qudits for which all sub-instances can be simultaneously satisfied.

To begin, recall that there are two types of sub-instances: those that can be decided classically, and those that make use of a quantum sub-routine. Let us discuss each type separately. We make the assumption that all sub-instances below contain at least one $\Pi_{init}^{|00,11\rangle}$ clause, as otherwise they would be no different than those of Section 4 whose satisfiability is independent of a witness.

Classical witness for classical sub-instances

By the lemmas of Section 3.3.1, we know that each qudit of the instance must serve a single role and all sub-instances must be of one of two types: either they have no ACCs, or they have truncated TACCs with no simultaneous propagation clauses. In addition, the shared logical qudits between sub-instances must agree with the lemmas of Section 3.3.2.

For sub-instances without ACCs, the witness and logical qudits are irrelevant as the sub-instance can be satisfied by simply choosing the state of the clock qudits appropriately. Thus, the free qudits can be in any basis state.

For sub-instances with truncated TACCs, there must be a state $|\psi_{\text{wit}}\rangle$ of the free qudits such that the truncated history state

$$|\psi_{thist}\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} \left[U_{t,1} \dots U_{0,1} \, |0\rangle^{\otimes q} \otimes |\psi_{\text{wit}}\rangle \right] \otimes |\underbrace{d \dots d}_{t} a_t \underbrace{r \dots r}_{T-t}\rangle \tag{59}$$

satisfies all clauses of the TACC. Here, $T < L$ and $|0\rangle^{\otimes q}$ are the logical qudits within the TACC that are present in $\Pi_{init}^{|0\rangle}$ clauses (including $\Pi_{init}^{|0\rangle}$ clauses from other sub-instances), but not present in any $\Pi_{init}^{|00,11\rangle}$ clause. On the other hand, $|\psi_{\text{wit}}\rangle$ might contain logical qudits that are acted on by $\Pi_{init}^{|0\rangle}$ clauses and thus constrained to be $|0\rangle$ (these qudits are then not truly "free", but grouping them this way helps simplify the arguments below). As stated in Proposition 5.1, the witness must be of the form $|\psi_{\text{wit}}\rangle = \sum_{b\in\{0,1\}^p} \alpha_b \, |f(b)\rangle_L \otimes |b\rangle_W$ with $\sum_b |\alpha_b|^2 = 1$. We can show that there is a classical witness state using similar arguments to those given in the toy example. First, observe that all $|f(b)\rangle \otimes |b\rangle$ with $|\alpha_b| > 0$ in the decomposition of the witness satisfy the $\Pi_{init}^{|0\rangle}$ and $\Pi_{init}^{|00,11\rangle}$ clauses of the sub-instance. Then, given that the sub-instance does not contain simultaneous propagation clauses and the sub-instance is truncated, the Π_{prop} and Π_{out} clauses are satisfied regardless of the initial state. In other words, replacing $|\psi_{\text{wit}}\rangle$ with any $|f(b)\rangle \otimes |b\rangle$ (as long as $|\alpha_b| > 0$) in the history state above also leads to a satisfying history state.

Classical witness for quantum sub-instances

Recall that quantum sub-instances must have one of the three forms shown in Fig. 6. Let us begin with the third type of sub-instance as these are similar to the truncated classical sub-instances above, except their TACC now contains simultaneous propagation clauses.

Sub-instances of this first type are also satisfied by Eq. (59). By the same argument, any basis state $|f(b)\rangle \otimes |b\rangle$ with $|\alpha_b| > 0$ in the superposition of the witness state satisfies the $\Pi_{init}^{|0\rangle}$ and $\Pi_{init}^{|00,11\rangle}$ clauses. The Π_{out} clauses of the instance are again trivially satisfied, and the lone Π_{prop} clauses are satisfied independently from the initial state. Then, it only remains to show that simultaneous propagation clauses can also be satisfied with a classical witness state as input.

Proposition 5.2. *Suppose two Π_{prop} clauses with unitaries $U_{t+1,i}$ and $U_{t+1,j}$ within the TACC of a sub-instance are satisfiable, i.e. $U_{t+1,i}\,|\phi_t\rangle = U_{t+1,j}\,|\phi_t\rangle$ where $|\phi_t\rangle = U_{t,1}\dots U_{1,1}\,|\phi_0\rangle$ is the state of the data qudits at time t (see Proposition 3.3). For sub-instances of* Witnessed SLCT-QSAT, $|\phi_0\rangle = |0\rangle^{\otimes q} \otimes |\psi_{\text{wit}}\rangle$ *where in general $|\psi_{\text{wit}}\rangle = \sum_{b\in\{0,1\}^p} \alpha_b\,|f(b)\rangle_L \otimes |b\rangle_W$ (see Proposition 5.1). Replacing $|\psi_{\text{wit}}\rangle$ with any of its basis states $|f(b)\rangle \otimes |b\rangle$ that has $|\alpha_b| > 0$ still results in a state $|\phi_t\rangle$ that satisfies the simultaneous Π_{prop} clauses.*

Proof. Let us write the intermediate state of the computation as $|\phi_t\rangle = \sum_{b\in\{0,1\}^p} \alpha_b\,|\xi_b^t\rangle \otimes |b\rangle$, where $|\xi_b^t\rangle = U_{t,1}\dots U_{1,1}\,|0\rangle^{\otimes q} \otimes |f(b)\rangle$. Then, by assumption

$$\begin{aligned} 1 &= \langle\phi_t|\, U_{t+1,j}^{\dagger} U_{t+1,i}\,|\phi_t\rangle \\ &= \sum_{b,b'\in\{0,1\}^p} \alpha_b \alpha_{b'}^{*}\, \langle\xi_{b'}^t| \otimes \langle b'|\, U_{t+1,j}^{\dagger} U_{t+1,i}\,|\xi_b^t\rangle \otimes |b\rangle \\ &= \sum_{b\in\{0,1\}^p} |\alpha_b|^2\, \langle\xi_b^t|\, U_{t+1,j}^{\dagger} U_{t+1,i}\,|\xi_b^t\rangle\,. \end{aligned} \tag{60}$$

For this equation to be true, it must be that $\langle\xi_b^t|\, U_{t+1,j}^{\dagger} U_{t+1,i}\,|\xi_b^t\rangle = 1$ for all $b \in \{0,1\}^p$ with $|\alpha_b| > 0$. Consequently, the state $|\phi_t'\rangle = |0\rangle^{\otimes q} \otimes |f(b)\rangle \otimes |b\rangle$, where $|f(b)\rangle \otimes |b\rangle$ is any basis state of $|\psi_{\text{wit}}\rangle$ with $|\alpha_b| > 0$, can satisfy the Π_{prop} clauses. □

An important observation here is that since any basis state $|f(b)\rangle \otimes |b\rangle$ with $|\alpha_b| > 0$ can help satisfy a pair of Π_{prop} clauses, any single state we either pick or receive as input will also help satisfy all other subsequent simultaneous Π_{prop} clauses of the TACC.

For the remaining two types of sub-instances, we simply consider the second as it is more general than the first. For this type of sub-instance, the TACC encompasses the whole ACC, and the satisfying state is given by Eq. (59), except now $T = L$. As a consequence, the Π_{out} clauses are no longer trivially satisfied. As we have shown previously, any basis state $|f(b)\rangle \otimes |b\rangle$ in the superposition of the quantum witness makes a valid classical witness that leads to a history state that satisfies the $\Pi_{init}^{|0\rangle}$, $\Pi_{init}^{|00,11\rangle}$, and Π_{prop} clauses. To see that this state also satisfies the Π_{out} clauses, we can simply generalize the analysis of the toy example. Here, the sub-instances may now have a more erratic arrangement of their $\Pi_{init}^{|00,11\rangle}$ and $\Pi_{init}^{|0\rangle}$, but as shown previously, the witness state can be always be expressed as $|\psi_{\text{wit}}\rangle = \sum_{b\in\{0,1\}^p} \alpha_b\,|f(b)\rangle \otimes |b\rangle$. Then, following the same reasoning as in the toy example, it is easy to show that $\langle\xi_b^L|\,\Pi^{(1)}\,|\xi_b^L\rangle = 1$ for all b, where now $|\xi_b^L\rangle = U_L \dots U_1\,|0\rangle^{\otimes q} \otimes |f(b)\rangle$. Therefore, any classical witness state $|f(b)\rangle \otimes |b\rangle$ with non-zero amplitude in $|\psi_{\text{wit}}\rangle$ also leads to acceptance.

Instance-wide classical witness and perfect completeness

We have just shown that if the instance is satisfiable, for each sub-instance with a TACC and "free" qudits, there exists a classical state of its free qudits such that the sub-instance can be satisfied. Moreover, there may be more than one such state for each sub-instance. One might then think that the overall classical witness state $|y_x\rangle$ could be the union of any arbitrary combination of these classical witnesses, and while almost correct, there is one caveat. Since the sub-instances may not be completely independent and may share logical and witness qudits, it is possible that the state of one qudit leads to a satisfying state in one sub-instance but not in the others. However, since the instance must be satisfiable, there should exist a state of the qudit that can help satisfy all of them. Thus, $|y_x\rangle$ must be some particular combinations of the classical witnesses of each individual sub-instance.

To show perfect completeness, observe that by the correctness of the lemmas in this section and the previous ones, if presented with $|y_x\rangle$, the algorithm—which is based on these lemmas—never wrongfully rejects.

5.3.4 Soundness

Now, let us show that when x is a no-instance, the algorithm rejects the instance with high probability for any input classical witness state.

To begin, assume that there's a pair of clauses that cannot be mutually satisfied. If the problem lies in the roles of qudits, the clock component, or in the shared logical qudits between sub-instances where the clauses considered are $\{\Pi_{init}^{|0\rangle}, \Pi_{prop}, \Pi_{out}\}$, the algorithm always rejects regardless of the witness state. This again follows from the correctness of the analysis of clock components and shared logical qudits from previous sections of the paper. If the unsatisfiable clauses are those involving the witness and logical qudits, then no matter which classical witness state is input to the algorithm, the algorithm rejects the instance as it is based on Lemmas 5.1–5.3 (and the other three which we didn't explicitly write down) which cover all possible scenarios.

If x is a no-instance and is not rejected before the quantum subroutine, then the unsatisfiable clauses must lie in one of the quantum sub-instances. We can also assume that the witness state can mutually satisfy the $\Pi_{init}^{|00,11\rangle}$ and $\Pi_{init}^{|0\rangle}$ clauses as bad witnesses would be caught in steps (QCMAa) and (QCMAb) of the algorithm. Then, either the simultaneous propagation clauses cannot be satisfied, or the final state of the computation results in a state where the logical qudits present in Π_{out} clauses are $|0\rangle$. This is the same scenario as in Section 3.4.4, and since the quantum subroutine here is the same as in that section, the probability that the algorithm accepts the instance is $\leq 1/3$.

5.4 QCMA-hard

Here, we show that Witnessed SLCT-QSAT is QCMA-hard, thus completing the proof of Theorem 3.

Let $U_x = U_L \dots U_1$ with $U_i \in \mathcal{G}$ and $L = poly(n)$ be the QCMA verifier circuit that given an instance x of a problem A, decides whether $x \in A_{yes}$ or $x \in A_{no}$. The input to the circuit is the p-qubit classical witness state $|y_x\rangle$, and the q-qubit ancilla register $|0\rangle^{\otimes q}$, where p, q are two polynomials scaling with $n = |x|$. Denote as ans the logical qubit that when measured provides this decision. The reduction $x \mapsto x'$ requires $2p + q + L + 1$ qudits and is the following:

1. Choose $2p$ qudits to serve as the logical and witness qudits of the computation, choose other q qudits to serve as the ancilla qudits, and $L + 1$ qudits to serve as the clock qudits. For each class of qudit, define an ordering.

2. For all $i \in [q]$, create a $\Pi_{init}^{|0\rangle}$ clause acting on c_0 and l_i.

3. For all $j \in [p]$, create a $\Pi_{init}^{|00,11\rangle}$ clause acting on c_0, l_{q-1+j}, and w_j.

4. For all $k \in [L]$, create a Π_{prop} clause with associated unitary U_{k+1} acting on clock qudits k and $k + 1$ (with k being the predecessor of $k + 1$) and the two logical qudits that the unitary originally acts on.

5. Create one Π_{out} clause acting on c_L and l_{ans}.

Similarly to the previous problems, the resulting instance is one-dimensional, with no simultaneous propagation clauses and initialized logical qudits (some with $\Pi_{init}^{|0\rangle}$ clauses and others with $\Pi_{init}^{|00,11\rangle}$).

5.4.1 Completeness and soundness

If x is a yes-instance, then the Witnessed SLCT-QSAT instance x' is satisfied with certainty by the history state

$$|\psi_{hist}\rangle = \frac{1}{\sqrt{L+1}} \sum_{t=0}^{L} U_t \dots U_0 |0\rangle^{\otimes q} \otimes |y_x\rangle \otimes |\underbrace{d \dots d}_{t} a_t \underbrace{r \dots r}_{L-t}\rangle . \tag{61}$$

When x is a no-instance, the only nontrivial change in Eq. (52) happens within H_{init}. This term now collects the $(I - |0\rangle\langle 0|) \otimes |a\rangle\langle a|$ terms (q of them), the $|?\rangle\langle ?| \otimes |a\rangle\langle a|$ terms (p of them), and the $(I - |00\rangle\langle 00| - |11\rangle\langle 11|) \otimes |a\rangle\langle a|$ terms (p of them). Thus, it is only necessary to show that $H|_{\mathcal{S}_{clock}}$ has no small eigenvalues in both cases where there is at least one logical qudit in the $|?\rangle$ state, and when there's no such state. We note that we cannot simply repeat the argument of Ref. [3] for the QMA-complete 3-QSAT problem because the soundness of problem A—"for any classical witness state $|y_x\rangle$, $\Pr[U_n$ accepts $(x, y_x)] \leq 1/3$"—is restricted to classical states instead of arbitrary quantum states.

At least one undefined logical qudit
This case is quite similar to the one in the BQP_1 problem. We still have that the Hamiltonian H within this subspace is $H|_{\mathcal{S}_{clock}} = H_{init} + H_{prop}(T,u)$. Then, it is easy to see that even with the additional terms, H_{init} remains a diagonal matrix and $\gamma(H_{init}) \geq 1$. On the other hand, H_{prop} remains unchanged and so $\gamma(H_{prop}) \geq c/T^2$ for some positive constant c. The null spaces of H_{init} and $H_{prop}(T,u)$ are:

$$\mathcal{S}_{init} = \mathcal{D}_u \otimes \operatorname{span}\{|C_1\rangle, \ldots, |C_T\rangle\} \qquad \text{and} \qquad \mathcal{S}_{prop} = \mathcal{D}_u \otimes \frac{1}{\sqrt{T+1}}\sum_{t=0}^{T}|C_t\rangle\,, \tag{62}$$

where now $\mathcal{D}_u = (\mathbb{C}^2)^{\otimes u} \otimes |?\rangle_u \otimes (\mathbb{C}^2)^{\otimes q+p-u-1} \otimes (\mathbb{C}^2)^{\otimes p}$ and $0 \leq u < q+p$ represents the position of the undefined logical qudit. As before, it is straightforward to show that $\alpha = \max_{|\eta\rangle\in\mathcal{S}_{prop}} \langle\eta|\,\Pi_{\mathcal{S}_{init}}\,|\eta\rangle \geq 1 - T^{-1}$, and so by the Geometric Lemma, $H|_{\mathcal{S}_{clock(T,u)}} \succeq c(2T^3)^{-1}$.

No undefined logical qudit
In this scenario, $H|_{\mathcal{S}_{clock}} = H_{init} + H_{prop} + H_{out}$. Defining $H_1 := H_{init} + H_{out}$ and $H_2 := H_{prop}$ as before, we can find that $\gamma(H_1) \geq 1$, $\gamma(H_2) \geq c/L^2$, and possess null spaces

$$\begin{aligned}\mathcal{S}_1 = {} & |0\rangle^{\otimes q} \otimes \operatorname{span}\{|b\rangle \otimes |b\rangle \mid b \in \{0,1\}^p\} \otimes |C_0\rangle \oplus \\ & (\mathbb{C}^2)^{\otimes q} \otimes (\mathbb{C}^2)^{\otimes 2p} \otimes \operatorname{span}\{|C_1\rangle, \ldots, |C_{L-1}\rangle\} \oplus \\ & U^\dagger\left[|1\rangle_{ans} \otimes (\mathbb{C}^2)^{\otimes q+p-1}\right] \otimes (\mathbb{C}^2)^{\otimes p} \otimes |C_L\rangle\,,\end{aligned} \tag{63}$$

and

$$\mathcal{S}_2 = (\mathbb{C}^2)^{\otimes q} \otimes (\mathbb{C}^2)^{\otimes 2p} \otimes \frac{1}{\sqrt{L+1}}\sum_{t=0}^{L}|C_t\rangle\,. \tag{64}$$

For the Geometric Lemma, we then require to calculate the quantity $\alpha = \max_{|\eta\rangle\in\mathcal{S}_2} \langle\eta|\,\Pi_{\mathcal{S}_1}\,|\eta\rangle$. Following the steps of Ref. [1] or Ref. [42], and writing projector onto the null space of $\mathcal{S}_1$ as $\Pi_{\mathcal{S}_1} = \Pi_{\mathcal{S}_{1,1}} + \Pi_{\mathcal{S}_{1,2}} + \Pi_{\mathcal{S}_{1,3}}$, we can show that in our case

$$\max_{|\eta\rangle\in\mathcal{S}_2} \langle\eta|\,\Pi_{\mathcal{S}_{1,2}}\,|\eta\rangle = \frac{L-1}{L+1}, \tag{65}$$

while the contribution of the other two terms can be bounded as

$$\max_{|\eta\rangle\in\mathcal{S}_2} \langle\eta|\,\Pi_{\mathcal{S}_{1,1}} + \Pi_{\mathcal{S}_{1,3}}\,|\eta\rangle \leq \frac{1}{L+1}\left(1 + \sqrt{\max_{|\beta\rangle\in\tilde{\mathcal{S}}_{1,1}} \langle\beta|\,\Pi_{\tilde{\mathcal{S}}_{1,3}}\,|\beta\rangle}\right), \tag{66}$$

where $\tilde{S}_{1,1} := |0\rangle^{\otimes q} \otimes \operatorname{span}\{|b\rangle \otimes |b\rangle \mid b \in \{0,1\}^p\}$ and $\tilde{S}_{1,3} := U^\dagger\left[|1\rangle_{ans} \otimes (\mathbb{C}^2)^{\otimes q-1} \otimes (\mathbb{C}^2)^{\otimes p}\right] \otimes (\mathbb{C}^2)^{\otimes p}$ are the nullspaces $S_{1,1}$ and $S_{1,3}$ without the clock register. Then, as any state $|\beta\rangle \in \tilde{\mathcal{S}}_{1,1}$ can be written as $|\beta\rangle = |0\rangle^{\otimes q} \otimes \sum_{b\in\{0,1\}^p} \alpha_b\,|b\rangle \otimes |b\rangle$ with $\sum_b |\alpha_b|^2 = 1$, we have that

$$\begin{aligned}\max_{|\beta\rangle\in\tilde{\mathcal{S}}_{1,1}} \langle\beta|\,\Pi_{\tilde{\mathcal{S}}_{1,3}}\,|\beta\rangle &= \sum_{b,b'\in\{0,1\}^p} \alpha^*_{b'}\alpha_b\,\langle 0|^{\otimes q} \otimes \langle b'| \otimes \langle b'|\left[U^\dagger(|1\rangle\langle 1|_{ans} \otimes I)U \otimes I\right]|0\rangle^{\otimes q} \otimes |b\rangle \otimes |b\rangle \\ &= \sum_{b,b'\in\{0,1\}^p} \alpha^*_{b'}\alpha_b\,\langle 0|^{\otimes q} \otimes \langle b'|\left[U^\dagger(|1\rangle\langle 1|_{ans} \otimes I)U\right]|0\rangle^{\otimes q} \otimes |b\rangle \otimes \langle b'|b\rangle \\ &= \sum_{b\in\{0,1\}^p} |\alpha_b|^2\,\langle 0|^{\otimes q} \otimes \langle b|\left[U^\dagger(|1\rangle\langle 1|_{ans} \otimes I)U\right]|0\rangle^{\otimes q} \otimes |b\rangle \\ &\leq \epsilon \sum_{b\in\{0,1\}^p} |\alpha_b|^2 \\ &= \epsilon,\end{aligned} \tag{67}$$

where in the fourth line we made use of the fact that A is a problem in QCMA and accepts with probability less than $\epsilon = 1/3$ when x is a no-instance. In other words, we observed that $\Pr[U_n \text{ accepts } (x, b)] = \langle 0|^{\otimes q} \otimes \langle b| U^\dagger (|1\rangle\langle 1|_{ans} \otimes I) U |0\rangle^{\otimes q} \otimes |b\rangle \leq \epsilon$ for any classical witness $b \in \{0,1\}^p$. Thus,

$$\alpha = \max_{\eta \in \mathcal{S}_2} \langle \eta| \Pi_{\mathcal{S}_{1,1}} + \Pi_{\mathcal{S}_{1,2}} + \Pi_{\mathcal{S}_{1,3}} |\eta\rangle \leq \frac{L-1}{L+1} + \frac{1+\sqrt{\epsilon}}{L+1} = 1 - \frac{1+\sqrt{\epsilon}}{L+1}, \tag{68}$$

and so, plugging this in the Geometric Lemma, we obtain that $H|_{S_{clock}} \succeq 2^{-1} \min\{\gamma(H_1), \gamma(H_2)\}(1-\alpha) \succeq c(1-\sqrt{\epsilon})(2L^3)^{-1}$.

6 coRP-complete Problem

By combining elements of the previous constructions, we can generate a problem that is coRP-complete.

6.1 Construction overview

In the BQP_1 constructions of Sections 3 and 4, we made use of a quantum circuit in order to decide the satisfiability of "quantum" sub-instances. In particular, this circuit verified the satisfiability of the simultaneous Π_{prop} clauses and final Π_{out} clauses. For the latter, the circuit executed the quantum circuit $U_T \dots U_1 |0\rangle^{\otimes q}$, measured some of the qubits, and accepted or rejected depending on the measurement outcomes. Intuitively, to generate the coRP complete problem, we would like to replace the universal quantum circuit by a universal classical *reversible* circuit $R = R_T \dots R_1$ (reversibility is needed since the best potentially satisfying state is still a *quantum* history state) and introduce randomness into the instance by initializing p ancilla qubits to $|+\rangle$. Then, for these new sub-instances, we could analogously verify the Π_{out} clauses by sampling a bitstring $b \in \{0,1\}^p$, evaluating the circuit R on input $(0^q, b)$ and deciding on its satisfiability based on the final state of the bits. While this idea is close to the actual construction, it is not sufficient to decide the satisfiability of the simultaneous propagation clauses.

For the construction to work, we also incorporate elements of the QCMA problem from the previous section. Namely, we will modify the $\Pi_{init}^{|00,11\rangle}$ clause so it initializes both a witness (now referred to as *auxiliary* qudit as we remove the freedom) and a logical qudit to the maximally entangled state $|\Phi^+\rangle$. This new clause is denoted $\Pi_{init}^{|\Phi^+\rangle}$.

At a later time we will demonstrate that this idea does allow verification of the simultaneous propagation clauses. For the moment, let us show in slightly more detail that this construction does allow us to verify the satisfiability of Π_{out} clauses. Again consider the toy example of Fig. 12 where now the $\Pi_{init}^{|00,11\rangle}$ clauses are replaced by $\Pi_{init}^{|\Phi^+\rangle}$ clauses and the unitaries U_i with reversible classical gates R_i. If satisfiable, the satisfying state must be the history state

$$\begin{aligned} |\psi_{hist}\rangle &= \frac{1}{\sqrt{5}} \sum_{t=0}^{4} \left[R_t \dots R_0 |00\rangle \otimes |\Phi^+\rangle^{\otimes 2} \right] \otimes |\underbrace{d \dots d}_{t} a_t \underbrace{r \dots r}_{4-t}\rangle \\ &= \frac{1}{\sqrt{5}} \sum_{t=0}^{4} \sum_{b \in \{0,1\}^2} \frac{1}{2} |\xi_b^t\rangle \otimes |b\rangle_{Aux} \otimes |\underbrace{d \dots d}_{t} a_t \underbrace{r \dots r}_{4-t}\rangle , \end{aligned} \tag{69}$$

where in the second line we observed that $|\Phi^+\rangle^{\otimes p}$ can be written as $|\Phi^+\rangle^{\otimes p} = 2^{-\frac{p}{2}} \sum_{b \in \{0,1\}^p} |b\rangle_L \otimes |b\rangle_{Aux}$ for any $p \in \mathbb{N}$, and defined $|\xi_b^t\rangle := R_t \dots R_1 |00\rangle \otimes |b\rangle_L$. The Π_{out} clause is satisfied if at time $t = 4$, the probability that the second qubit yields outcome "1" when measured is 1. This probability is

$$\begin{aligned} \Pr(\text{outcome } 1) &= \frac{1}{4} \sum_{b,b' \in \{0,1\}^2} \langle \xi_{b'}^4| \otimes \langle b'| \Pi^{(1)} |\xi_b^4\rangle \otimes |b\rangle \\ &= \frac{1}{4} \sum_{b \in \{0,1\}^2} \langle \xi_b^4| \Pi^{(1)} |\xi_b^4\rangle , \end{aligned} \tag{70}$$

from where it is evident that if the instance is satisfiable, $\langle\xi_b^4|\,\Pi^{(1)}\,|\xi_b^4\rangle = 1$ for all $b \in \{0,1\}^2$. Hence, this shows that it is possible to verify the Π_{out} clause by sampling one of the strings b, running circuit R on input $(0^2, b)$, and measuring the state of the second qubit.

Another important consideration is that the classical reversible gate set must be chosen with care. In previous constructions, $\mathcal{G}$ was chosen for two reasons: first, so all gates in the set changed the basis states upon application, and second, so the unitary $V(\Pi_i)$, or alternatively the circuit $\mathcal{C}$ of Fig. 7, could be implemented perfectly with gates from this set. The first property played an important role in the proof of Lemma 3.8, and in this construction we still desire to make use of this lemma. However, as we will soon show, circuit $\mathcal{C}$ will no longer be necessary. To make sure that the first property is satisfied, we choose

$$\mathcal{G} = \{X, (X \otimes X \otimes X)\text{Toffoli}\}, \tag{71}$$

which is a universal gate set for reversible classical computation as the Toffoli alone is known to be universal. As a consequence of this choice, the quantum satisfiability problem we define below will have some 5-local clauses since the unitary gate in the Π_{prop} clauses may be a Toffoli. This is the best locality we can achieve as it is also well-known that any universal gate set for reversible quantum computation must include a 3-bit gate.

Now that we have presented the general idea, let us define the problem more formally.

6.2 Problem definition

The problem we claim is coRP-complete is the following:

Definition 6.1 (Classical SLCT-QSAT)**.** *The problem* Classical SLCT-QSAT *is a QCSP defined on the* 8*-dimensional Hilbert space.*

$$\mathcal{H} = \text{span}\{|0_L\rangle\,, |1_L\rangle\,, |?_L\rangle\} \oplus \text{span}\{|0_{Aux}\rangle\,, |1_{Aux}\rangle\} \oplus \text{span}\{|r_C\rangle\,, |a_C\rangle\,, |d_C\rangle\}, \tag{72}$$

which now consists of a logical, an auxiliary, and a clock subspace. Furthermore, here $\mathcal{G} = \{X, (X \otimes X \otimes X)\text{Toffoli}\}$*, and* Π_{prop} *now acts on at most* 5 *qudits. Besides the operators* O_{init}*,* O_{prop} *and* O_{out} *defined in Eqs.* (49)*–*(51)*, the problem contains a new* 3*-local operator*

$$O_{init}^{|\Phi^+\rangle} := \Pi_{L,1} + \Pi_{Aux,2} + \Pi_{C,3} + \Pi_{start,3} + |?\rangle\langle?|_1 \otimes |a\rangle\langle a|_3 + (I - |\Phi^+\rangle\langle\Phi^+|)_{1,2} \otimes |a\rangle\langle a|_3\,, \tag{73}$$

where $\Pi_{Aux} := I - (|0\rangle\langle 0|_{Aux} + |1\rangle\langle 1|_{Aux})$ *is the projector to the auxiliary subspace. As before,* $\Pi_{init}^{|\Phi^+\rangle}$ *refers to the projector with the same null space as this operator.*

The promise remains the same as before. We are promised that for every instance considered either (1) there exists a quantum *state* $|\psi_{sat}\rangle$ *on* n 8*-dimensional qudits such that* $\Pi_i|\psi_{sat}\rangle = 0$ *for all* i*, or (2)* $\Sigma_i\,\langle\psi|\Pi_i|\psi\rangle \geq 1/poly(n)$ *for all* quantum *states* $|\psi\rangle$*.*

The goal is to output "YES" if (1) is true, or output "NO" otherwise.

Before showing that Classical SLCT-QSAT is contained within coRP and is also coRP-hard, let us make an observation about the problem.

Remark 2. *While the projectors of the problem are defined with a restricted set of gates, the algorithm that we use to decide the instance may use any reversible or irreversible classical gate set. Indeed, the circuit* R *can be perfectly simulated by any another circuit employing a universal gate set with polynomial overhead. As mentioned earlier, this is not a feature present in quantum circuits.*

Thus, the claim really is that this problem is complete for coRP *and not* $\text{coRP}^{\mathcal{G}}$ *like we have in the* BQP_1 *problem.*

6.3 In coRP

With the introduction of the $\Pi_{init}^{|\Phi^+\rangle}$, we have to verify that the new clause does not break anything from the analysis of clock components and shared logical qudits of the previous construction. In the analysis of clock components of Section 4.1.1, the lemmas are stated in terms of Π_{init} and Π_{out} clauses. However, the elements of these clauses that matter are the Π_{start} and Π_{stop} projectors within, which together require that one of the clock qudits in the chain is in an active state. Since $\Pi_{init}^{|\Phi^+\rangle}$ also contains the Π_{start} projector, for these particular statements, $\Pi_{init}^{|0\rangle}$ and $\Pi_{init}^{|\Phi^+\rangle}$ clause are equivalent. Thus, there are no new cases to consider in the analysis of clock components. On the other hand, there are four new cases to consider concerning the logical qudits of an instance: two for the logical qudits of a single clock component, and two slightly different versions of these where the logical qudit is shared across sub-instances.

Lemma 6.1. *Consider a clock component that has at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. If a logical qudit is acted on by both a $\Pi_{init}^{|0\rangle}$ and a $\Pi_{init}^{|\Phi^+\rangle}$ clause, stemming from the same clock qudit, the instance is unsatisfiable.*

Proof. First, recall that Proposition 4.1 shows that if the instance is satisfiable, both clauses must enforce their constraints on the logical qudit. Satisfying the $\Pi_{init}^{|\Phi^+\rangle}$ clause requires the logical and witness qudits to be in the state $|\Phi^+\rangle$. It is easy to see that $(I - |0\rangle\langle 0|)_1 |\Phi^+\rangle \neq 0$ and so the Π_{init} clause cannot be satisfied. □

Lemma 6.2. *Consider two clock components, both of which have at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. If one clock component acts on a logical qudit with a $\Pi_{init}^{|0\rangle}$ clause, and the other component also acts on this qudit with a $\Pi_{init}^{|\Phi^+\rangle}$ clause, the instance is unsatisfiable.*

Proof. Just like in the previous proof, Proposition 4.1 shows that if the instance is satisfiable, both clauses must enforce their constraints on the logical qudit. Now, recall that if the qudit is also present in a Π_{prop} clause from any either clock component, the instance is unsatisfiable by Lemma 3.8. Therefore, the qudit must then remain fixed to one state. Again, the qudit cannot be both $|0\rangle$ and part of a maximally entangled state $|\Phi^+\rangle$, and so the instance is unsatisfiable. □

One might think that there is a third scenario with a similar statement, which is when the $\Pi_{init}^{|0\rangle}$ and $\Pi_{init}^{|\Phi^+\rangle}$ clause stem from different clock qudits of the same clock component (which may occur given the strangely shaped sub-instances like those of Fig. 9). For this case, we simple observe that by the analysis of Section 4.1.1, if the sub-instance is satisfiable, these two clock qudits can be thought of as forming part of two smaller disjoint clock components. This is the scenario of Lemma 6.2.

The other two new cases are:

Lemma 6.3. *Consider a clock component that has at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. If two $\Pi_{init}^{|\Phi^+\rangle}$ clauses act on the same clock and auxiliary qudits, but act on distinct logical qudits, the instance is unsatisfiable. Similarly, if two $\Pi_{init}^{|\Phi^+\rangle}$ clauses act on the same clock and logical qudits, but act on distinct auxiliary qudits, the instance is unsatisfiable.*

Proof. This follows from monogamy of entanglement. First, observe that since the $\Pi_{init}^{|\Phi^+\rangle}$ clauses must enforce their constraints on the auxiliary and logical qudits (Proposition 4.1), the logical qudits cannot be $|?\rangle$ due to the $|?\rangle\langle ?| \otimes |a\rangle\langle a|$ projector. Hence, the logical qudits become qubits. Then, in the former scenario, the auxiliary qubit is required to be maximally entangled with two logical qudits. In the latter scenario, it is the logical qubit that is required to be maximally entangled with two auxiliary qubits. Both violate monogamy of entanglement and so the instance is unsatisfiable. □

Lemma 6.4. *Consider two clock components, both of which have at least one $\Pi_{init}^{|\cdot\rangle}$ and one Π_{out} clause. If one clock component acts on a logical qudit with a $\Pi_{init}^{|\Phi^+\rangle}$ clause, and the other component also acts on it with a $\Pi_{init}^{|\Phi^+\rangle}$ clause, the instance is unsatisfiable. Similarly, if this is also the case for an auxiliary qudit, the instance is unsatisfiable.*

Proof. The proof of the lemma is the same as for Lemma 6.3 if we note that due to Lemma 3.8, these qudits must always be fixed to a single state. □

Now, let us address the simultaneous propagation clauses. In Proposition 3.3, we showed that k simultaneous Π_{prop} clauses were satisfiable iff $U_{t+1,i} |\phi_t\rangle = U_{t+1,j} |\phi_t\rangle \quad \forall i, j \in [k]$, where $|\phi_t\rangle = U_t \dots U_1 |\phi_0\rangle$ and $|\phi_0\rangle$ was the initial state of the computation. In this construction we can extend this result by making use of the new clauses of the instance and the reversible nature of the gate set.

Lemma 6.5. *Suppose k distinct Π_{prop} clauses act on clock qudits c_t and c_{t+1} with $0 \le t < T$, and let $R_{t+1,1}, \ldots, R_{t+1,k}$ denote the reversible gates associated with these k clauses. The clauses are satisfiable iff $R_{t+1,i} |\xi_b^t\rangle = R_{t+1,j} |\xi_b^t\rangle$ for all $i, j \in [k]$ and for all $b \in \{0,1\}^p$.*

Proof. Since the classical reversible gates are unitaries, Proposition 3.3 still holds and these simultaneous propagation clauses are satisfiable iff $R_{t+1,i} |\phi_t\rangle = R_{t+1,j} |\phi_t\rangle$ for all $i, j \in [k]$. For an initial state with p $|\Phi^+\rangle$ states, we can write the intermediate state of the computation as $|\phi_t\rangle = 2^{-\frac{p}{2}} \sum_{b\in\{0,1\}^p} |\xi_b^t\rangle \otimes |b\rangle$, and so the clauses are satisfiable iff

$$\begin{aligned} 1 &= \langle\phi_t| R_{t+1,j}^\dagger R_{t+1,i} |\phi_t\rangle \\ &= \frac{1}{2^p} \sum_{b,b'\in\{0,1\}^p} \langle\xi_b^t| \otimes \langle b'| R_{t+1,j}^\dagger R_{t+1,i} |\xi_b^t\rangle \otimes |b\rangle \\ &= \frac{1}{2^p} \sum_{b\in\{0,1\}^p} \langle\xi_b^t| R_{t+1,j}^\dagger R_{t+1,i} |\xi_b^t\rangle . \end{aligned} \tag{74}$$

Then, it is evident that for the right hand-side to be equal to 1, $R_{t+1,i} |\xi_b^t\rangle = R_{t+1,j} |\xi_b^t\rangle$ for all basis states $b \in \{0,1\}^p$. □

In the algorithm of previous constructions, we had to evaluate the satisfiability of the simultaneous propagation clauses by executing a quantum circuit for every pair of clauses. Now, this new lemma will allow us to decide these constraints classically. This will be explained in detail at a later time.

The discussion for active dots remains the same as before.

Finally, let us make a small observation. In Remark 1, we mentioned two "quantum" sub-instances that were actually trivially decidable but were intentionally left for the quantum algorithm to decide. Because of the $\Pi_{init}^{|\Phi^+\rangle}$ clause, we now have an extra two such sub-instances. These are sub-instances with no undefined or simultaneous propagation clauses, where the logical qudits present in Π_{out} clauses are (1) initialized to one half of a Bell pair $|\Phi^+\rangle$ but not acted by any Π_{prop} clause, or (2) not initialized. In the first case, the instance is trivially unsatisfiable since its reduced density matrix $\rho = 1/2(|0\rangle\langle 0| + |1\rangle\langle 1|)$ has support on $|0\rangle$. As before, the second case is trivially satisfiable. All four sub-instances will be decided in the randomized classical algorithm explained below.

6.3.1 Algorithm

The first step of the algorithm remains the same as in Section 4.2.1. The second step requires a small change to indicate that either type of initialization clause is allowed.

(2) *Ignore the trivial sub-instances.* For every sub-instance, check whether it contains both $\Pi_{init}^{|\cdot\rangle}$ and Π_{out} clauses. If it does not, ignore all clauses of the sub-instance for the rest of the algorithm. Correctness is given by Lemma 4.2.

We now add two new steps between (2) and (3). One to account for Lemma 6.1 and Lemma 6.2, and the second to account for Lemma 6.3 and Lemma 6.4.

(coRPa) For every logical qudit in the instance, check whether it is acted on by both a $\Pi_{init}^{|0\rangle}$ and a $\Pi_{init}^{|\Phi^+\rangle}$ clause. If such a qudit exists, reject. Correctness is given by Lemma 6.1 and Lemma 6.2.

(coRPb) For every logical qudit check whether it is connected to several auxiliary qudits. If it is, reject. Similarly, for every auxiliary qudit check whether it is connect to several logical qudits. If it is, reject. Correctness is given by Lemma 6.3.

These steps must take place in this specific point in the algorithm—after processing the trivial sub-instances, but before assigning the state of the logical qudits, which occurs in the following step.[31] Afterwards, step (3) requires a small modification to incorporate the $\Pi_{init}^{|\Phi^+\rangle}$ clause:

[31] Steps (coRPa) and (coRPb) also processes cases related to the shared logical qudits of distinct clock components, which before would only be processed in step (4). Evaluating these sooner does not have any undesired effect.

(3) *Identify the undefined Π_{prop} clauses.* For every logical qudit, check whether it is acted on by at least one $\Pi_{init}^{|0\rangle}$ or $\Pi_{init}^{|\Phi^+\rangle}$ clause. If it is not, mark all Π_{prop} clauses this qudit is part of as undefined.

Subsequently, steps (4), (5), and (7) are the same as before. In step (6), we replace the quantum algorithm with a classical subroutine acting on ancilla and randomly sampled bits. Let us explain this step in more detail.

Classical subroutine

As before, the inputs to this classical subroutine are the sub-instances of Fig. 6, except now the unitaries are reversible classical gates. We let p be the number of logical qudits present in the $\Pi_{init}^{|\Phi^+\rangle}$ clauses of a sub-instance and q the number of logical qudits present in $\Pi_{init}^{|0\rangle}$ clauses. Additionally, let T denote the length of the sub-instance and $k_{t,t+1}$ denote the number of simultaneous Π_{prop} clauses acting on clock qudits c_t and c_{t+1}.

Ultimately, the goal of the quantum algorithm in previous constructions was to evaluate whether the simultaneous Π_{prop} clauses and the final Π_{out} clauses could be satisfied, as all other clauses could be determined to be satisfiable. To do so, we discussed an approach based on Bravyi's ideas for k-QSAT, which relied on generating the history state and measuring the eigenvalues of all projectors. This method nicely related to the promise of the problem which allowed us to demonstrate the BQP_1 and QCMA soundness of the algorithm. However, we mentioned that the resulting algorithm was difficult to implement and wasteful. Instead, we calculated the acceptance/rejection probabilities of measuring the eigenvalue of a clause, and developed a more efficient method that reproduced the same statistics. To decide two simultaneous propagation clauses, we applied circuit $\mathcal{C}$ (more accurately $\mathcal{C}_{\mathcal{G}}$) shown in Fig. 7, and to decide Π_{out} clauses, we executed the circuit and at the end of the computation measured the qubits present in these clauses. In this construction, we must rethink the algorithm since we cannot perform Bravyi's method with a classical computer as it requires creating the highly-entangled history state, and we cannot the repeat the same algorithm as before because we cannot execute circuit $\mathcal{C}$. What we can do is to design a classical algorithm that yields the same probability as measuring the eigenvalue of a simultaneous Π_{prop} clause.

Claim 6.1. *Let $\Pi_{prop}^{(c_t,c_{t+1})}$ be a clause acting on clock qudits c_t, c_{t+1} with associated reversible classical gate $R_{t+1,j}$, where $j \in [k_{t,t+1}]$. Measuring the eigenvalue of projector $\Pi_{prop}^{(c_t,c_{t+1})}$ in the state $|\psi_{phist}\rangle$ is equivalent to the following classical steps:*

1. *Create the basis state $|\xi_b^t\rangle = R_{t,1} \dots R_{1,1} |0\rangle^{\otimes q} \otimes |b\rangle$, where $b \in \{0,1\}^p$ is a bitstring sampled uniformly at random.*

2. *Flip a coin and compare $R_{t+1,1} |\xi_b^t\rangle$ with $R_{t+1,j} |\xi_b^t\rangle$. Reject if the coin is "Heads" and $R_{t+1,1} |\xi_b^t\rangle \neq R_{t+1,j} |\xi_b^t\rangle$.*

Proof. In the proof of Claim 3.1, we showed that the probability of measuring eigenvalue 1 for this propagation clause was given by $\Pr(\text{outcome } 1) = 2^{-1}(1 - \langle\phi_t| U_{t+1,j}^\dagger U_{t+1,1} |\phi_t\rangle)$. Then, by a similar calculation to that in the proof of Lemma 6.5, in this construction, this probability becomes $\Pr(\text{outcome } 1) = 2^{-1}(1 - 2^{-p} \sum_{b\in\{0,1\}^p} \langle\xi_b^t| R_{t+1,j}^\dagger R_{t+1,1} |\xi_b^t\rangle)$.

To show that the classical algorithm above rejects the instance with this probability, first observe that $2^{-p} \sum_{b\in\{0,1\}^p} \langle\xi_b^t| R_{t+1,j}^\dagger R_{t+1,1} |\xi_b^t\rangle$ is the fraction of bitstrings b for which the basis state $|\xi_b^t\rangle$ satisfies $R_{t+1,1} |\xi_b^t\rangle = R_{t+1,j} |\xi_b^t\rangle$. In other words, this is the probability that in the algorithm we sample a bitstring b for which $R_{t+1,1} |\xi_b^t\rangle = R_{t+1,j} |\xi_b^t\rangle$. Denote this event as A. Then, the probability the algorithm rejects, i.e. observes both "Heads" and A^C, is $\Pr(\text{Heads} \cap A^C) = \Pr(\text{Heads}) \Pr(A^C) = 2^{-1}(1 - 2^{-p} \sum_{b\in\{0,1\}^p} \langle\xi_b^t| R_{t+1,j}^\dagger R_{t+1,1} |\xi_b^t\rangle)$.[32] Thus, both methods yield the same statistics and are equivalent. □

Making use of this claim, the subroutine is then the following:

(6) For every sub-instance, do the following:

[32] A^C denotes the event where A does not occur.

(6.1) Create the initial state $|\xi_b^0\rangle$ by sampling p bits at random and initializing another q bits to "0".

(6.2) For every $t \in [T]$, do the following:

(6.2.1) If $k_{t,t+1} \geq 1$:

- Choose an index $1 < j < k_{t,t+1}$. Create another copy of the current state of the computation to obtain two copies of $|\xi_b^t\rangle$. Apply $R_{t+1,1}$ to the first, and $R_{t+1,j}$ to the second.
- Remove the Π_{prop} clause with unitary $R_{t+1,j}$ from the instance.
- Flip a coin. If $R_{t+1,1}\,|\xi_b^t\rangle \neq R_{t+1,j}\,|\xi_b^t\rangle$ and the coin is "Heads", reject. Otherwise, go back to step (6.1).

(6.2.2) Otherwise, apply $R_{t+1,1}$ to the state to obtain $|\xi_b^{t+1}\rangle$.

(6.3) *At this step, the algorithm has produced the basis state* $|\xi_b^T\rangle$. For every logical qudit in the instance with a Π_{out} clause, check if any of the corresponding logical bits are "0". If there is a such bit, reject.

We note that although in this construction there is no need to restart the algorithm after checking whether two simultaneous propagation clauses can be simultaneously satisfied, we choose to do so in order to maintain some resemblance with the algorithms of the previous quantum satisfiability problems.

6.3.2 Runtime

The small changes in steps (2) and (3) and the new steps (coRPa), (coRPb), and (6) do not affect the algorithm's runtime significantly and so it still runs in polynomial time. In particular, in steps (coRPa) and (coRPb), the algorithm iterates over all $\mathcal{O}(n)$ logical qudits of the instance, and for each qudit, checks whether it is acted on by a $\Pi_{init}^{|0\rangle}$ or $\Pi_{init}^{|\Phi^+\rangle}$ clause. These steps take $\mathcal{O}(poly(n))$ time. For step (6.1), observe that since $p \in \mathcal{O}(n)$, this step takes $\mathcal{O}(n)$ time. Step (6.2) is similar to step (6.2) of the first construction (see Section 3.4), however, instead of executing circuit C to evaluate the satisfiability of the simultaneous propagation clauses, here we create two copies of the state, apply a reversible gate, and compare each pair of bits. These steps clearly also take $\mathcal{O}(n)$ time, and given that there are at most $\mathcal{O}(poly(n))$ simultaneous propagation clauses, step (6.2) takes $\mathcal{O}(poly(n))$ time for the whole sub-instance. In step (6.3), the algorithm iterates over a subset of the logical qudits in the sub-instance and checks whether they are "0". This step takes $\mathcal{O}(n)$ time. Overall, the algorithm takes $\mathcal{O}(poly(n))$ time to decide each sub-instance, and given that there are at most $\mathcal{O}(poly(n))$ sub-instances, step (6) also takes $\mathcal{O}(poly(n))$ time.

6.3.3 Completeness and soundness

Similar to the BQP_1 construction of Section 3, sub-instances that are trivially satisfiable and those that have truncated active clock chains can be verified classically and deterministically. Thus, we are interested in the instances that make use of the subroutine outlined in step (6). As shown in Fig. 6, there are three types of such instances. Part of our task is to demonstrate that if a quantum satisfying state exists, we can reach this conclusion classically.

In Section 3 we showed that the algorithm's acceptance/rejection probabilities (and therefore completeness and soundness) depended only on the outcomes of measuring the eigenvalues of the simultaneous Π_{prop} and final Π_{out} clauses of each sub-instance. In Claim 6.1 we showed that measuring the eigenvalues of the simultaneous propagation clauses could be done with a coRP circuit, and the same can be shown for the Π_{out} clauses by a straightforward generalization of the analysis presented at the beginning of the section for longer chains.[33] Therefore, we can reach the same conclusions as in that section: if x is a yes-instance, the algorithm's acceptance probability is $AP = 1$, while in the case that x is a no-instance, $AP \leq 1 - 1/poly(n)$ which can then be improved to $\leq 1/3$ by repeating the algorithm several times and taking the majority vote.

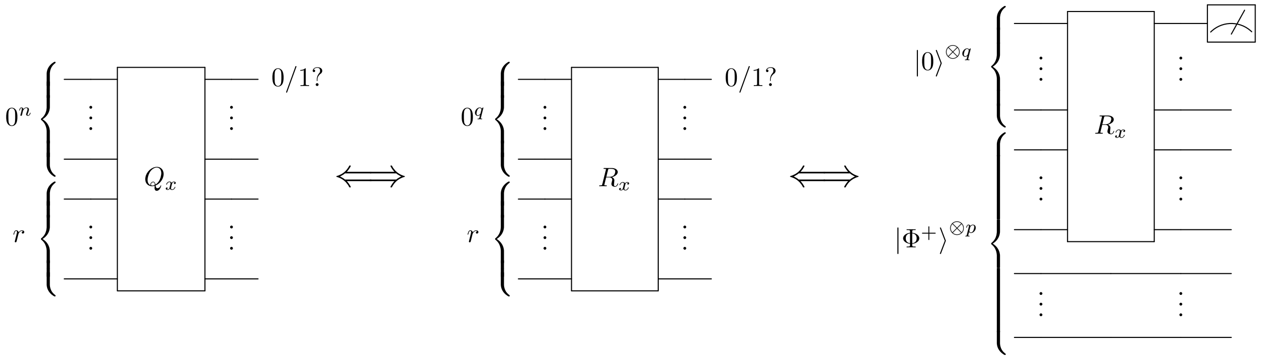


Figure 13: Circuits demonstrating the equivalence between coRP and coRP_q. Left: A coRP circuit that decides an instance x of an arbitrary promise problem. Q_x is a deterministic classical circuit acting on n ancilla bits and a random bitstring $r \sim \text{Unif}(\{0,1\}^p)$ of size $p(n)$. Middle: A coRP circuit where Q_x is replaced by a classical reversible circuit R_x now acting on $q(n) = n + poly(n)$ ancilla bits. Right: a coRP_q quantum circuit where R_x now acts on $q(n)$ ancilla qubits and on the first subsystem of $p(n)$ maximally entangled states. Since the second half of these states are not acted on by R_x, these qudits can be thought of as measured at the beginning of the computation. The measurement outcomes are random, thus obtaining a random bitstring r. Consequently, due to the maximal entanglement, the first half of the qubits must also be r, and the input to R_x is then also $|0\rangle^{\otimes q} \otimes |r\rangle$, concluding that $\mathsf{coRP} = \mathsf{coRP}_q$.

6.4 coRP-hard

To show that Classical SLCT-QSAT is coRP-hard and complete the proof of Theorem 4, we use similar ideas to Bravyi *et al.* [41] when showing that the *stoquastic* LH problem is MA-hard. They show that to encode the evaluation of a classical circuit into a quantum instance, it is useful to first represent the classical circuit as a quantum circuit. To this end, they define MA_q—the quantum version of this class composed purely of reversible classical gates and ancilla qudits $|0\rangle$ and $|+\rangle$—and show that $\mathsf{MA}_q = \mathsf{MA}$. Similarly, we define the complexity class coRP_q and show that it is equal to coRP. However, instead of using $|+\rangle$ states, we use maximally entangled states $|\Phi^+\rangle$ in preparation for the reduction to a Classical SLCT-QSAT instance.

Definition 6.2 (coRP_q)**.** *A promise problem $A = (A_{yes}, A_{no})$ is in coRP_q iff there exist polynomials p, q and a uniform family of polynomial-time classical reversible circuits $\{R_n\}$, that take as input a binary string x with $|x| = n$, $q(n)$ ancilla qubits in the state $|0\rangle$, and $p(n)$ Bell states $|\Phi^+\rangle$, such that each R_n acts only on one of the halves of each Bell state, and:*

- *(Completeness) If $x \in A_{yes}$, then* $\Pr[R_n \text{ accepts } x] = 1$.
- *(Soundness) If $x \in A_{no}$, then* $\Pr[R_n \text{ accepts } x] \leq 1/3$.

Lemma 6.6. $\mathsf{coRP}_q = \mathsf{coRP}$.

Proof. $\mathsf{coRP} \subseteq \mathsf{coRP}_q$: Let A be an arbitrary (promise) problem in coRP. In other words, there exists a probabilistic classical algorithm $Q_x = Q_L \dots Q_0$ with $L = poly(n)$ using $p(n)$ random bits that decides an instance x of A with perfect completeness. As mentioned earlier, any coRP circuit can be perfectly and efficiently simulated by any other circuit using a universal gate set. Hence, there exists a reversible circuit $R_x = R_{L'} \dots R_0$ with $L' = poly(L) = poly(n)$, acting on $p(n)$ random bits and an additional $poly(n)$ ancilla bits that is equivalent to Q_x. Moreover, we can take R_x to be composed of Toffoli and X gates since this set is both universal and reversible.[34] We let $q(n) = n + poly(n)$ denote the total number of ancilla bits in the

[33] In contrast to the previous QCMA problem where there were more differences between the toy instance and arbitrary sub-instances, here the sub-instances maintain the same structure: a logical qudit acted on by a $\Pi_{init}^{|\Phi^+\rangle}$ clause cannot be acted by another such clause or a $\Pi_{init}^{|0\rangle}$ clause.

[34] The X gates are necessary because of our choice to have all ancillas initialized to bit 0. Otherwise, the Toffoli gates would only act trivially.

circuit. The coRP_q quantum circuit that decides A is simply R_x, except each input random bit is replaced with a two-qubit Bell state $|\Phi^+\rangle$, and each ancilla bit is replaced by an ancilla qubit. These circuits are illustrated in Fig. 13.

To demonstrate the coRP_q circuit meets the completeness and soundness conditions, we again observe that it has the same structure as in previous subsections and the evolution of the input state can be broken down into the evolution of independent bitstrings. The probability the circuit accepts is given by

$$\begin{aligned} p_{acc} &= 2^{-p} \sum_{b,b' \in \{0,1\}^p} \langle b'| \otimes \langle \xi_b^L | \Pi^{(1)} |\xi_b^L\rangle \otimes |b\rangle \\ &= 2^{-p} \sum_{b \in \{0,1\}^p} \langle \xi_b^L | \Pi^{(1)} |\xi_b^L\rangle \\ &= 2^{-p} \sum_{b \in \{0,1\}^p} p_{acc,b}, \end{aligned} \tag{75}$$

where $|\xi_b^L\rangle$ is the state at the end of the computation for bitstring b and $p_{acc,b} := \langle \xi_b^L | \Pi^{(1)} |\xi_b^L\rangle = \langle b| \otimes \langle 0|^{\otimes q} R_1^\dagger \dots R_L^\dagger \Pi^{(1)} R_L \dots R_1 |0\rangle^{\otimes q} \otimes |b\rangle$ is the probability the circuit accepts (x, b). From this equation it becomes apparent that p_{acc} is the fraction of bitstrings b that accept. Then, observe that if x is a yes-instance, the classical circuit accepts the instance with certainty for any random bitstring $b \in \{0,1\}^p$, i.e. $p_{acc,b} = 1$ for all b. Hence, $p_{acc} = 1$ and the quantum circuit also accepts with certainty. On the other hand, if x is a no-instance, we have that in the original circuit at most a third of the bitstrings lead to wrongful acceptance, implying that $p_{acc} \leq 2^{-p}(2^p/3) = 1/3$.

$\mathsf{coRP}_q \subseteq \mathsf{coRP}$: Suppose A is a problem in coRP_q, and let the rightmost circuit in Fig. 13 be the coRP_q circuit decides an instance x of A. The classical circuit that also decides this instance is the middle circuit of Fig. 13, where r is a bitstring sampled uniformly at random. To prove that it satisfies the completeness and soundness conditions, consider the acceptance probability of the original coRP_q circuit given by Eq. (75). If x is a yes-instance, this time we have that $p_{acc} = 1$. The r.h.s. of the equation can only be 1 if $p_{acc,b} = 1$ for all b. Since this is the probability that the classical algorithm accepts with input bitstring b, the classical algoritm also accepts with certainty. On the other hand, if x is a no-instance, $p_{acc} = 2^{-p} \sum_b p_{acc,b} \leq 1/3$. Since this is the fraction of bitstrings that accept, the classical algorithm—which samples one of these random bitstrings—also accepts with probability $\leq 1/3$. □

Now, we can more comfortably show how to encode a coRP circuit into an instance of our problem. Let $R = R_L \dots R_1$ be the coRP_q circuit that given an instance x of problem A, decides whether $x \in A_{yes}$ or $x \in A_{no}$. The input to the circuit are two ancilla registers $|0\rangle^{\otimes q}$ and $|\Phi^+\rangle^{\otimes p}$ and let ans be the ancilla qubit that provides the decision. The reduction $x \mapsto x'$ then requires $q + p + (L+1)$ qudits, each of dimension 6, and is the following:

1. Choose $q + p$ qudits to serve as the logical qudits of the computation, and $L + 1$ to function as the clock qudits. Define an ordering for each group of qudits.

2. For all $i \in [q]$, create a $\Pi_{init}^{|0\rangle}$ clause acting on c_0 and l_i.

3. For all $j \in [p]$, create a $\Pi_{init}^{|\Phi^+\rangle}$ clause acting on c_0 and l_{q-1+j}.

4. For all $k \in [L]$, create a Π_{prop} clause with associated unitary R_{k+1} where the two clock qudits of the clause are c_k and c_{k+1} (with c_k being the predecessor of c_{k+1}) and the two logical qudits are those that the gate originally acts on.

5. Create one Π_{out} clause acting on c_L and l_{ans}.

The resulting instance x' consists of a one-dimensional clock chain of length L ($L + 1$ clock qudits) with a unique direction, no simultaneous propagation clauses, and all logical qudits initialized to either $|0\rangle$ or $|\Phi^+\rangle$.

6.4.1 Completeness and soundness

In the case that $x \in A_{yes}$, x' is satisfied by the history state

$$|\psi_{hist}\rangle = \frac{1}{\sqrt{L+1}} \sum_{t=0}^{L} \left[R_t \dots R_0 \, |0\rangle^{\otimes q} \, |\Phi^+\rangle^{\otimes p} \right] \otimes |\underbrace{d \dots d}_{t} a_t \underbrace{r \dots r}_{L-t}\rangle \,, \tag{76}$$

and so completeness is preserved.

When $x \in A_{no}$, the only nontrivial change in Eq. (52) is that H_{init} now collects both the $(I - |0\rangle\langle 0|) \otimes |a\rangle\langle a|$ terms (q of them) and the $(I - |\Phi^+\rangle\langle\Phi^+|) \otimes |a\rangle\langle a|$ terms (p of them). Again, we must only show that $H|_{\mathcal{S}_{init}}$ has no small eigenvalues in both scenarios where one of the logical qudits in the TACC is in the undefined state, and when there's no such logical qudit. Given that we have shown several similar arguments before, we only provide the key points.

At least one undefined logical qudit
Here, $H|_{\mathcal{S}_{init}} = H_{init} + H_{prop}(T, u)$. While H_{init} is no longer diagonal, the $(I - |0\rangle\langle 0|) \otimes |a\rangle\langle a|$ terms commute with the $(I - |\Phi^+\rangle\langle\Phi^+|) \otimes |a\rangle\langle a|$ terms. Analyzing each set of terms separately, we can observe that $\gamma(H_{init}) \geq 1$. The rest follows in the same way as in the corresponding section of the QCMA problem.

No undefined logical qudit
As before, $H|_{\mathcal{S}_{clock}} = H_1 + H_2$ where $H_1 = H_{init} + H_{prop}$ and $H_2 = H_{out}$. In this construction, $\gamma(H_1) \geq 1$, $\gamma(H_2) \geq c/T^2$, and their null spaces are given by

$$\begin{aligned} \mathcal{S}_1 = {} & |0\rangle^{\otimes q} \otimes \left(2^{-p/2} \sum_{b \in \{0,1\}^p} |b\rangle \otimes |b\rangle \right) \otimes |C_0\rangle \oplus \\ & (\mathbb{C}^2)^{\otimes q} \otimes (\mathbb{C}^2)^{\otimes 2p} \otimes \mathrm{span}\{|C_1\rangle, \dots, |C_{L-1}\rangle\} \oplus \\ & U^\dagger [|1\rangle_{ans} \otimes (\mathbb{C}^2)^{\otimes q+p-1}] \otimes (\mathbb{C}^2)^{\otimes p} \otimes |C_L\rangle \,, \end{aligned} \tag{77}$$

and

$$\mathcal{S}_2 = (\mathbb{C}^2)^{\otimes q} \otimes (\mathbb{C}^2)^{\otimes 2p} \otimes \frac{1}{\sqrt{L+1}} \sum_{t=0}^{L} |C_t\rangle \,. \tag{78}$$

We can again show that $\max_{|\eta\rangle \in \mathcal{S}_2} \langle\eta| \Pi_{\mathcal{S}_{1,2}} |\eta\rangle = L - 1/L + 1$, but now

$$\max_{|\beta\rangle \in \tilde{\mathcal{S}}_{1,1}} \langle\beta| \Pi_{\tilde{\mathcal{S}}_{1,3}} |\beta\rangle = \frac{1}{2^p} \langle 0|^{\otimes q} \otimes \langle b| \, [U^\dagger (|1\rangle\langle 1|_{ans} \otimes I) U] \, |0\rangle^{\otimes q} \otimes |b\rangle \,. \tag{79}$$

This latter quantity represents the fraction of bitstrings that lead to acceptance, and so by the soundness property of the coRP problem A, $\max_{|\beta\rangle \in \tilde{\mathcal{S}}_{1,1}} \langle\beta| \Pi_{\tilde{\mathcal{S}}_{1,3}} |\beta\rangle \leq \epsilon$ with $\epsilon = 1/3$. Plugging this in the Geometric Lemma, we again obtain that $H|_{\mathcal{S}_{clock}} \succeq c(1 - \sqrt{\epsilon})/(2L)^{-3}$, concluding that H has no small eigenvalues.

7 Universality of qubits for QCSPs

In the previous sections, we showed there are three QSAT problems acting on qudits that are complete for some complexity classes. Here, we refine these statements and show that there are QSAT problems on qubits (albeit with higher locality) that are also complete for these classes. To achieve this, we show that any QCSP on qudits can be reduced to another QSAT problem on qubits using little computational power. We note that this section, apart from some changes in the exposition, stems directly from Meiburg's previous iteration of the paper.

At first glance this statement may seem trivial as operations on qubits are universal for quantum computation, i.e. we can emulate a d-qudit with a $\lceil \log_2(d) \rceil$ qubits and carry out unitaries on those qubits. However, there is a nuance around reductions on QCSPs that has not come up before, as previous QCSPs were defined on qubits. To see this, consider one of our QSAT problems, say SLCT-QSAT. It is evident that the instance that appears in the hardness section can be mapped to one using qubits, demonstrating that there is a qubit QCSP that is hard for BQP_1. However, it is not clear if any input instance generated with these qubit clauses is contained within this class. As a simple example, let us represent the basis clock states as qubits, $|r\rangle = |00\rangle$, $|a\rangle = |01\rangle$, and $|d\rangle = |10\rangle$, and so the $\Pi_{start} = |r\rangle\langle r|$ clause within H_{init} can now be written as $\Pi_{start} = |00\rangle\langle 00| + |11\rangle\langle 11|$ (the last term is to prevent the fourth basis state that did not exist before). The clause $|0_1 0_2\rangle\langle 0_1 0_2| + |1_1 1_2\rangle\langle 1_1 1_2| + |0_2 0_3\rangle\langle 0_2 0_3| + |1_2 1_3\rangle\langle 1_2 1_3|$ acting on three qubits is now valid. The clauses can be satisfied by the state $|0_1 0_2 1_3\rangle$, but now there is some ambiguity on the state that they represent: either we have $|r\rangle$ on qubits 1 and 2, or $|a\rangle$ on qubits 2 and 3. In general, doing this for all of our clauses and considering all input instances that may occur adds a significant level of complexity to the problem, making it difficult to determine if it remains in BQP_1. Moreover, we remark that this is not only the case in our QSAT problems, but applies to all QCSPs and classical CSPs defined on qudits or non-Boolean variables! In general, the issue is that we cannot ensure that the new qubit clauses are applied to qubits in a consistent fashion based on its parent qudit problem: one qubit clause might treat a particular qubit as "qubit 1" of a previous d-qudit, while another clause might refer to the same qubit as "qubit 2". Moreover, the qubit clauses could also "mix and match", combining "qubit 1" from one previous d-qudit with "qubit 2" from another d-qudit. Overall, these lead to constraints that were unrealizable in the parent qudit problem.

Here, we show that with a different reduction, we can guarantee that the qubit problem remains within the complexity class. This is not something that is not known to be possible classically! In the quantum world, we can fix this, again by using monogamy to bind together our constituent qubits into ordered, entangled larger systems. Each clause in the resulting problem is given a projector that forces this particular ordering of qubits, and any two clauses that try use the same qubits in multiple ways are frustrated. Formally,

Theorem 6. *For any QCSP $\mathcal{C}$ on d-qudits, there is another QCSP $\mathcal{C}'$ on qubits, and AC^0 circuits f and g, such that f reduces $\mathcal{C}$ to $\mathcal{C}'$, and g reduces $\mathcal{C}'$ to $\mathcal{C}$. If $\mathcal{C}$ is k-local, then $\mathcal{C}'$ can be chosen to be $4{\cdot}2^{\lceil \log_2(\lceil \log_2(d)\rceil)\rceil}k$ local (that is, $O(\log(d))$ times larger.)*

Proof. First, we will show that for any d-qudits, we can reduce to 4-qudits; after that we will reduce to qubits. Finally, we show that the reduction is in AC^0.

We will view a 4-qudit as the product of a "data" qubit and an "entanglement" qubit. A d-qudit will be replaced by $n = \lceil \log_2(d)\rceil$ many 4-qudits, and the state of the d-qudit will be encoded in the product of all the data qubits. If $d < 2^{\lceil \log_2(d)\rceil} = 2^n$, that is, if d is not exactly a power of 2, we will have a Hamiltonian term T_1 in our clauses to ensure that the last $2^n - d$ states are not used. Acting on entanglement subspaces of the 4-qudits, consider a term T_2 whose nullspace consists of just the vector

$$\left(1 \otimes X^{\theta} \otimes X^{2\theta} \otimes \dots X^{n\theta}\right) \frac{|0\rangle^n + |1\rangle^n}{\sqrt{2}}$$

where $\theta = \frac{1}{2(n+1)}$. This is a kind of GHZ state, which uses a slightly different pair of basis states (instead of just $|0\rangle$ and $|1\rangle$) on each separate qubit. Any bipartition of this state is impure, but since T_2 has a one-dimensional nullspace, it cannot be satisfied by any mixed state. Thus the sum of two T_2 on any two overlapping but distinct sets of 4-qudits will be frustrated. If two copies of T_2 act on the same 4-qudits in a different order, they will apply the wrong angles $X^{k\theta}$ at those sites, and the ground states will not align—also leading to frustration.

Each clause H of $\mathcal{C}$ is mapped to a new clause H' that acts as H on the data subspaces of each set of 4-qudits; that has $T_1 = 1 - \sum_i^d |i\rangle\langle i|$ on each clumping of 4-qudits, to ensure that only the first d states are used; and T_2 on each clumping of 4-qudits, to ensure that they will only be clustered to each other and in a particular order.

Then, we want to reduce this from 4-qudits to qubits. Consider the following Hamiltonian on 4 qubits:

$$H_{4\to 2} = 1 - |\psi_1\rangle\langle\psi_1| - |\psi_2\rangle\langle\psi_2| - |\psi_3\rangle\langle\psi_3| - |\psi_4\rangle\langle\psi_4|$$

$$|\psi_1\rangle = \frac{1}{2}\left(\frac{3}{5}|0000\rangle - \frac{4}{5}|0001\rangle + |0100\rangle + |1010\rangle + \frac{8}{17}|1100\rangle + \frac{15}{17}|1111\rangle\right)$$

$$|\psi_2\rangle = \frac{1}{2}\left(\frac{4}{5}|0000\rangle + \frac{3}{5}|0001\rangle - |0110\rangle + |1001\rangle + \frac{20}{29}|1101\rangle + \frac{21}{29}|1110\rangle\right)$$
$$|\psi_3\rangle = \frac{1}{2}\left(\frac{5}{13}|0010\rangle + \frac{12}{13}|0011\rangle - |0111\rangle + |1000\rangle - \frac{21}{29}|1101\rangle + \frac{20}{29}|1110\rangle\right)$$
$$|\psi_4\rangle = \frac{1}{2}\left(\frac{-12}{13}|0010\rangle + \frac{5}{13}|0011\rangle - |0101\rangle + |1011\rangle - \frac{15}{17}|1100\rangle + \frac{8}{17}|1111\rangle\right).$$

Each $|\psi_i\rangle$ is orthonormal, so $H_{4\to 2}$ has a null space of dimension four. By inspecting the 840 distinct ways to apply two copies of $H_{4\to 2}$ to seven qubits, it can be checked that each sum will have a ground state above zero, except for the case where they are applied in the same way. This is a kind of "uniqueness" property that could be very loosely interpreted as monogamy for whole subspaces, instead of just one state.

By counting dimensions, one can check that this property is generic: it would hold almost always for any four random vectors. Unfortunately, for any simple and clean expressions one would write down, it would lack this property by one symmetry or another. This is why the simplest construction readily available, given above, is actually quite ugly.

Given a problem on 4-qudits, we can replace each 4-qudit with a collection of four qubits. Each clause is modified to act on the $|\psi_i\rangle$ basis of qubits instead of $|i\rangle$ basis of the 4-qudits. Then, for each 4-qudit that the clause acted on, we add a copy of $H_{4\to 2}$. The above uniqueness property ensures that no other clause can act on the same qubits in any other order, or mix them with any other set of qubits.

In the $\mathcal{C}'$ QCSP, any problem where qubits are mixed or applied in inconsistent orders, can immediately be rejected. Some qubits may not be acted on by any clause, and so not correspond to a d-qudit in $\mathcal{C}$, but then those qubits can simply be ignored. This leaves us with only correctly grouped qubits, in a certain subspace, that thus function equivalently to the d-qudits.

Combined, this gives a faithful reduction from d-qudits to qubits, and back again. It turns a k-local Hamiltonian into a $4\lceil\log_2(d)\rceil k$ local Hamiltonian. It remains to check the complexity of the reductions.

In the above description, the expansion factor is $4\lceil\log_2(d)\rceil$. To get a low circuit complexity, we want the expansion to be a power of two. Thus we round this up to $4\cdot 2^{\lceil\log_2(\lceil\log_2(d)\rceil)\rceil}$, which will denote x; the QCSP $\mathcal{C}'$ is augmented through just adding more qubits to increase the subspace dimension, and then T_1 is modified again to prevent those states from being occupied.

An instance of a k-local QCSP $\mathcal{C}$ can be written down as a list of integers, each given by an integer clause type, and k many integers representing the qudits they act on. The clause types of $\mathcal{C}'$ are in one-to-one correspondence with the clause types of $\mathcal{C}$, so those integers remain unchanged. Each clause acting on qudit i now instead acts on qudits $[xi, xi+1, \dots xi+(x-1)]$ in that order. Thus a reduction circuit f only needs to be able to replicate an integer several times, multiply by a constant power of 2, x, and add a number $i \in [0, x)$. This is in AC^0 (in fact it requires no gates at all).

For a circuit g to convert back from $\mathcal{C}'$ to $\mathcal{C}$, we need to map qubit numbers back to qudits numbers, and check that no qubits are used in inconsistent fashion. For each collection of qubits $[a_1, a_2, \dots a_x]$ that a clause in $\mathcal{C}'$ is acting on, we can map that to the d-qudit number a_1. Thus, many qudit numbers will go unused, but this doesn't affect the correctness: as long as all collections use the same numbers in the same order, they will all be mapped to a_1. To check that all qudits are used in a consistent fashion, we need to check for each pair of collections $([a_1, \dots a_x], [b_1, \dots b_x])$ that they do not use qubits in inconsistently. Logically, this reads:

$$\big(((a_i = b_i) \wedge (a_j = b_j)) \vee ((a_i \neq b_i) \wedge (a_j \neq b_j))\big) \wedge (a_i \neq b_j) \wedge (a_j \neq b_i)$$

and this must be checked for every collection, for every i and j, and then combined by an unbounded fan-in AND. The integer equalities $a_i = b_i$ and $a_i \neq b_i$ can be evaluated with unbounded fan-in AND and OR respectively. This check is all in AC^0. If the check fails, the circuit outputs some fixed clause(s) that are unsatisfiable, otherwise it outputs a repetition of the first clause. This AC^0 circuit checks that the qudits are used consistently, and if they are, gives an equivalent instance in the original language. □

If we don't care about having the reductions be in AC^0, and instead allow P-reductions, then $4\lceil\log_2(d)\rceil k$ locality suffices. This reduction is optimal within a factor of 4, in the sense that encoding one d-qudit in several qubits requires at least $\lceil\log_2(d)\rceil$ many qubits. In Section 4 we showed that there is a 5-local 13-qudit problem that is BQP_1-complete. Together with add here, we have:

Corollary 7.1. *There is a BQP_1-complete QCSP on qubits, with 80-local interactions.*

8 Direct Sum and Direct Product Classes

There is a notion of *direct sum* (denoted by "$\oplus$") and *direct product* (denoted by "$\otimes$") on CSP languages. To be clear, this is an operation on a CSP, not on instances. For instance, we can talk about the languages 3-Colorable $\oplus$ 4-SAT and 3-Colorable $\otimes$ 4-SAT. This notion extends in a natural way to the quantum setting. This notion appears quite natural, but we were unable to find any sources actually discussing this precise notion—possibly because the classical theory is not as exciting, for reasons we will also discuss. Since sum and product CSPs inherit completeness properties from their constituents, we will be able to construct (quantum) CSPs complete for additional classes in this way. We start with the classical case, since the quantum case follows almost identically. I think before jumping to 7.1 we need to introduce a bit of notation. What are domains, constraints, and arity.

8.1 Direct Product Constraint Problems

Let L_1 and L_2 be two CSPs, with domains D_1 and D_2, and allowed constraints C_1 and C_2.

Definition 8.1 (Direct Product of Constraint Problems)**.** *Given the CSPs L_1 and L_2, their* direct product *$L_1 \otimes L_2$ is a CSP whose domain is the Cartesian product $D_1 \times D_2$. Each constraint $c_i \in C_1$ (resp C_2) of locality k leads to a constraint c_i' in $L_1 \otimes L_2$, also of locality k, as follows. A tuple $(v_1, v_2, \dots, v_k) \in (D_1 \times D_2)^k$, where each entry $v_i = (v_{i,1}, v_{i,2})$, belongs to c_i' if the tuple $(v_{i,1}, \dots, v_{k,1})$ belongs to c_i. Each constraint in $L_1 \otimes L_2$ arises this way from a constraint in L_1 or L_2.*

As a direct product, we have *projection* functions π_1 and π_2 that map a constraint problem $\mathcal{P}$ in $L_1 \otimes L_2$ to problems $\mathcal{P}_1$ of L_1 and $\mathcal{P}_2$ of L_2. There are also projections that take variable assignments on $\mathcal{P}$ to variable assignments on $\mathcal{P}_1$ or $\mathcal{P}_2$, by projecting each variable component-wise to D_1 or D_2. The problem $\mathcal{P}$ is then satisfiable if and only if the projections $\mathcal{P}_1$ and $\mathcal{P}_2$ are both satisfiable. This immediately leads to the following properties.

Theorem 7 (Containment of Direct Products)**.** *Let $\mathcal{C}$ be a complexity class that is closed under intersection, and powerful enough to compute projections of a direct product CSP. (For instance, any class closed under reductions by local functions[35].) If the satisfiability problems for CSPs L_1 and L_2 are both contained in $\mathcal{C}$, then the direct product $L_1 \otimes L_2$ is also contained in $\mathcal{C}$.*

Proof. Given an instance $\mathcal{P}$ of $L_1 \otimes L_2$, the projection $\mathcal{P}_1$ can be computed, and then solved, in $\mathcal{C}$. This process forms one language $\mathcal{L}_1$ in $\mathcal{C}$. Likewise, there is a language $\mathcal{L}_2$ in $\mathcal{C}$ for the L_2 projection. Then the intersection of $\mathcal{L}_1$ and $\mathcal{L}_2$ is also in $\mathcal{C}$. □

Being closed under intersection is a very mild requirement, essentially equivalent to being able to compute a logical AND. It is satisfied by all classes discussed in this paper, with simple constructions. It is well-known that all classes discussed up to this point satisfy this requirement. An example of a nontrivial case is the closure of PP under intersection [47]. Similarly, closure under local functions is also a weak requirement, essentially expressing irrelevance up to local changes in problem encoding.

The direct product construction also works the other way, giving hardness:

Theorem 8 (Hardness of Direct Products)**.** *Let $\mathcal{C}$ be a complexity class, so that the satisfiability problem for the CSP L_1 is hard for $\mathcal{C}$ under M-reductions, where M is a class of functions closed under composition with computing projections of a CSP. (It suffices that M is closed under composition with local functions.) Then, for any CSP L_2, the direct product $L_1 \otimes L_2$ is also hard for $\mathcal{C}$.*

[35] Here by *local function* we mean an operation which at each symbol of input outputs zero or one symbols of output, using only the last k symbols of input—for some fixed k. There is a similar notion of local functions for circuits, where each bit of output only depends on finitely many bits of input, but we include geometric locality. This is an extremely weak notion of reduction, much weaker than AC^0-reducibility. The fact that this reduction is weak means that that our theorems apply to essentially all complexity classes of interest.

Proof. The hardness of L_1 for $\mathcal{C}$ under M-reductions means that for any language $\mathcal{L} \in \mathcal{C}$ there is a function $f \in M$ so that $f(s)$ describes a satisfiable instance of L_1 iff $s \in \mathcal{L}$. We can map from the instance of L_1 to an instance of $L_1 \otimes L_2$ with the canonical injection ι_1. This produces an instance of $L_1 \otimes L_2$ which is satisfiable iff $s \in \mathcal{L}$ which is a function also in M, and so the composition $\iota_1 \circ f$ is also in M, and so $L_1 \otimes L_2$ is also hard for $\mathcal{C}$. □

Again, the requirement on the reduction is very mild; for instance, completeness under P- and L-reductions, the two most common notions of completeness, both satisfy this property. These two theorems together lead to a simple result about completeness of direct products.

Definition 8.2 (Pairwise Intersection of Classes)**.** *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are two complexity classes (any sets of languages), then $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$ is the class that denotes the* pairwise intersection *of $\mathcal{C}_1$ and $\mathcal{C}_2$. In other words, it is the class of languages that can be written as the intersection, i.e. the logical AND, of a language in $\mathcal{C}_1$ and a language in $\mathcal{C}_2$.*

Theorem 9 (Completeness of Direct Products)**.** *Let M be a set of functions closed under composition with local functions, and closed under concatenations (i.e. if some $f, g : \Sigma_1^* \to \Sigma_2^*$ are each in M, then $h : x \to f(x)g(x)$ is as well). Let L_1 be a CSP complete under M-reductions for a class $\mathcal{C}_1$, and likewise L_2 be complete for $\mathcal{C}_2$. Assume that each of $\mathcal{C}_1$ and $\mathcal{C}_2$ are closed under reductions by local functions, closed under intersections, and contain the language* All *of all strings, Σ^*. Then, the direct product $L_1 \otimes L_2$ is complete under M-reductions for $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$.*

Proof. First we show containment, then hardness. **Containment:** Since All is in $\mathcal{C}_2$, every language $\mathcal{L}_1$ in $\mathcal{C}_1$ is also in $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$, as $\mathcal{L}_1 = \mathcal{L}_1 \cap \text{All}$. Likewise, every language in $\mathcal{L}_2$ is in $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$.

Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are each individually closed under intersection, the class $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$ will be closed under intersection as well. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are each closed under local functions, $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$ is closed under local functions as well. As $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$ satisfies the requirements of Theorem 7, and the problems L_1 and L_2 are both contained in $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$, by this theorem, $L_1 \otimes L_2$ is contained in $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$.

Hardness: Any language $\mathcal{L}$ in $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$ can be written as $\mathcal{L}_1 \cap \mathcal{L}_2$. Since L_1 is complete (and thus hard) for $\mathcal{C}_1$, and L_2 is complete for $\mathcal{C}_2$, by Theorem 8, their direct product $L_1 \otimes L_2$ is hard for both $\mathcal{C}_1$ and $\mathcal{C}_2$. Accordingly, there are M-computable functions mapping $\mathcal{L}_1$ and $\mathcal{L}_2$ to CSP instances that are satisfiable iff the original strings are in the language. By concatenating these instances (as lists of constraints on numbered variables), we get a new instance that is satisfiable iff both are, i.e. only if the string is in both $\mathcal{L}_1$ and $\mathcal{L}_2$. Since M is closed under concatenation, this is an M-computable reduction, and $L_1 \otimes L_2$ is hard for $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$. □

The assumptions on M in Theorem 9 are again very mild, satisfied for AC^0-reductions or even weaker. This essentially says that if we have constraint problems complete for "reasonable" classes $\mathcal{C}_1$ and $\mathcal{C}_2$, then there is a constraint problem complete for $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2)$.

When is this interesting? If $\mathcal{C}_1 \subseteq \mathcal{C}_2$, then $\mathsf{PI}(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{C}_2$. This is the case with P and NP, the two most widely studied classes, so there is nothing new. For instance, the direct product of 2-SAT and 3-SAT gives another NP-complete CSP. Schaefer's Dichotomy Theorem states that all Boolean CSPs are either in co-NLOGTIME, L-complete, NL-complete, $\oplus\mathsf{L}$-complete, P-complete or NP-complete under AC^0 reductions [48]. With exception of NL and $\oplus\mathsf{L}$, all possible pairs from this list have an obvious containment relation, so the only nontrivial consequence would be that there exists a CSP, on a domain of size four, that is $\mathsf{PI}(\oplus\mathsf{L}, \mathsf{NL})$-complete under AC^0 reductions. But under coarser reductions such as P-reductions, all CSPs collapse to P or NP, so we would have nothing to say. This will become more interesting in the quantum case!

8.2 Direct Sum Constraint Problems

If direct products let us express (informally) a "two-input logical AND" of two CSPs, then direct sums let us express "unbounded-fanin AND of fanin-2 ORs".

Definition 8.3 (Direct Sum of Constraint Problems)**.** *Given the CSPs L_1 and L_2, their* direct sum *$L_1 \oplus L_2$ is a CSP whose domain is the disjoint union $D_1 \uplus D_2$. Each constraint in $L_1 \oplus L_2$ is of the form $c_i \cup \left(D_2^k\right)$, where $c_i \in C_1$ is a constraint of locality k; or it is $c_i \cup \left(D_1^k\right)$ for some $c_i \in C_2$.*

Informally, a constraint $c_i \in C_1$ becomes a constraint in $L_1 \oplus L_2$, stating "all variables in the support belong to D_2, or they are all belong to D_1, and in the latter case they form a satisfying assignment to c_i". Given an instance $\mathcal{P}$ of $L_1 \oplus L_2$ with a single connected component,[36] any satisfying assignment must put all variables in D_1 or put all variables in D_2. Each option can be inspected and reduces to solving one instance of L_1 or L_2. If either one is satisfiable, the component is as well. Solving the whole problem then amounts to doing this check for each connected component. This leads to the following property:

Theorem 10 (Containment of Direct Sums). *Let $\mathcal{C}$ be a complexity class that is closed under union and delimited concatenation. Furthermore assume it is powerful enough to compute projections of a direct sum CSP, and powerful enough to separate a CSP into its connected components. If the satisfiability problems for CSPs L_1 and L_2 are both contained in $\mathcal{C}$, then the direct sum $L_1 \oplus L_2$ is also contained in $\mathcal{C}$.*

Proof. To solve an instance of $L_1 \oplus L_2$, the problem is first separated into its connected components. These become separate problems, but can be treated as one, concatenated and appropriately delimited. (In terms of strings, the problem would be written COMPONENT1|COMPONENT2|COMPONENT3, etc.) This is solvable if each component is solvable individually. Each component $C \in L_1 \oplus L_2$ is solved by computing the projections $\mathcal{P}_1 \in L_1$ and $\mathcal{P}_2 \in L_2$. The component is satisfiable if either projection is satisfiable, and since $\mathcal{C}$ is closed under union, it suffices that each of L_1 and L_2 are in $\mathcal{C}$. □

The first several requirements on $\mathcal{C}$ are again mild. Closure under delimited concatenation here means that if L is a language in $\mathcal{C}$, then $L(dL)^*$ is also in $\mathcal{C}$, where d is a fresh symbol not in the alphabet of L and * is the Kleene star as before. In other words, take any combination of one or more strings from L, and concatenate them with a fresh symbol d in between, segmenting them; the language of all such concatenations must also belong to $\mathcal{C}$. Intuitively, we can recognize this language by splitting at delimiter, checking "each part" of an input string individually and accepting only if they are all individually accepted. The assertion that $\mathcal{C}$ is closed under this operation means that machines recognizing $\mathcal{C}$ are able to carry out such a split-and-check operation.

On the other hand, computing connected components is slightly nontrivial, but as shown in Ref. [49], it can be done in L. Otherwise, these closure properties are satisfied by every "reasonable" class above L.

One might expect that, by analogy with the direct product, the direct sum should then be complete for the pairwise union of two classes. This would be the case if we only ever had to worry about problems that form a single connected component. By embedding a constraint problem from each of L_1 and L_2 on the same set of variables, we get a union-like problem. But by placing each constraint problem on disjoint sets of variables, we get an intersection-like problem, that must satisfy each. This motivates the following definition:

Definition 8.4 (Star of Pairwise Unions of Classes). *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are two complexity classes, then their* star of pairwise unions*, denoted $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$, is a complexity class defined as follows: for each language $L_1 \in \mathcal{C}_1$ and $L_2 \in \mathcal{C}_2$, let d be a fresh symbol that is not in the alphabet of L_1 or L_2. Then, the language $(dL_1|dL_2)^*$ is in $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$. $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$ is the closure of all such languages under L (log-space reductions).*

This definition merits a brief explanation. For a pair of languages L_1 and L_2, what do the strings in the language $L \equiv (dL_1|dL_2)^*$ look like? Given an input string like $d01001110d101101d101001$, it will belong to L if and only if each of $\{01001110, 101101, 101001\}$ belongs to either L_1 or L_2. So, with an oracle for L_1 or L_2, the task becomes segmenting out these s-delimited substrings, querying both oracles to see if each segment passes, and accepting the string only if each segment is accepted. If C is a complexity class powerful enough to recognize L_1 and L_2 itself, and do the segmenting, then $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2) \in C$.

We now formalize and proof these facts.

Theorem 11 (Hardness of Direct Sums). *Let $\mathcal{C}$ be a complexity class and L_1 be a CSP, such that $L_1{-}\mathsf{SAT}$ is hard for $\mathcal{C}$ under M-reductions, where M is a class of functions closed under composition with AC^0. Let L_2 be any CSP with at least one unsatisfiable instance. Then the direct sum $L_1 \oplus L_2$ is also hard for $\mathcal{C}$.*

Proof. The hardness of L_1 for $\mathcal{C}$ under M-reductions means that for any language $\mathcal{L} \in \mathcal{C}$ there is a function $f \in M$ so that $f(s)$ describes a satisfiable instance of L_1 iff $s \in \mathcal{L}$. We can map from the $f(s)$ instance of

[36] A *connected component* of a CSP is a connected component in the graph for that CSP, where the vertices are variables, and there is an edge between variables if they share a constraint.

L_1 to an instance s' of $L_1 \oplus L_2$ with the canonical injection ι_1. This produces an instance of $L_1 \oplus L_2$ which is satisfiable whenever s is, by mapping a satisfying assignment along ι_1; but it is also trivially satisfiable by assigning variables any value in the domain of L_2.

So, let U be any unsatisfiable instance of L_2, and let v_U be the number of variables in that instance. Modify s' by adding dummy variables until the number is a multiple of v_U. Arbitrarily partitioning the variables of s' into sets of size v_U, add constraints representing $v_{s'}/v_U$ many copies of U to the problem.

The resulting problem s' is an instance of $L_1 \oplus L_2$. Since each variable belongs to one of the copies of U, there cannot be a satisfying instance of s' that assigns any variable a value in the domain of L_2. The dummy variables can take any value in the domain of L_1, while the original variables produce a satisfying instance of s.

In this way, we've produced an instance of $L_1 \oplus L_2$ which is satisfiable iff $s \in \mathcal{L}$. The initial mapping $f : \mathcal{L} \to L_1$ is in M, and the embedding from L_1 to $L_1 \oplus L_2$ is in AC^0, so the (?) is also in M. Since this mapping is complete, sound, and in M for any $\mathcal{L} \in \mathcal{C}$, we have proved that $L_1 \oplus L_2$ is also hard for $\mathcal{C}$. □

Theorem 12 (Completeness of Direct Sums)**.** *Let be M a set of functions closed under composition with logspace-computable functions (such as the set of logspace functions themselves, FL). Let L_1 be a CSP complete under M-reductions for a class $\mathcal{C}_1$, and likewise L_2 be M-complete for $\mathcal{C}_2$. Assume that each of $\mathcal{C}_1$ and $\mathcal{C}_2$ are closed under M-reductions, and contain the language* None *of no strings, $\emptyset$. Then, the direct sum $L_1 \oplus L_2$ is complete under M-reductions for $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$.*

Proof. First we show containment, then hardness.

Containment: We might seek to apply Theorem 10 to show that, since L_1 and L_2 are each in $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$, their direct sum is too. This approach is limited by the fact that SoPU is not generally going to be closed under union, a necessary property for Theorem 10. But the same algorithm described in the proof of Theorem 10 essentially applies.

To solve an instance of $L_1 \oplus L_2$ in $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$, first separate the CSP into its connected components, and join them in a delimited fashion, e.g. sComponent1sComponent2sComponent3. The problem is solvable if each component is solvable individually. Each component c_i is an instance of $L_1 \oplus L_2$ and has projections $p_{i,1} \in L_1$ and $p_{i,2} \in L_2$. The component is satisfiable if either projection is. So reduce (in logspace) this to the string $sp_{1,1}tp_{1,2}sp_{2,1}tp_{2,2}sp_{3,1}tp_{3,2}\ldots$, where t is another new fresh symbol. The resulting language is not the star-of-union of the languages L_1 and L_2, but of two other languages $L_1' := L_1 t \Sigma^*$ and $L_2' := \Sigma^* t L_2$. Since L_1' and L_2' are equivalent to L_1 and L_2 under logspace reductions, and L_1 and L_2 belong to the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ which are closed under logspace reductions, L_1' and L_2' belong to $\mathcal{C}_1$ and $\mathcal{C}_2$ as well, so this language belong to $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$.

Hardness: Any language $\mathcal{L}$ in $\mathsf{SoPU}(\mathcal{C}_1, \mathcal{C}_2)$ is equivalent under an L reduction to $(s\mathcal{L}_1|s\mathcal{L}_2)^*$. Since L_1 is complete for $\mathcal{C}_1$ and $\mathcal{L}_1 \in \mathcal{C}_1$, we can map from $\mathcal{L}_1$ to equivalent instances of L_1, and likewise for L_2. So given a string for which we want to test membership in $(s\mathcal{L}_1|s\mathcal{L}_2)^*$, we use the following logspace reduction to $L_1 \oplus L_2$:

Segment the string by s. Each segment will produce a disconnected part of the resulting CSP. One string segment x then gets mapped to instances p_1 of L_1 and p_2 of L_2. Separate p_1 and p_2 into their connected components, labelled $p_{1,i}$ and $p_{2,j}$. Then, for each pair of components (i, j), output a set of $\max(|p_{1,i}|, |p_{2,j}|)$ many fresh variables, and map the constraints from $p_{1,i}$ and $p_{2,j}$ onto these variables. The resulting CSP will be satisfiable iff the string belonged to $(s\mathcal{L}_1|s\mathcal{L}_2)^* = (s(\mathcal{L}_1|\mathcal{L}_2))^*$: each individual segment $x \overset{?}{\in} (\mathcal{L}_1|\mathcal{L}_2)$ should be accepted only if at least one of p_1 or p_2 are satisfiable. p_1 is satisfiable iff all $p_{1,i}$ are satisfiable, and likewise for p_2 and $p_{2,j}$. Each output component (i, j) is satisfiable only if $p_{1,i}$ or $p_{2,j}$ are, and taking the intersection of these conditions, we see that this is exactly equivalent to the original problem. □

8.3 Quantum Sums and Products

The above constructions transfer in a very natural way to the quantum setting. Now, instead of domains that are a Cartesian product or disjoint union, the Hilbert spaces are a tensor product or direct sum. The clauses are accordingly built as tensor products and direct sums.

Definition 8.5 (Direct Product of Quantum Constraint Problems)**.** *Given the QCSPs L_1 and L_2, their* direct product *$L_1 \otimes L_2$ is a QCSP whose domain is the tensor product Hilbert space $D_1 \otimes D_2$. Each operator H_i in L_1 leads to an operator $H_i \otimes I$, a tensor product with the identity, and likewise for L_2.*

Definition 8.6 (Direct Sum of Quantum Constraint Problems)**.** *Given the QCSPs L_1 and L_2, their* direct sum *$L_1 \uplus L_2$ is a QCSP whose domain is the direct sum Hilbert space $D_1 \oplus D_2$. Each operator H_i in L_1 leads to an operator $H_i \oplus 0$, a direct sum with the 0 operator, requiring that a frustration-free state lies in the nullspace of H_i* or *the right half of the direct sum (or a linear combination). Likewise for operators in L_2.*

These have the same essential properties as the direct product and sum for classical CSPs, that we can produce product and sum instances that are satisfiable iff both (resp. either) of the original instances were satisfiable.

TODO: I think the precise statements of the resulting theorems, as well as the versions for promise problems (which in principle we need) would be best suited to an appendix. It will be otherwise very repetitive with the above, but sadly I don't see a way to actually do the proofs together.

8.4 Basic Properties

Above we have proved that $\mathsf{PI}(A, B)$ and $\mathsf{SoPU}(A, B)$ have constraint problems complete for them as a class, as long the original classes A and B do. Motivated by this, we give some basic properties of these classes.

Theorem 13. *If the class B includes the language* ALL *of all strings, then $A \subseteq \mathsf{PI}(A, B)$.*

Proof. Since $\text{ALL} \in B$, for any $a \in A$, we have $a = \{a\} \cap \text{ALL} \in \mathsf{PI}(A, B)$. □

Theorem 14. *If the class B includes the language* NONE *of no strings, then $A \subseteq \mathsf{SoPU}(A, B)$.*

Proof. Take any $L_1 \in A$. Simplify $(s(L_1|\emptyset))^* = (sL_1)^*$, and then we can do a reduction in logspace to only look at the first s-delimited subsequence to get $L_1 \in \mathsf{SoPU}(A, B)$ □

Theorem 15. PI *and* SoPU *respect the inclusion order of complexity classes. That is, $A \subseteq C$ and $B \subseteq D$ implies $\mathsf{PI}(A, B) \subseteq \mathsf{PI}(C, D)$ and $\mathsf{SoPU}(A, B) \subseteq \mathsf{SoPU}(C, D)$.*

Proof. Immediate from the definitions. □

The SoPU construction generally leads to a more powerful class than the PI construction.

Theorem 16. *If classes A and B are closed under reduction by local functions, then $\mathsf{PI}(A, B) \subseteq \mathsf{SoPU}(A, B)$.*

Proof. A language in $\mathsf{PI}(A, B)$ is of the form $L_A \cap L_B$. Since A and B are closed under reduction by local functions, the languages $0L_A$ and $1L_B$—the languages where we add one additional 0 or 1 to the start of all strings—are also in A and B respectively. Applying the SoPU construction to this pair of languages, we get the language $(s0L_A|s1L_B)^*$. Then, we can apply the logspace reduction (while staying in $\mathsf{SoPU}(A, B)$) of mapping any string x (over an alphabet without s) to the string $s0xs1x$, so that x is only accepted if it belongs to both L_A and L_B. Thus we have a mechanism to recognize $L_A \cap L_B$ in $\mathsf{SoPU}(A, B)$. □

PI and SoPU do not increase the power of classes by combining with something weaker. Formally:

Theorem 17. *If A is closed under intersection and $B \subseteq A$, then $\mathsf{PI}(A, B) \subseteq A$. If A is closed under logspace reductions, unions, and delimited concatenation, then $\mathsf{SoPU}(A, B) \subseteq A$.*

Proof. The first is immediate from the definition. The second comes from applying each closure property to $(s\mathcal{L}_A|s\mathcal{L}_B)^*$ to see that it is still in A. □

This explains why these constructions have not been of much interest until the quantum setting. In the classical setting, the only two complexity classes (up to P-reduction) that arise from constraint problems are P and NP. Both are closed under all the properties discussed above (logspace, unions, intersections, and delimited concatenation), and $\mathsf{P} \subseteq \mathsf{NP}$. Thus we can conclude, by Theorems 13, 14, and 17:

Corollary 8.1. $\mathsf{PI}(\mathsf{P}, \mathsf{NP}) = \mathsf{SoPU}(\mathsf{P}, \mathsf{NP}) = \mathsf{NP}$.

8.5 New Complete Classes

The 7 classes discussed so far as having complete constraint problems are P, coRP, BQP_1, NP, QCMA_1, and QMA_1. Recognize that all of these have all of the closure properties discussed in this section so far: union, intersection, logspace reductions, and delimited concatenation; and they all include the trivial problems ALL and NONE.

Among these seven classes, most pairs $\{A, B\}$ have $A \subseteq B$, in which cases $\mathsf{PI}(A, B) = \mathsf{SoPU}(A, B) = B$. There are just three pairs that are not known to contain each other: $\mathsf{coRP} \overset{?}{\subseteq} \mathsf{NP}$, $\mathsf{BQP}_1 \overset{?}{\subseteq} \mathsf{NP}$, and $\mathsf{BQP}_1 \overset{?}{\subseteq} \mathsf{MA}$. Each of these pairs leads to two classes $\mathsf{PI}(A, B)$ and $\mathsf{SoPU}(A, B)$, that are not obviously equal to some other known class, for a total of six more complexity classes with complete quantum constraint problems. The resulting thirteen classes are shown in Figure 1.

Notably, $\mathsf{coRP} \overset{?}{\subseteq} \mathsf{NP}$ involves only classical classes, and accordingly one could hope that the $\mathsf{PI}(\mathsf{coRP}, \mathsf{NP})$ and $\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP})$ would have more existing theory already developed around them. There are at least two interesting cases where a collapse can occur:

1. If $\mathsf{P} = \mathsf{RP}$ (derandomization), then $\mathsf{coRP} = \mathsf{P} \subseteq \mathsf{NP}$ and so

   $$\mathsf{NP} \subseteq \mathsf{PI}(\mathsf{coRP}, \mathsf{NP}) \subseteq \mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP}) = \mathsf{SoPU}(\mathsf{NP}, \mathsf{NP}) = \mathsf{NP}$$

   But an even weaker version of derandomization, the possibility that $\mathsf{NP} = \mathsf{MA}$, would also lead to a collapse, because $\mathsf{coRP} \subseteq \mathsf{MA}$:

   $$\mathsf{NP} \subseteq \mathsf{PI}(\mathsf{coRP}, \mathsf{NP}) \subseteq \mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP}) \subseteq \mathsf{SoPU}(\mathsf{MA}, \mathsf{NP}) = \mathsf{SoPU}(\mathsf{NP}, \mathsf{NP}) = \mathsf{NP}$$

2. Similarly, if $\mathsf{NP} = \mathsf{coNP}$ (concise refutations), then $\mathsf{coRP} \subseteq \mathsf{coNP} = \mathsf{NP}$ and we have a collapse.

The complexity class $\mathsf{DP} = \mathsf{PI}(\mathsf{NP}, \mathsf{coNP})$ has been studied before, and forms the second layer of the *boolean hierarchy* BH: $\mathsf{DP} = \mathsf{BH}_2$, see [29, 28] for discussion. Theorem 15 tells us that $\mathsf{PI}(\mathsf{coRP}, \mathsf{NP}) \subseteq \mathsf{DP}$ as a result. No inclusions are known between DP and MA.

$\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP})$ does not obviously lie in the Boolean hierarchy. If it was a simple pairwise union, it would lie in BH_3, the pairwise union of DP and NP. But the long list of checks to do means that we cannot obviously condense down the queries. Queries to an NP oracle don't need to depend on each other adaptively, though, so we do know that $\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP})$ is contained in $\mathsf{P}^{||\mathsf{NP}} = \mathsf{P}^{\mathsf{NP}[\log]}$, studied in [50, 51]; problems that can be solved by a P machine with polynomially many nonadaptive queries, or logarithmically many adaptive queries. This would generally be seen as a harder class than MA, but since $\mathsf{MA} \overset{?}{\subseteq} \Delta_2\mathsf{P} = \mathsf{P}^{\mathsf{NP}}$ is not even known, this is not implied by $\mathsf{SoPU}(\mathsf{coRP}, \mathsf{NP}) \subseteq \mathsf{MA}$.

TODO: I think I had a proof that if $NP = SoPU(coRP, NP)$, then $MA = NP$ would collapse, but I can't remember it now.

Your arguments so far relate to languages and not promise problems. Is it clear how these arguments translate to promise problems? Could we restate Theorem 5 and prove it here using the machinery you have mentioned?

The quantum classes, that involve BQP_1 with NP or MA, it seems difficult to say anything at all about—-besides the fact that they lie above BQP_1 and below QCMA, and between these there are very few (any?) classes ever described.

Acknowledgements

We thank Alex Grilo and Dorian Rudolph for helpful discussions, and the authors of Ref. [52] for the inspiration of Fig. 8. RR was supported by . . .

9 To-do

1. Recheck discussion about active dots in Section 3 and Section 4.

References

[1] Alexei Y. Kitaev, Alexander Shen, and Mikhail N. Vyalyi. *Classical and quantum computation.* 47. American Mathematical Soc., 2002.

[2] Julia Kempe, Alexei Kitaev, and Oded Regev. “The Complexity of the Local Hamiltonian Problem”. In: *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science.* Springer, 2004, pp. 372–383. ISBN: 978-3-540-24058-7. DOI: 10.1007/978-3-540-30538-5_31.

[3] Sergey Bravyi. *Efficient algorithm for a quantum analogue of 2-SAT.* 2006. arXiv: quant-ph/0602108 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0602108.

[4] David Gosset and Daniel Nagaj. “Quantum 3-SAT is QMA_1-complete”. In: *SIAM Journal on Computing* 45.3 (2016), pp. 1080–1128.

[5] Matthew Amy et al. “Exact Synthesis of Multiqubit Clifford-Cyclotomic Circuits”. In: *Reversible Computation.* Ed. by Torben Ægidius Mogensen and Łukasz Mikulski. Cham: Springer Nature Switzerland, 2024, pp. 238–245.

[6] Sergey Bravyi, Arvid J. Bessen, and Barbara M. Terhal. *Merlin-Arthur Games and Stoquastic Complexity.* 2006. arXiv: quant-ph/0611021 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0611021.

[7] Sergey Bravyi and Mikhail Vyalyi. “Commutative version of the local Hamiltonian problem and common Eigenspace problem”. In: *Quantum Info. Comp.* 5 (May 2005), pp. 187–215. DOI: 10.26421/QIC5.3-2.

[8] Yi-Kai Liu, Matthias Christandl, and F. Verstraete. “Quantum Computational Complexity of the N-Representability Problem: QMA Complete”. In: *Phys. Rev. Lett.* 98 (11 Mar. 2007), p. 110503. DOI: 10.1103/PhysRevLett.98.110503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.98.110503.

[9] Tzu-Chieh Wei, Michele Mosca, and Ashwin Nayak. “Interacting Boson Problems Can Be QMA Hard”. In: *Phys. Rev. Lett.* 104 (4 Jan. 2010), p. 040501. DOI: 10.1103/PhysRevLett.104.040501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.104.040501.

[10] Norbert Schuch and Frank Verstraete. “Computational complexity of interacting electrons and fundamental limitations of density functional theory”. In: *Nature physics* 5.10 (2009), pp. 732–735.

[11] Andrew M Childs, David Gosset, and Zak Webb. “The Bose-Hubbard model is QMA-complete”. In: *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014.* Springer. 2014, pp. 308–319.

[12] Roberto Oliveira and Barbara M. Terhal. “The complexity of quantum spin systems on a two-dimensional square lattice”. In: *Quantum Info. Comput.* 8.10 (Nov. 2008), pp. 900–924. ISSN: 1533-7146.

[13] Sean Hallgren, Daniel Nagaj, and Sandeep Narayanaswami. “The Local Hamiltonian Problem on a Line with Eight States is QMA-Complete”. In: *Quantum Info. Comput.* 13.9–10 (Sept. 2013), pp. 721–750. ISSN: 1533-7146.

[14] Dorit Aharonov, Oded Kenneth, and Itamar Vigdorovich. “On the Complexity of Two Dimensional Commuting Local Hamiltonians”. In: *13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).* Ed. by Stacey Jeffery. Vol. 111. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 2:1–2:21. ISBN: 978-3-95977-080-4. DOI: 10.4230/LIPIcs.TQC.2018.2. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2018.2.

[15] Sandy Irani and Jiaqing Jiang. *Commuting Local Hamiltonian Problem on 2D beyond qubits.* 2023. arXiv: 2309.04910 [quant-ph]. URL: https://arxiv.org/abs/2309.04910.

[16] Asad Raza, Jens Eisert, and Alex B. Grilo. *Complexity of geometrically local stoquastic Hamiltonians.* 2024. arXiv: 2407.15499 [quant-ph]. URL: https://arxiv.org/abs/2407.15499.

[17] Toby Cubitt and Ashley Montanaro. “Complexity classification of local Hamiltonian problems”. In: *SIAM Journal on Computing* 45.2 (2016), pp. 268–316.

[18] Sergey Bravyi and Matthew Hastings. “On complexity of the quantum Ising model”. In: *Communications in Mathematical Physics* 349.1 (2017), pp. 1–45.

[19] Toby S. Cubitt, Ashley Montanaro, and Stephen Piddock. “Universal quantum Hamiltonians”. In: *Proceedings of the National Academy of Sciences* 115.38 (2018), pp. 9497–9502. DOI: 10.1073/pnas.1804949115. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1804949115.

[20] Stephen Piddock and Ashley Montanaro. “Universal qudit Hamiltonians”. In: *Communications in Mathematical Physics* 382 (2021), pp. 721–771.

[21] Stephen A Cook. “The complexity of theorem-proving procedures”. In: *Proceedings of the third annual ACM symposium on Theory of computing*. 1971, pp. 151–158.

[22] L. A. Levin. “Universal Sequential Search Problems”. In: *Probl. Peredachi Inf.* 9:3 (1973), pp. 115–116.

[23] Alberto Baiardi, Michał Lesiuk, and Markus Reiher. “Explicitly correlated electronic structure calculations with transcorrelated matrix product operators”. In: *Journal of Chemical Theory and Computation* 18.7 (2022), pp. 4203–4217.

[24] Francesco Petiziol et al. “Quantum simulation of three-body interactions in weakly driven quantum systems”. In: *Phys. Rev. Lett.* 126.25 (2021), p. 250504.

[25] J. H. Busnaina et al. *Native Three-Body Interactions in a Superconducting Lattice Gauge Quantum Simulator*. 2025. arXiv: 2501.13383 [quant-ph]. URL: https://arxiv.org/abs/2501.13383.

[26] Alexander Y. Chuang et al. *Observation of a Halo Trimer in an Ultracold Bose-Fermi Mixture*. 2024. arXiv: 2411.04820 [cond-mat.quant-gas]. URL: https://arxiv.org/abs/2411.04820.

[27] R. Cruz-Torres et al. “Probing Few-Body Nuclear Dynamics via ^{3}H and ^{3}He $(e, e^{'}p)$pn Cross-Section Measurements”. In: *Phys. Rev. Lett.* 124 (21 May 2020), p. 212501. DOI: 10.1103/PhysRevLett.124.212501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.212501.

[28] C.H. Papadimitriou and M. Yannakakis. “The complexity of facets (and some facets of complexity)”. In: *Journal of Computer and System Sciences* 28.2 (Apr. 1984), pp. 244–259. ISSN: 0022-0000. DOI: 10.1016/0022-0000(84)90068-0. URL: http://dx.doi.org/10.1016/0022-0000(84)90068-0.

[29] Jin-Yi Cai et al. “The Boolean Hierarchy I: Structural Properties”. In: *SIAM Journal on Computing* 17.6 (Dec. 1988), pp. 1232–1252. ISSN: 1095-7111. DOI: 10.1137/0217078. URL: http://dx.doi.org/10.1137/0217078.

[30] Alex Meiburg. *Quantum Constraint Problems can be complete for* BQP*,* QCMA*, and more*. 2021. arXiv: 2101.08381 [quant-ph]. URL: https://arxiv.org/abs/2101.08381.

[31] Dorian Rudolph. *Towards a universal gateset for* QMA_1. 2024. arXiv: 2411.02681 [quant-ph]. URL: https://arxiv.org/abs/2411.02681.

[32] Richard P Feynman. “Quantum mechanical computers.” In: *Found. Phys.* 16.6 (1986), pp. 507–532.

[33] Dorit Aharonov and Alex B. Grilo. “Two Combinatorial MA-Complete Problems”. In: *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Ed. by James R. Lee. Vol. 185. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 36:1–36:20. ISBN: 978-3-95977-177-1. DOI: 10.4230/LIPIcs.ITCS.2021.36. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2021.36.

[34] Stephen P. Jordan et al. “Achieving perfect completeness in classical-witness quantum Merlin-Arthur proof systems”. In: *Quantum Info. Comput.* 12(5-6) (2012), pp. 461–471.

[35] Thomas J. Schaefer. “The complexity of satisfiability problems”. In: *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*. STOC ’78. San Diego, California, USA: Association for Computing Machinery, 1978, pp. 216–226. ISBN: 9781450374378. DOI: 10.1145/800133.804350. URL: https://doi.org/10.1145/800133.804350.

[36] Dmitriy Zhuk. “A proof of the CSP dichotomy conjecture”. In: *Journal of the ACM (JACM)* 67.5 (2020), pp. 1–78.

[37] A Yu Kitaev. “Quantum computations: algorithms and error correction”. In: *Russian Mathematical Surveys* 52.6 (1997), p. 1191.

[38] Brett Giles and Peter Selinger. “Exact synthesis of multiqubit Clifford+ T circuits”. In: *Physical Review A* 87.3 (2013), p. 032332.

[39] Dorit Aharonov and Tomer Naveh. *Quantum NP - A Survey.* 2002. arXiv: quant-ph/0210077 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0210077.

[40] Scott Aaronson. “On perfect completeness for QMA”. In: *Quantum Info. Comput.* 9 (2009), pp. 81–89.

[41] Sergey Bravyi et al. “The complexity of stoquastic local Hamiltonian problems”. In: *Quantum Info. Comput.* 8.5 (2008), pp. 361–385. ISSN: 1533-7146.

[42] Sevag Gharibian et al. “Quantum hamiltonian complexity”. In: *Foundations and Trends in Theoretical Computer Science* 10.3 (2015), pp. 159–282.

[43] Sevag Gharibian and Jamie Sikora. “Ground state connectivity of local Hamiltonians”. In: *International Colloquium on Automata, Languages, and Programming.* Springer. 2015, pp. 617–628.

[44] Pawel Wocjan, Dominik Janzing, and Thomas Beth. “Two QCMA-complete problems”. In: *Quantum Info. Comput.* 3.6 (2003), pp. 635–643. ISSN: 1533-7146.

[45] David Gosset, Jenish C. Mehta, and Thomas Vidick. “QCMA hardness of ground space connectivity for commuting Hamiltonians”. In: *Quantum* 1 (2017), p. 16.

[46] Ying-hao Chen. *2-Local Hamiltonian with Low Complexity is QCMA*. 2019. arXiv: 1909.03787 [cs.CC]. URL: https://arxiv.org/abs/1909.03787.

[47] Richard Beigel, Nick Reingold, and Daniel Spielman. “PP is closed under intersection”. In: *Proceedings of the twenty-third annual ACM symposium on Theory of computing - STOC ’91.* STOC ’91. ACM Press, 1991, pp. 1–9. DOI: 10.1145/103418.103426. URL: http://dx.doi.org/10.1145/103418.103426.

[48] Eric Allender et al. “The complexity of satisfiability problems: Refining Schaefer’s theorem”. In: *Journal of Computer and System Sciences* 75.4 (June 2009), pp. 245–254. ISSN: 0022-0000. DOI: 10.1016/j.jcss.2008.11.001. URL: http://dx.doi.org/10.1016/j.jcss.2008.11.001.

[49] Omer Reingold. “Undirected connectivity in log-space”. In: *Journal of the ACM* 55.4 (Sept. 2008), pp. 1–24. ISSN: 1557-735X. DOI: 10.1145/1391289.1391291. URL: http://dx.doi.org/10.1145/1391289.1391291.

[50] Samuel R. Buss and Louise Hay. “On truth-table reducibility to SAT”. In: *Information and Computation* 91.1 (Mar. 1991), pp. 86–102. ISSN: 0890-5401. DOI: 10.1016/0890-5401(91)90075-d. URL: http://dx.doi.org/10.1016/0890-5401(91)90075-D.

[51] Lane A. Hemachandra. “The strong exponential hierarchy collapses”. In: *Journal of Computer and System Sciences* 39.3 (Dec. 1989), pp. 299–322. ISSN: 0022-0000. DOI: 10.1016/0022-0000(89)90025-1. URL: http://dx.doi.org/10.1016/0022-0000(89)90025-1.

[52] Chris Cade, Marten Folkertsma, and Jordi Weggemans. *Complexity of the Guided Local Hamiltonian Problem: Improved Parameters and Extension to Excited States.* 2024. arXiv: 2207.10097 [quant-ph]. URL: https://arxiv.org/abs/2207.10097.

A Proofs of Lemmas: BQP_1 with monogamy

Lemma 3.1 (Single-type qudits)**.** *If there is a* 17*-dimensional qudit in the instance acted on by at least two projectors* Π_α *and* $\Pi_{\alpha'}$*, where* $\alpha, \alpha' \in \{C, E, L\}$ *and* $\alpha \neq \alpha'$*, the instance is unsatisfiable.*

Proof. Let $|\psi\rangle$ and $|\phi\rangle$ be states in the null space of Π_α and Π'_α respectively. Since Π_α and $\Pi_{\alpha'}$ act on orthogonal subspaces, it follows that $|\psi\rangle$ and $|\phi\rangle$ are orthogonal and $|\psi\rangle \neq |\phi\rangle$. Therefore, there is no state that can simultaneously satisfy both Π_α and $\Pi_{\alpha'}$, making the instance unsatisfiable. □

A.1 Clock component

Lemma 3.2 (Clock qudit two neighbor maximum)**.** *If a clock qudit in a clock component is connected to more than two other clock or endpoint qudits, the instance is unsatisfiable.*

Proof. Recall that any clause from $\{\Pi_{init}, \Pi_{prop}, \Pi_{out}\}$ acting on a clock qudit requires one of the clock qudit's auxiliary subspaces to form a Bell pair. A clock qudit is only equipped with two such subspaces (*CA* and *CB*) and so a clock qudit that is connected to more than two clock or endpoint qudits necessarily entails a violation of monogamy of entanglement. □

Lemma 3.3 (Unique direction of a clock chain)**.** *If a clock qudit in the chain has two successors or two predecessors, the instance is unsatisfiable.*

Proof. Recall that satisfying a Π_{prop} clause between two clock qudits c_0 and c_1 where c_0 is the predecessor of c_1 requires forming a Bell pair between the *CB* subspace of c_0 and the *CA* subspace of c_1. Then, in the case where a clock qudit has two predecessors, satisfying both Π_{prop} clauses demands that its *CA* subspace is maximally entangled with the *CB* subspace of two different clock qudits. This is a violation of monogamy of monogamy of entanglement. Similarly, if a clock qudit has two successors, its *CB* subspace would also be required to be maximally entangled with two different clock qudits. □

Lemma 3.4 (Unique endpoint qudit for Π_{init}/Π_{out} clauses)**.** *If an endpoint qudit is connected to more than one clock qudit, or connected to a single clock qudit via a Π_{init} and Π_{out} clause, the instance is unsatisfiable.*

Proof. Satisfying a single Π_{init} or Π_{out} clause requires the endpoints' single subspace *EC* to be maximally entangled with one of the subspaces of the clock qudit. Then, any two clauses that require entangling *EC* to two different subspaces (even if its the same *CA* or *CB* of two distinct clock particles) would entail a violation of monogamy of entanglement. □

Lemma 3.5 (Unique clock qudit for Π_{init}/Π_{out} clauses)**.** *Let c_0 refer to the clock qudit that all Π_{prop} clauses point away from, and c_L the one at the other end of the chain. If a Π_{init} clause acts on a clock qudit that is not c_0, the instance is unsatisfiable. Similarly, if a Π_{out} clause acts on a clock qudit that is not c_L, the instance is unsatisfiable.*

Proof. Satisfying a Π_{init} clause implies that the *CA* subspace of the clock qudit present in the clause is maximally entangled with the *EC* subspace of the endpoint qubit. Similarly, Π_{prop} clauses require that the *CB* subspace of the predecessor is maximally entangled with the *CA* subspace of its successor. Then, in a one-dimensional chain, there is only a single free *CA* subspace on clock qudit c_0. Therefore, any Π_{init} clause that acts on a clock qudit other than c_0 violates the monogamy of entanglement conditions. The same argument applies to Π_{out}, the *CB* subspace, and c_L. □

Lemma 3.6 (Lack of an ACC)**.** *If a clock component does not have at least one Π_{init} and one Π_{out} clause, the sub-instance is satisfiable.*

Proof. If the clock component has no Π_{init} (Π_{out}), the assignment where all clock qudits are set to $|r\rangle$ ($|d\rangle$) is a satisfying assignment. The Π_{init} and Π_{out} clauses contain the Π_{start} and Π_{stop} projectors forbidding clock qudits to be in the inactive states $|r\rangle$ and $|d\rangle$. Then, if one clause is missing in the clock component, the clock qudits can be set to the inactive state that is not forbidden. Furthermore, this assignment does not violate any of the Π_{clock} clauses (whether they are $\Pi_{clock,D}$ or $\Pi_{clock,?}$) since these allow both the successor and predecessor to be in the same inactive state. Additionally, the logical qudits of the Π_{prop} clauses remain unconstrained since there is not active state to demand otherwise.

Clearly, if the instance has neither of these two clauses, then either assignment of the clock qudits suffices. □

Proposition 3.1 (Initialization of logical qudits)**.** *Let $c_0, \ldots, c_L$ with $L \geq 0$ be the clock qudits of an ACC. Let c_0 be the qudit present in the Π_{init} clauses and c_L the one in the Π_{out} clauses. A state of the qudits of the sub-instance where c_0 is not $|a\rangle$ in any basis state of the superposition is not a satisfying state.*

Proof. For the sake of contradiction, assume that a satisfying state cannot have clock qudit c_0 in the state $|a\rangle$. Let us show that all other assignments of the clock qudits result in either a violation of the assumption or a violation of a clause in the sub-instance.

To begin, observe that c_0 cannot be $|r\rangle$ as this would violate Π_{start}. c_0 must then be permanently fixed to $|d\rangle$. Now, observe that if $\Pi_{prop}^{(c_0,c_1)}$ acts on an undefined logical qudit, this results in a violation of the

$\Pi_{clock,?}$ clause within since $|d_0 \mathrm{x}_1\rangle$ is not in the null space of this clause for any $\mathrm{x} \in \{r, a, d\}$. Thus, let us consider the case where this first Π_{prop} clause is well-defined. In this case, the $\Pi_{clock,D}$ clause only allows c_1 to be either $|d\rangle$ or $|a\rangle$. If we let c_1 be in state $|a\rangle$, satisfying $\Pi_{prop}^{(c_0,c_1)}$ requires the active spot to shift towards the left (see Eq. (22)), in which case c_0 becomes $|a\rangle$, contradicting the assumption. Therefore c_1 must also be $|d\rangle$. By induction, these steps show that the only state of the clock that can satisfy the Π_{prop} (and Π_{clock} clauses within) must be $|d_1 d_2 \dots d_L\rangle$. However, this state violates the Π_{stop} clause acting on clock qudit c_L. Therefore, there is no state that avoids contradicting the assumption or a violation of one the clauses. □

Proposition 3.2 (Unique active state in a TACC)**.** *Let $c_0, \dots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. A state in which at any given time there are two clock qudits c_a and c_b with $0 \leq a < b \leq T$ in the state $|a\rangle$ is not a satisfying state.*

Proof. Observe that since all Π_{prop} clauses within the chain are oriented towards c_T, the only assignment of the clock qudits that agrees with the $\Pi_{clock,D}$ clauses is that where qudits c_i with $i < a$ are in the state $|d\rangle$ and qudits c_i with $i > a$ are in the state $|r\rangle$. Therefore, if c_b is also in the state $|a\rangle$, there must be a violation of a $\Pi_{clock,D}$ clause. □

Lemma 3.7 (Satisfying state of a truncated ACC)**.** *Let $c_0, \dots, c_L$ be the clock qudits of an ACC, and suppose $\Pi_{prop}^{(c_T,c_{T+1})}$ with $0 \leq T < L$ is the first propagation clause that acts on a free logical qudit l_u. The clock qudits $c_0, \dots, c_T$ then form a TACC. Furthermore, assume the TACC has no simultaneous propagation clauses, and let $U_1, \dots, U_T$ be the unitaries associated with the Π_{prop} clauses of this chain. Finally, let S denote the set of all logical qudits of the sub-instance, and D with $|D| = q$ the subset of these qudits that are initialized by the Π_{init} clause on c_0. Under these conditions, the sub-instance is trivially satisfiable and the satisfying state is given by*

$$|\psi_{thist}\rangle := \left[\frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t \dots U_0 |0\rangle^{\otimes q} \otimes |?\rangle_{l_u} \otimes |\underbrace{d \dots d}_{t} a_t \underbrace{r \dots r}_{T\text{-}t}\rangle \right] \otimes |\phi\rangle_{S \setminus D} \otimes |\underbrace{r \dots r}_{L\text{-}T}\rangle , \tag{24}$$

where $|\phi\rangle$ is an arbitrary state of the logical qudits of the sub-instance not present in clauses within the TACC.

Proof. To show that this state is a satisfying assignment of the sub-instance, we first consider the satisfying assignment of the TACC. Afterwards, we determine which states the other qudits should be in order to satisfy the remaining clauses.

According to Proposition 3.1 and Proposition 3.2, if a satisfying assignment of the TACC exists, then it must must contain exactly one clock qudit in an active state at any given time. These conditions mirror those of an active clock chain devoid of undefined clauses, which we know is most likely satisfied by a history state. The only uncertainty is whether this state also satisfies the final Π_{out} clauses. For the TACC considered here, there is no such uncertainty as the Π_{out} clauses on qudit c_L are never part the chain. Thus, this TACC is satisfied by the history state expressing the computation of circuit $U_T \dots U_1$.

For the satisfiability of the clauses outside the TACC, observe that the undefined $\Pi_{prop}^{(c_T,c_{T+1})}$ clause can be satisfied alongside the TACC by fixing the state of the successor qudit to $|r\rangle$. Similarly, the Π_{prop} clauses acting on the clock qudits that follow $(c_{T+1}, \dots, c_L)$ can also be satisfied by fixing the clock qudits to state $|r\rangle$, regardless of whether they are well-defined or undefined. As a consequence, the clauses outside the TACC do not exert any constraints on the logical qudits. Finally, observe that since the logical qudits in the set $S \setminus D$ do not form part of clauses within the TACC, they may be in any state $|\phi\rangle$. Eq. (24) is the state just described.

The instance is trivially satisfiable because the Π_{out} clauses lie outside the TACC and hence do not demand that the logical qudits they act on are in the state $|1\rangle$ at the end of the computation. Consequently, there is no need for a quantum computer to verify whether the circuit produces such results. □

A.2 Logical Qudits

Lemma 3.8 (No Π_{prop} clause on shared qudit)**.** *If a logical qudit is present in a Π_{prop} clause of a TACC and this qudit is also acted on by a clause from a different TACC, the instance is unsatisfiable.*

Proof. To begin, recall that the unique satisfying state of a TACC is the history state corresponding to the circuit expressed by the chain. This state is pure. If we suppose that the chain has length $T \geq 1$, the history state is given by $|\psi_{hist}\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |\psi_t\rangle \otimes |C_t\rangle$, where we use the shorthand $|C_t\rangle$ to represent the legal states of the clock register, and $|\psi_t\rangle := U_t \dots U_0 |0\rangle^{\otimes q}$. Notice that due to our choice of universal gate set, when a logical qudit is acted on by a unitary for the first time, the state of the qudit necessarily changes. That is, if we suppose a unitary acts on a fresh logical qudit at time $\tau \in [T]$, then $|\psi_{\tau+1}\rangle \neq |\psi_\tau\rangle$, meaning that at time $\tau + 1$, the logical and clock register become entangled.

Now, consider an instance with two disjoint clock components C_1 and C_2, each with a respective set of logical qudits L_1 and L_2 with a non-empty intersection L_3. Furthermore, suppose that $C_1 \cup L_1$ is the component that surely acts with a unitary on a shared logical qudit. For the sake of contradiction, assume that $C_1 \cup L_1$ and $C_2 \cup L_2$ have satisfying history states. Under these conditions, L_3 must be entangled with C_1, implying that any subsystem including only one of L_3 or C_1 will inevitably be impure. $C_2 \cup L_2$ includes L_3 but not C_1, leading to the conclusion that this solution must be impure. This is a contradiction. □

Lemma 3.9 (No Π_{init} and Π_{out} clauses on shared qudit)**.** *If a logical qudit is acted on by a Π_{init} clause from a clock component and a Π_{out} clause from another clock component, the instance is unsatisfiable.*

Proof. Satisfying the Π_{init} clause demands that the logical qudit must be in the state $|0\rangle$ at the beginning of the computation of the first component. As there can't be any unitaries acting on the qudit, the logical qudit should remain fixed to this state. Therefore, the logical qudit can't satisfy the Π_{out} clause requiring that it is either $|1\rangle$ or $|?\rangle$. □

Lemma 3.10 (Either all Π_{init} or all Π_{out} clauses)**.** *If a logical qudit is present in clauses from multiple clock components but the clauses are either all Π_{init} or all Π_{out} clauses, there exists a state of the logical qudit that simultaneously satisfies these clauses.*

Proof. In the former case, the Π_{init} clauses can be satisfied by setting the clock qudit to the state $|0\rangle$. In the latter case, the Π_{out} clauses can be satisfied by setting the clock qudit to either $|1\rangle$ or $|?\rangle$. □

A.3 Simultaneous Propagation

Proposition 3.3. *Let $c_0, \dots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length, and suppose k distinct Π_{prop} clauses act on c_t and c_{t+1} with $0 \leq t < T$ and an arbitrary pair of logical qudits. The logical qudits may or may not be repeated. Let $U_{t+1,1}, \dots, U_{t+1,k}$ denote the unitaries associated with these k clauses, and $|\phi_t\rangle$ be the state of the logical register at time t, i.e. $|\phi_t\rangle = U_{t,1} \dots U_{0,1} |0\rangle^{\otimes q}$ with $U_{0,1} = I$. All k Π_{prop} clauses can be satisfied simultaneously if and only if $U_{t+1,i} |\phi_t\rangle = U_{t+1,j} |\phi_t\rangle$ for all $i, j \in [k]$.*

Proof. First, recall that as shown in Eq. (22), a Π_{prop} clause with associated unitary U can be satisfied by three distinct states: either $|\phi\rangle \otimes |rr\rangle$, $|\phi\rangle \otimes |dd\rangle$, or $2^{-1/2}(|\phi\rangle \otimes |ad\rangle + U|\phi\rangle \otimes |da\rangle)$. Here, $|\phi\rangle$ is an arbitrary state of the two logical qudits. As we have established in Proposition 3.1, all clock qudits of the TACC must be active at some point in time and so here we must consider this latter case.

($\Rightarrow$): Assume the k Π_{prop} clauses are satisfiable. Then, as just mentioned, the ith Π_{prop} clause must be satisfied by a state of the form $2^{-1/2}(|\phi_t\rangle \otimes |a_t d_{t+1}\rangle + U_{t+1,i} |\phi_t\rangle \otimes |d_t a_{t+1}\rangle)$. To simultaneously these clauses, it is then evident that $U_{t+1,i} |\phi_t\rangle = U_{t+1,j} |\phi_t\rangle$ for all $i, j \in [k]$.

($\Leftarrow$): Assume that $U_{t+1,i} |\phi_t\rangle = U_{t+1,j} |\phi_t\rangle$ for all $i, j \in [k]$. In other words, $U_{t+1,i} |\phi_t\rangle = |\phi_{t+1}\rangle$ for all $i \in [k]$. Then, the state $2^{-1/2}(|\phi_t\rangle \otimes |a_t r_{t+1}\rangle + |\phi_{t+1}\rangle \otimes |d_t a_{t+1}\rangle)$ satisfies all k Π_{prop} clauses. □

Claim 3.1. *Let $\Pi_{prop}^{(c_t, c_{t+1})}$ be a clause acting on clock qudits c_t and c_{t+1} with associated unitary $U_{t+1,j}$, where $j \in [k_{t,t+1}]$. Measuring the eigenvalue of this clause on the state $|\psi_{phist}\rangle$, is equivalent to applying circuit $\mathcal{C}$ of Fig. 7 to the state $|\phi_t\rangle$ and an ancilla qubit in the state $|0\rangle$. Here, $|\phi_t\rangle$ is the state of the data register at time t, i.e. $|\phi_t\rangle = U_{t,1} \dots U_{0,1} |0\rangle^{\otimes q}$ with $U_{0,1} = I$. Furthermore, this circuit can be simulated perfectly by a circuit $\mathcal{C}_\mathcal{G}$ that uses a constant number of gates from the set $\mathcal{G} = \{H, T, \text{CNOT}\}$ and an additional ancilla qubit.*

Proof. Before starting the proof, there is a clarification that needs to be made. The claim states that we measure the eigenvalue of projector Π_{prop} acting on clock qudits c_t and c_{t+1}, yet, it has never been explicitly defined. Really, we mean that we measure the eigenvalue of the Π_{work} projector within the O_{prop} operator. This is the only projector of interest as all other projectors have already been checked by the classical algorithm. Indeed, the privileged history state $|\psi_{phist}\rangle$ has a correct assignment of the clock, logical, and endpoint qudits, all of its auxiliary subspaces are entangled accordingly, all of the clauses within the chain are well-defined, and the clock register represents a legal state of the clock at all times. Π_{work} is the only term left that hasn't been verified.

To simplify things further, observe that since the projector is only concerned with two clock qudits $\Pi_{work}|\psi_{phist}\rangle = \Pi_{work}|\theta_{t,t+1}\rangle$, where $|\theta_{t,t+1}\rangle := 2^{-1/2}(|\phi_t\rangle \otimes |C_t\rangle + U_{t+1,j}|\phi_t\rangle \otimes |C_{t+1}\rangle)$. Thus, we only care about measuring the eigenvalue of Π_{work} in the state $|\theta_{t,t+1}\rangle$. Are these two paragraphs okay here?

To begin the proof, recall that the probability of measuring eigenvalue 1 of a projector Π_i on a state $|\psi\rangle$ is given by $\langle\psi|\Pi_i|\psi\rangle$ (see Eq. (2)). In our case, this probability is given by $\langle\theta_{t,t+1}|\Pi_{work}|\theta_{t,t+1}\rangle$. First, note that the action of the projector on the state is given by

$$\begin{aligned}\Pi_{work}|\theta_{t,t+1}\rangle &= \frac{1}{2\sqrt{2}}\left[|\phi_t\rangle \otimes |C_t\rangle + U_{t+1,1}|\phi_t\rangle \otimes |C_{t+1}\rangle - U_{t+1,j}|\phi_t\rangle \otimes |C_{t+1}\rangle - U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle \otimes |C_t\rangle\right] \\ &= \frac{1}{2\sqrt{2}}\left[(I - U^\dagger_{t+1,j}U_{t+1,1})|\phi_t\rangle \otimes |C_t\rangle + (U_{t+1,1} - U_{t+1,j})|\phi_t\rangle \otimes |C_{t+1}\rangle\right],\end{aligned} \tag{80}$$

and so the probability of measuring eigenvalue 1 is

$$\begin{aligned}\langle\theta_{t,t+1}|\Pi_{work}|\theta_{t,t+1}\rangle &= \frac{1}{4}\left[\langle\phi_t| I - U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle + \langle\phi_t| I - U^\dagger_{t+1,1}U_{t+1,j}|\phi_t\rangle\right] \\ &= \frac{1}{2}\Re(\langle\phi_t| I - U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle) \\ &= \frac{1}{2} - \frac{1}{2}\Re(\langle\phi_t| U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle).\end{aligned} \tag{81}$$

On the other hand, the action of circuit $\mathcal{C}$ on the state $|\phi_t\rangle$ yields

$$\begin{aligned}\mathcal{C}|\phi_t\rangle &= (XH \otimes I)\left[\frac{|0\rangle \otimes |\phi_t\rangle - |1\rangle \otimes U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle}{\sqrt{2}}\right] \\ &= \frac{1}{2}\left[|1\rangle \otimes (I - U^\dagger_{t+1,j}U_{t+1,1})|\phi_t\rangle + |0\rangle \otimes (I + U^\dagger_{t+1,j}U_{t+1,1})|\phi_t\rangle\right],\end{aligned} \tag{82}$$

from which it follows that the probability of measuring outcome "1" is

$$\begin{aligned}\Pr(\text{outcome } 1) &= \langle\phi_t|\mathcal{C}^\dagger(|1\rangle\langle 1| \otimes I)\mathcal{C}|\phi_t\rangle \\ &= \frac{1}{4}\left[\langle\phi_t|(I - U^\dagger_{t+1,1}U_{t+1,j})(I - U^\dagger_{t+1,j}U_{t+1,1})|\phi_t\rangle\right] \\ &= \frac{1}{4}\left[2 - \langle\phi_t| U^\dagger_{t+1,1}U_{t+1,j}|\phi_t\rangle - \langle\phi_t| U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle\right] \\ &= \frac{1}{2} - \frac{1}{2}\Re(\langle\phi_t| U^\dagger_{t+1,j}U_{t+1,1}|\phi_t\rangle).\end{aligned} \tag{83}$$

Since both methods produce the same statistics, we can say that they are equivalent.

Although the Hadamard gates originally belong to $\mathcal{G}$, the controlled version of $U_{t+1,1}$, $U_{t+1,j}$ and the Pauli-X gate do not. However, they can be decomposed perfectly into gates of this set. Indeed, observe that in the computational basis, the 3- or 2-local controlled unitaries can be written as $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$, and the X gate as $|0\rangle\langle 1| + |1\rangle\langle 0|$. In this form it becomes clear that the matrix elements are of the form $\frac{1}{2^s}(a + ib + \sqrt{2}c + i\sqrt{2}d)$ with $s \in \mathbb{N}$ and $a, b, c, d \in \mathbb{Z}$. Therefore, by the result of Giles and Selinger [38], these gates can be implemented perfectly with a constant number of gates from the set $\{H, T, \text{CNOT}\}$ if given access to an additional ancilla qubit. □

B Proofs of Lemmas: BQP_1 without monogamy

Lemma 4.1 (Single-type qudits)**.** *If a* 6*-dimensional qudit in the instance is acted on by both a* Π_C *and* Π_L *clause, the instance is unsatisfiable.*

Proof. The proof of Lemma 3.1 is still valid for this statement if we simply omit Π_E. □

Lemma 4.2 (Lack of an ACC)**.** *If a clock component does not have at least one* Π_{init} *and one* Π_{out} *clause, the sub-instance is satisfiable.*

Proof. Since the proof of Lemma 3.6 is agnostic to the direction of the Π_{prop} clauses, it is still valid for this statement. □

Proposition 4.1 (Initialization of logical qudits)**.** *Let* $c_0, \dots c_L$ *with* $L \geq 0$ *be a chain of clock qudits where* c_0 *is a qudit present in a* Π_{init} *clause, and* c_L *is a qudit present in a* Π_{out} *clause. Suppose the* Π_{prop} *clauses of the chain point in arbitrary directions and suppose that there might be other* Π_{init} *clauses acting on other clock qudits within the chain (i.e.* $c_0, \dots, c_L$ *need not be an ACC). A state where* c_0 *is not* $|a\rangle$ *in any basis state of the superposition is not a satisfying state.*

Proof. The proof is similar to that of Proposition 3.1 except that here, the Π_{prop} clauses within the chain may point in different directions, and there might be other Π_{init} clauses in the chain. As we will see, the latter does not influence the arguments below in any way.

For the sake of contradiction, assume the satisfying state cannot have clock qudit c_0 in the state $|a\rangle$. We now show that all other assignments of the clock qudits result in either a violation of the assumption or a violation of a clause in the sub-instance.

To begin, observe that c_0 cannot be $|r\rangle$ as this would violate Π_{start}. Thus, c_0 must be permanently fixed to $|d\rangle$. Next, let us show that under this assumption, there can be no undefined clause.

Suppose that $\Pi_{prop}^{(c_t,c_{t+1})}$ with $0 \leq t \leq T-1$ is the first undefined clause of the chain and points in an arbitrary direction. As we have just established, if $c_t = c_0$, it cannot be in the active state. Then, for a different c_t, there must be a chain of clock qudits $c_0, \dots, c_t$ connected by well-defined Π_{prop} clauses. Observe that if any of these qudits are in an active state, satisfying the Π_{prop} clauses (regardless of their direction) requires this state to shift towards the left until c_0 becomes $|a\rangle$, contradicting the initial assumption. This chain thus cannot have any active states. Then, since c_0 must be in state $|d\rangle$ and the qudits within this chain cannot be $|a\rangle$, they must all be $|d\rangle$ in order to satisfy the $\Pi_{clock,D}$ clauses. However, this assignment includes c_t, and as seen in Eq. (23), $|d_t \mathrm{x}_{t+1}\rangle$ does not satisfy $\Pi_{clock,?}^{(c_t,c_{t+1})}$ for any state of the qudit c_{t+1}. This shows that there can be no undefined clause.

Finally, the only other possibility is that the ACC only contains well defined clauses. Given that the ACC has at least one Π_{init} and a one Π_{out} clause, there must exist be an active state somewhere. Observe that because the clauses are well-defined, to satisfy the Π_{prop} clauses, this state is again free to shift towards the left until c_0 becomes active. Therefore, we can conclude that the assumption cannot hold.

We remark that the extra Π_{init} clauses within the chain can only set the constraints that their clock qudits cannot be in state $|r\rangle$. This does not influence any of the arguments above. □

Lemma 4.3 (Π_{prop} clauses acting on c_0 point outward)**.** *Consider a clock qudit present in a* Π_{init} *clause of a clock component. If this qudit is also present in a* Π_{prop} *clause (either well-defined or undefined) that points towards it, the instance is unsatisfiable.*

Proof. To show this, first recall that by Proposition 4.1 this clock qudit must be in state $|a\rangle$ at some point in time. Then, as shown in Eq. (23), there is no state that satisfies an undefined clause if the successor qudit is in state $|a\rangle$.

On the other hand, if the Π_{prop} clause is well-defined, satisfying this clause requires the active state to shift towards the right. Then, to keep a valid state of the clock, c_0 must become $|r\rangle$ which is forbidden by the Π_{start} clause. □

Proposition B.1. *Let* $c_0, \dots, c_T$ *with* $1 < T \leq L$ *be the clock qudits of a TACC of non-zero length. If a qudit* c_j *with* $0 < j < T$ *has more than one well-defined successor or predecessor, the instance is unsatisfiable.*

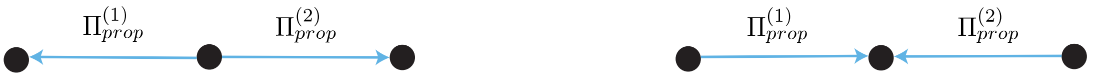


Figure 14: Left: A clock qudit in a chain with two successors. Right: A clock qudit in a chain with two predecessors.

Proof. Fig. 14 illustrates these two cases. Since c_j is part of a TACC, there will be a time where it is $|a\rangle$. Without loss of generality, let us assume that this is the initial state of the clock. Then, in the case where the qudit has two successors, the only valid starting state of the clock is $|\dots rar \dots\rangle$. Then, observe that the state that satisfies $\Pi^{(1)}_{prop}$, given by $|\dots rar \dots\rangle + |\dots adr \dots\rangle$, violates $\Pi^{(2)}_{clock}$ since the second basis state has a $|d\rangle$ preceded by $|r\rangle$. For the case where the qudit has two predecessors, the valid starting state of the clock is $|\dots dad \dots\rangle$. In a similar way as before, the state that satisfies $\Pi^{(1)}_{prop}$ is $|\dots dad \dots\rangle + |\dots ard \dots\rangle$, which has a troublesome term where $|r\rangle$ is preceded by $|d\rangle$. □

Lemma 4.4 (One well-defined Π_{prop} clause maximum)**.** *Consider a clock qudit present in a Π_{init} clause and a well-defined Π_{prop} clause. If this qudit is also present in another Π_{prop} clause (either well-defined or undefined) that connects to a new clock qudit, the instance is unsatisfiable.*

Proof. Proposition 4.1 states that this clock qudit must be in the state $|a\rangle$ at some point in time. Satisfying the well-defined Π_{prop} clause then requires the clock qudit to become $|d\rangle$. If we suppose the second Π_{prop} clause is undefined, then this already presents a violation of the undefined clause since as can be seen in Eq. (23), there is no satisfying state where the clock qudit is in this state. On the other hand, if the second Π_{prop} clause is also well-defined, the clock qudit has two successors and can be shown to be unsatisfiable by Proposition B.1. □

Proposition 4.2 (Unique active state in a TACC)**.** *Let $c_0, \dots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. A state in which at any given time there are two clock qudits c_a and c_b with $0 \leq a < b \leq T$ in the state $|a\rangle$ is not a satisfying state.*

Proof. Depending on how the Π_{prop} clauses are oriented in the instance, there are many cases in which such a state already violates numerous Π_{clock} clauses. Let us assume that the Π_{prop} clauses are oriented in such a way that the state doesn't immediately violate any Π_{clock} clause.[37] Then, given that the Π_{prop} clauses within the chain are well-defined, satisfying the Π_{prop} clauses in between qudits c_a and c_b by shifting the left active spot towards c_b implies that at some moment, we must reach a basis state of the form $|\dots a_{b-1} a_b \dots\rangle$. This state violates $\Pi^{(c_{b-1}, c_b)}_{clock}$. □

Lemma 4.5 (No forks in a TACC)**.** *Let $c_0, \dots, c_T$ with $1 < T \leq L$ be the clock qudits of a TACC with length at least two, and let c_j be a clock qudit that is not at the ends of the chain, i.e. $0 < j < T$. If c_j is connected to more than two other clock qudits, the instance is unsatisfiable.*

Proof. Since c_t represents a qudit within the TACC that is not situated at the ends of the chain, it must be connected to two other qudits via well-defined Π_{prop} clauses. As stated by the previous lemma, one of these qudits must serve as the predecessor, and the other as the successor. Then, depending on the direction of the third clause, c_t results in having either two successors or two predecessors. To show that this unsatisfiable, we must consider the cases where the clause could be well-defined or undefined. Proposition B.1 states that the former case is unsatisfiable. The latter requires a deeper analysis.

Let $c_{t'}$ refer to the other clock qudit of this undefined clause. Without loss of generality, assume that the starting state of the clock is when c_{t+1} is in state $|a\rangle$. Then, the state of the clock that satisfies the Π_{clock} clauses is $|\dots d_t a_{t+1} r_{t+1} \dots \mathrm{x}_{t'}\rangle$ where we have appended the state of qudit $c_{t'}$ at the end of the clock register. This already poses a problem since $|d_t \mathrm{x}_{t'}\rangle$ is not in the null space of the undefined $\Pi^{(c_t, c_{t'})}_{prop}$ clause for any $\mathrm{x} \in \{r, a, d\}$. □

[37] An example of such instances can be seen in Fig. 14. Suppose the TACC only consists of these three clock qudits and let the left-most and right-most qudits be in the state $|a\rangle$. Then, $|ada\rangle$ (for the left image) and $|ara\rangle$ (for the right image) do not immediately violate any Π_{clock} clause of the instance.

Lemma 4.6 (Unique direction of a TACC)**.** *Let $c_0, \dots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. If there exists a Π_{prop} clause within the TACC that points towards c_0, the instance is unsatisfiable.*

Proof. To begin, let us assume that $\Pi_{prop}^{(c_0,c_1)}$ is one of the clauses that points towards c_0. As shown in Proposition 4.1, c_0 must be $|a\rangle$ at some point in time, meaning that $|ad\dots\rangle$ must be a basis state in the superposition. Satisfying this clause requires the active state to shift towards the right, implying that c_0 must become $|r\rangle$ in order to maintain a valid state of the clock. This clock state violates the Π_{start} clause. Therefore, $\Pi_{prop}^{(c_0,c_1)}$ clause must point away from c_0.

Now, suppose that $\Pi_{prop}^{(c_t,c_{t+1})}$ with $1 < t \leq T$ is the first clause that points towards c_0. Observe that since $\Pi_{prop}^{(c_0,c_1)}$ must point away from c_0, and $\Pi_{prop}^{(c_t,c_{t+1})}$ is the first clause of the chain pointing towards this qudit, c_t must have two predecessors. By Proposition B.1, the instance is then unsatisfiable. □

Lemma 4.7 (Neighboring clauses of a non-zero length TACC)**.** *Let $c_0, \dots, c_T$ with $0 < T \leq L$ be the clock qudits of a TACC with non-zero length. Π_{prop} clauses that lie outside the TACC with the following properties result in an unsatisfiable instance:*

1. *If they are well-defined or undefined and include c_0.*
2. *If they are well-defined and include c_T.*
3. *If they are undefined, include c_T, and point towards it.*

Proof. Case (1) is trivial. By assumption, c_0 is present in a well-defined clause of the TACC that points away from it. This is the same setting as in Lemma 4.4, which concludes that the instance is unsatisfiable.

Consider case (2) and let c_{T+1} be the other qudit of the external Π_{prop} clause. Here, we have to show that for both possible directions of the clause, the instance is unsatisfiable. (I) Suppose that $\Pi_{prop}^{(c_T,c_{T+1})}$ is well-defined and points towards c_T. Recall that since the clauses of the TACC are all well-defined, we can assume that the clock begins in state $|d_0 \dots d_{T-1} a_T d_{T+1}\rangle$. Then, regressing this clock to satisfy $\Pi_{prop}^{(c_{T-1},c_T)}$ yields $|d_0 \dots d_{T-2} a_{T-1} r_T d_{T+1}\rangle$, which violates Π_{clock} as there is a $|r\rangle$ preceded by a $|d\rangle$. (II) If the clause points the other way, the state that satisfies $\Pi_{clock}^{(c_T,c_{T+1})}$ becomes $|d_0 \dots d_{T-1} d_T a_{T+1}\rangle$, which violates the Π_{stop} clause on c_T.

Consider case (3). Proposition 4.1 and Proposition 4.2 established that exactly one of the qudits in the TACC must be in an active state. Then, given that all Π_{prop} clauses within the TACC are well-defined, satisfying these clauses requires the active state to switch places towards the right until c_T eventually becomes $|a\rangle$. As shown in Eq. (23), there is no state that satisfies an undefined clause if the successor qudit is in state $|a\rangle$. Therefore, the instance must be unsatisfiable. □

Lemma 4.8 (Neighboring clauses of an active dot)**.** *Let c_d be an active dot. Π_{prop} clauses that act on c_d with the following properties result in an unsatisfiable instance:*

1. *If they are well-defined or undefined, and point towards c_d.*
2. *If they are well-defined and point away from c_d.*

Proof. The first case was already covered in Lemma 4.3. For the second case, recall that c_d must be $|a\rangle$ at all times. Satisfying the Π_{prop} clause requires the active state to shift towards the other clock qudit, leaving c_d to be in the state $|d\rangle$. If the active dot occurs because there is both a Π_{init} and Π_{out} clause acting on the qudit, having c_d as $|d\rangle$ causes a violation of the Π_{stop} clause. On the other hand, if the active dot occurs because the ACC is immediately truncated by an undefined clause, then this undefined Π_{prop} clause is infringed since, as shown in Eq. (23) there there is no state in the null space of the clause that has a $|d\rangle$ in the clock register. Write better. □

C Choice in Logical Qudits?

Now, why should we set the state of logical qudits in the instance since the very beginning (globally) and not for each individual sub-instance? To observe why this is, consider an instance where two clock components A and B share a single logical qudit l_x. Suppose A acts on l_x only with a Π_{init} clause. Then, suppose that B does not act on it with a Π_{init} clause, but acts on it with a Π_{prop} clauses stemming from its ACC. Let us look at the instance from the point of view of the individual clock components (unaware that l_x is also shared with another system). For component A, Proposition 3.1 demonstrates that this qudit must be in the state $|0\rangle$ at some point in time. Moreover, as there are no other clauses that change its state, we conclude that it must be fixed to $|0\rangle$. On the other hand, for component B, this qudit is unconstrained so it could be anything. Because of this freedom, it labels it as $|?\rangle$ so the component is trivial. Then, in a future step of the analysis of the instance where the shared logical qudits are analyzed, it becomes clear that there is an apparent conflict with this qudit as it cannot be in both states. As system A actually constrains this qudit, to satisfy this constraint, this qudit should be set to $|0\rangle$. This is more complicated than if we had originally considered both components as one and determined that the logical qudit should be set to $|0\rangle$ and then proceeded to evaluate the individual components.