On Efficiently Solvable Cases of Quantum k-SAT

Marco Aldi
Department of Mathematics and Applied Mathematics, Virginia Commonwealth University, Richmond, VA, USA
maldi2@vcu.edu

Niel de Beaudrap[1]
Department of Computer Science, University of Oxford, UK
niel.debeaudrap@cs.ox.ac.uk

Sevag Gharibian[2]
Department of Computer Science, University of Paderborn, Germany, and Virginia Commonwealth University, USA
sevag.gharibian@uni-paderborn.de

Seyran Saeedi
Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA
saeedis@vcu.edu

Abstract

The constraint satisfaction problems k-SAT and Quantum k-SAT (k-QSAT) are canonical NP-complete and QMA_1-complete problems (for $k \geq 3$), respectively, where QMA_1 is a quantum generalization of NP with one-sided error. Whereas k-SAT has been well-studied for special tractable cases, as well as from a parameterized complexity perspective, much less is known in similar settings for k-QSAT. Here, we study the open problem of computing satisfying assignments to k-QSAT instances which have a "matching" or "dimer covering"; this is an NP problem whose decision variant is trivial, but whose search complexity remains open.

Our results fall into three directions, all of which relate to the "matching" setting: (1) We give a polynomial-time classical algorithm for k-QSAT when all qubits occur in at most two clauses. (2) We give a parameterized algorithm for k-QSAT instances from a certain non-trivial class, which allows us to obtain exponential speedups over brute force methods in some cases by reducing the problem to solving for a single root of a single univariate polynomial. (3) We conduct a structural graph theoretic study of 3-QSAT interaction graphs which have a "matching". We remark that the results of (2), in particular, introduce a number of new tools to the study of Quantum SAT, including graph theoretic concepts such as transfer filtrations and blow-ups from algebraic geometry; we hope these prove useful elsewhere.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms, Theory of computation → Quantum complexity theory

Keywords and phrases search complexity, local Hamiltonian, Quantum SAT, algebraic geometry

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.38

Related Version A full version of this paper is available at `https://arxiv.org/abs/1712.09617`.

---

[1] NdB acknowledges support from the EPSRC National Quantum Technology Hub in Networked Quantum Information Processing.
[2] SG acknowledges support from NSF grants CCF-1526189 and CCF-1617710.

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 38; pp. 38:1–38:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding This work was partially supported by the EPSRC National Quantum Technology Hub in Networked Quantum Information Processing, and NSF grants CCF-1526189 and CCF-1617710.

Acknowledgements The first result of this project was partially completed while NdB was affiliated with the Centrum Wiskunde & Informatica; SG thanks Ronald de Wolf and Centrum Wiskunde & Informatica for their hospitality. SG thanks Howard Barnum and David Reeb regarding discussions on algebraic geometry, and David Gosset for discussions on Quantum SAT.

1 Introduction

Constraint satisfaction problems (CSPs) are cornerstones of both classical and quantum complexity theory. Indeed, CSPs such as 3-SAT and MAX-2-SAT are complete for NP [13], and their analogues Quantum 3-SAT (3-QSAT) and the 2-local Hamiltonian problem are QMA_1- and QMA-complete, respectively [3, 10, 16, 15]. (QMA is Quantum Merlin-Arthur, a quantum generalization of Merlin-Arthur, and QMA_1 is QMA with perfect completeness.) As such CSPs are intractable in the worst case, approaches such as approximation algorithms, heuristics, and exact algorithms are employed. In this paper, we focus on the latter technique, and ask: Which special cases of k-QSAT can be solved efficiently on a classical computer?

Unfortunately, this problem appears to be markedly more difficult than in the classical setting. For example, classically, if each clause c of a k-SAT instance can be matched with a unique variable v_c, then clearly the k-SAT instance is satisfiable, and finding a solution is trivial: Set variable v_c to satisfy clause c. (Note that the matching can be found efficiently via, e.g., the Ford-Fulkerson algorithm [11].) In the quantum setting, it has been known [17] since 2010 that k-QSAT instances with such "matchings" (also called a "dimer covering" in physics [17]) are also satisfiable, and moreover the satisfying assignment can be represented succinctly as a tensor product state. Yet, *finding* the satisfying assignment efficiently has proven elusive (indeed, the proof of [17] is *non-constructive*). In other words, we have a trivial NP *decision* problem whose analogous *search* version is not known to be efficiently solvable (see, e.g., [2] regarding the longstanding open question of decision versus search complexity for NP problems). This is the starting point of the present work.

Results and techniques. Our results fall under three directions, all of which are related to k-QSAT with matchings. For this, we first define Quantum k-SAT (k-QSAT) [3] and the notion of a system of distinct representatives (SDR). For k-QSAT, the input is a two-tuple $\Pi = (\{\Pi_i = |\psi_i\rangle\langle\psi_i|\}_i, \alpha)$ of rank 1 projectors or *clauses* $\Pi_i \in \mathcal{L}(\mathbb{C}^2)^{\otimes k}$, each acting non-trivially on a set of k (out of n) qubits, and non-negative real number $\alpha > 1/p(n)$ for some fixed polynomial p. The output is to decide whether there exists a satisfying assignment on n qubits $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$, i.e. to distinguish between the cases $\Pi_i|\psi\rangle = 0$ for all i (YES case), or whether $\langle\psi|\sum_i \Pi_i|\psi\rangle \geq \alpha$ (NO case). Note that k-QSAT generalizes k-SAT. As for a *system of distinct representatives (SDR)* (see, e.g., [12]), given a set system such as a hypergraph $G = (V, E)$, an SDR is a set of vertices $V' \subseteq V$ such that each edge in $e \in E$ is paired with a distinct vertex $v_e \in V'$ such that $v_e \in e$. In previous work on QSAT, an SDR has been referred to as a "dimer covering" [17].

1. Quantum k-SAT with bounded occurrence of variables. Our first result concerns the natural restriction of limiting the number of times a variable can appear in a clause. For example, 3-SAT with at most 3 occurrences per variable is NP-hard. We complement this as follows.

▶ **Theorem 1.** *There exists a polynomial time classical algorithm which, given an instance Π of k-QSAT in which each variable occurs in at most two clauses, outputs a satisfying product state if Π is satisfiable, and otherwise rejects. Moreover, the algorithm works for clauses ranging from 1-local to k-local in size.*

To show this, our idea is to "partially reduce" the k-QSAT instance to a 2-QSAT instance. We then use the transfer matrix techniques of [3, 18, 4] (particularly the notion of *chain reactions* from [4]), along with a new notion of "fusing" chain reactions, to deal with the remaining clauses of locality at least 3 in the instance.

Although this setting seems unrelated to the open question of computing solutions to k-QSAT instances with SDRs, we show the following. Denote the *interaction hypergraph* $G = (V, E)$ of a k-QSAT instance as a k-uniform hypergraph (i.e. all edges have size precisely k), in which the vertices correspond to qubits, and each clause c acting on a set of k qubits S_c, is represented by a hyperedge of size k containing the vertices corresponding to S_c.

▶ **Theorem 2.** *Let $G = (V, E)$ be a hypergraph with all hyperedges of size at least 2, and such that each vertex has degree at most 2. Then, G has an SDR.*

Thus, Theorem 1 resolves the open question of [17] for k-QSAT instances with SDRs in which (1) each variable occurs in at most two clauses and (2) there are no 1-local clauses. ((2) is necessary, as allowing edges of size 1 easily makes Theorem 2 false in general.)

2. On parameterized complexity for Quantum k-SAT. Our next result, and the main contribution of this paper, gives a parameterized algorithm[3] for explicitly computing (product state) solutions for a non-trivial class of k-QSAT instances. As discussed in Section 3, this algorithm in some cases provides an *exponential* speedup over brute force diagonalization.

At the core of the algorithm is a new graph theoretic notion of *transfer filtration of type* b for a k-uniform hypergraph $G = (V, E)$, for fixed $b > 0$. Intuitively, one should think of b as denoting the size of a set of b qubits which form the hard "foundation"' of any k-QSAT instance on G. With the notion of transfer filtration in hand, our framework for attacking k-QSAT can be sketched at a high level as follows.

1. First, given a k-QSAT instance Π on G with transfer filtration of type b, we "blow-up" Π to a larger, *decoupled* instance Π^+ (Decoupling Lemma, Lemma 9). The decoupled nature of Π^+ makes it "easier" to solve (Transfer Lemma, Lemma 17), in that any assignment to the b "foundation" qubits can be *extended* to a solution to all of Π^+. This raises the question – how does one map the solution of Π^+ back to a solution of Π?
2. We next give a set of "qualifier" constraints $\{h_s\}$ (Qualifier Lemma, Lemma 19) acting on *only the b foundation qubits*, with the following strong property: If a (product state) assignment $\mathbf{v}$ to the b foundation qubits satisfies the constraints $\{h_s\}$, then not only can we extend $\mathbf{v}$ via the Transfer Lemma to a full solution for Π^+ as in Step 1 above, but we can also map this extended solution back to one for the *original* k-QSAT instance Π.

Once the framework above is developed, we show that it applies to the non-trivial family of k-QSAT instances whose k-uniform hypergraph $G = (V, E)$ has a transfer filtration of type $b = |V| - |E| + 1$. This family includes, e.g., the semi-cycle, tiling of the torus, and "fir tree" (full version). Our main result (Theorem 23) says the following: For any k-QSAT instance Π on such a G and whose constraints are generic (see Section 3), computing a (product state) solution to Π reduces to solving for a root of a *single univariate* (see Remark 25) polynomial P – *any* such root (which always exists if the field $\mathbb{K}$ is algebraically closed) can then be extended back to a full solution for Π.

The key advantage of this approach, and what makes it a parameterized algorithm, is the following – the *degree* of P, and hence the runtime of the algorithm, scale exponentially only in b and a "radius" parameter r of the transfer filtration. Thus, given a transfer filtration

[3] Roughly, parameterized complexity characterizes the complexity of computational problems with respect to specific *parameters* of interest other than just the input size (e.g. the treewidth of the input graph).

where b and r are at most logarithmic, finding a (product state) solution to k-QSAT reduces to solving for a single root over $\mathbb{C}$ for a single univariate polynomial h_1 of polynomial degree, which can be done in polynomial time [25, 24]. Indeed, in Section 3 we give a non-trivial family of k-uniform hypergraphs, denoted Crash, for which our algorithm runs in polynomial time, whereas brute force diagonalization would require exponential time.

Conveniently, even when the foundation b and radius r are superlogarithmic, our algorithm still gives a *constructive* proof that all k-QSAT instances satisfying the preconditions of Theorem 23 have a (product state) solution. In particular, in Corollary 27, we observe that such hypergraphs must have SDRs, and so we constructively reproduce the result of [17] that any 3-QSAT instance with an SDR is satisfiable (by a product state) (again, assuming the additional conditions of Theorem 23 are met).

Finally, although this result stems primarily from tools of projective algebraic geometry (AG), the presentation herein avoids any explicit mention of AG terminology (with the exception of defining the term "generic" in Section 3.3) to be accessible to readers without an AG background. A brief overview of the ideas in AG terms is given in the full version.

3. A study of 3*-uniform hypergraphs with SDRs.* Our final contribution, which we hope guides future studies on the topic, is to take steps towards understanding the structure of all 3-QSAT instances with SDRs, particularly when $|E| = |V|$. Unfortunately, this seems a difficult task (if not potentially impossible, see "finite characterization" comments below). We first give various characterizations involving intersecting families (each pair of edges has non-empty intersection). We then study *linear* hypergraphs (each pair of edges intersects in at most one vertex), which are generally more complex. (For example, the set of edge-intersection graphs of 3-uniform linear hypergraphs is known not to have a "finite" characterization in terms of a finite list of forbidden induced subgraphs [19].) We study "extreme cases" of linear hypergraphs with SDRs, such as the Fano plane and "tiling of the torus", and in contrast to these two examples, demonstrate a (somewhat involved) linear hypergraph we call the iCycle which also satisfies the *Helly property* (which generalizes the notion of "triangle-free"). A main conclusion of this study is that even with multiple additional restrictions in place (e.g. linear, Helly), the set of 3-uniform hypergraphs with SDRs remains non-trivial. To complement these results, we show how to fairly systematically construct large linear hypergraphs with $|E| = |V|$ without SDRs. We hope this work highlights the potential complexity involved in dealing with even the "simple" case of 3-QSAT with SDRs.

Discussion, previous work and open questions. Regarding our parameterized algorithm, our notions of transfer filtrations and blow-ups apply to *any* instance of k-QSAT (and thus also[4] k-SAT), including QMA_1-complete instances. (For example, every k-uniform hypergraph has a trivial foundation obtained by iteratively removing vertices until the resulting set contains no edges. A key question is how *small* the foundation and radius of the filtration can be chosen for a given hypergraph, as our algorithm's runtime scales exponentially in these parameters.) More precisely, our techniques in Section 3, up to and including the Qualifier Lemma, apply to arbitrary k-QSAT instances. The main question is when *local* solutions to the qualifier constraints (which act only on b out of n qubits) can

[4] For the special case of k-SAT, note that it is not *a priori* clear that having a transfer filtration with a small foundation suffices to solve the system trivially. This is because the genericity assumption on constraints, which k-SAT constraints do not satisfy, is required to ensure that any assignment to the foundation propagates to all bits in the instance. Thus, the brute force approach of iterating through all 2^b assignments to the foundation does not obviously succeed.

be extended to *global* solutions to the entire k-QSAT instance. We answer this question affirmatively for the non-trivial class of k-QSAT instances which satisfy the preconditions of Theorem 23 (e.g. the semi-cycle, fir tree, crash, and any k-uniform hypergraph with $b = |V| - |E| + 1$), obtaining polynomial to exponential speedups over brute force in Section 3.

Moving to previous work, Quantum k-SAT was introduced by Bravyi [3], who gave an efficient (quartic time) algorithm for 2-QSAT, and showed that 4-QSAT is QMA_1-complete. Subsequently, Gosset and Nagaj [10] showed that Quantum 3-SAT is also QMA_1-complete, and independently and concurrently, Arad, Santha, Sundaram, Zhang [1] and de Beaudrap, Gharibian [4] gave linear time algorithms for 2-QSAT. The original inspiration for this paper was the work of Laumann, Läuchli, Moessner, Scardicchio and Sondhi [17], which showed *existence* of a product state solution for any k-QSAT instance with an SDR. Thus, the decision version of k-QSAT with SDRs is in NP and trivially efficiently solvable. However, whether the search version (i.e. compute an explicit satisfying assignment) is also in P remains open. The question of whether the decision and search complexities of NP problems are the same is a longstanding open problem in complexity theory; conditional results separating the two are known (see e.g. Bellare and Goldwasser [2]).

Regarding *classical* k-SAT, as mentioned above, in contrast to k-QSAT, solutions to k-SAT instances with an SDR can be trivially computed. As for parameterized complexity, classically it is a well-established field of study (see, e.g., [5] for an overview). The parameterized complexity of SAT and #SAT, in particular, has been studied by a number of works, such as [26, 6, 23, 7, 22, 21, 8], which consider parameterizations including based on tree-width, modular tree-width, branch-width, clique-width, rank-width, and incidence graphs which are interval bipartite graphs. Regarding parameterized complexity of *Quantum* SAT, as far as we are aware, our work is the first to initiate a "formal" study of the subject; however, we should be clear that existing works in Quantum Hamiltonian Complexity [20, 9] have long implicitly used "parameterized" ideas (e.g. in tensor network contraction, the bond dimension can be viewed as a parameter constraining the complexity of the contraction).

We close with open questions. Which ideas from classical parameterized complexity be generalized to the quantum setting? We develop a number of tools for studying Quantum SAT – can these be applied in more general settings, for example beyond the families of k-QSAT instances considered in Theorem 23? The "parameter" in our results of Section 3 involves the radius of a transfer filtration – whether a transfer filtration (of a fixed type b) of *minimum* radius can be computed efficiently, however, is left open for future work. Similarly, it is not clear that given $b \in \mathbb{N}$, the problem of deciding whether a given hypergraph G has a transfer filtration of type at most b is in P. We conjecture this latter problem is NP-complete. Finally, the question of whether solutions to arbitrary instances of k-QSAT with SDRs can be computed efficiently (recall they are guaranteed to exist [17]) remains open.

Organization. Section 2 gives an efficient algorithm for 3-QSAT with bounded occurrence of variables, and introduces the notion of transfer matrices (which are generalized via transfer functions in Section 3). Our main result is given in Section 3, and concerns a new parameterized complexity-type approach for solving k-QSAT. Our structural graph theoretic study of hypergraphs with SDRs, and any omitted proofs, are deferred to the full version.

Notation and basic definitions. For complex Euclidean space $\mathcal{X}$, $\mathcal{L}(\mathcal{X})$ denotes the set of linear operators mapping $\mathcal{X}$ to itself. For unit vector $|\psi\rangle \in \mathbb{C}^2$, the unique orthogonal unit vector (up to phase) is denoted $|\psi^\perp\rangle$, i.e. $\langle\psi|\psi^\perp\rangle = 0$.

▶ **Definition 3** (Hypergraph)**.** A *hypergraph* is a pair $G = (V, E)$ of a set V (vertices), and a family E (edges) of subsets of V. If each vertex has degree d, we say G is d-regular. When convenient we use $V(G)$ and $E(G)$ to denote the vertex and edge sets of G, respectively. We say G is k-uniform if all edges have size k.

▶ **Definition 4** (Cycle, Semicycle, Chain [14])**.** A k-uniform hypergraph $G = (V, E)$ is a *cycle* if there exists a sequence $S = (v_1, v_2, ..., v_l) \in V^l$ for $l \geq n$ such that (1) $v \in S$ for all $v \in V$, (2) for all $1 \leq i \leq l$, $e_i = \{v_i, v_{i+1}, ..., v_{i+k-1}\}$ are distinct edges in E, where indices are understood modularly. The length of the cycle G is $m = l$. If instead $1 \leq i \leq l - k + 1$ and $v_1 = v_l$ ($v_1 \neq v_l$), we obtain a *semicycle* (*chain*) of length $m = l - k + 1$.

2 Quantum SAT with bounded occurrence of variables

Transfer matrices, chain reactions, and cycle matrices. To study 3-QSAT with each qubit occurring in at most two constraints, we first recall transfer matrix tools from the study of 2-QSAT [3, 18, 4]. For any rank-1 constraint $\Pi_i = |\psi\rangle\langle\psi| \in \mathcal{L}((\mathbb{C}^2)^{\otimes k})$, consider Schmidt decomposition $|\psi\rangle = \alpha|a_0\rangle|b_0\rangle + \beta|a_1\rangle|b_1\rangle$, where $|a_i\rangle \in (\mathbb{C}^2)^{\otimes(k-1)}$ lives in the Hilbert space of the first $k-1$ qubits and $|b_i\rangle \in \mathbb{C}^2$ the last qubit. Then, the *transfer* matrix $T_\psi : (\mathbb{C}^2)^{\otimes k-1} \mapsto \mathbb{C}^2$ is given by $T_\psi = \beta|b_0\rangle\langle a_1| - \alpha|b_1\rangle\langle a_0|$. In words, given any assignment $|\phi\rangle$ to the first $k-1$ qubits, if $T_\psi|\phi\rangle \in \mathbb{C}^2$ is non-zero, then it is the *unique* assignment to qubit k (given $|\phi\rangle$ on qubits 1 to $k-1$) which satisfies Π_i.

In the special case of $k = 2$, transfer matrices are particularly useful. Consider first a 2-QSAT interaction graph (which is a 2-uniform hypergraph, or just a graph) $G = (V, E)$ which is a path, i.e. a sequence of edges $e_1 = (v_1, v_2), e_2 = (v_2, v_3), \ldots, e_m = (v_{m-1}, v_m)$ for distinct $v_i \in V$, and where edge e_i corresponds to constraint $|\psi_i\rangle$. Then, any assignment $|\phi\rangle \in \mathbb{C}^2$ to qubit 1 *induces a chain reaction (CR)* in G, meaning qubit 2 is assigned $T_{\psi_1}|\phi\rangle$, qubit 3 is assigned $T_{\psi_2}T_{\psi_1}|\phi\rangle$, and so forth. If this CR terminates before all qubits labelled by V receive an assignment, which occurs if $T_{\psi_i}|\phi'\rangle = 0$ for some i, this means that constraint i (acting on qubits i and $i+1$) is satisfied by the assignment $|\phi'\rangle$ to qubit i alone, and no residual constraint is imposed on qubit $i+1$. Thus, the graph G is reduced to a path $e_{i+1}, \ldots, e_m$. In this case, we say the CR is *broken.* Note that if G is a path, then it is a satisfiable 2-QSAT instance with a product state solution.

Finally, consider a 2-QSAT instance whose interaction graph G is a cycle $C = (v_1, \ldots, v_{m+1})$ with m. Then, a CR induced on vertex v_1 with any assignment $|\psi\rangle \in \mathbb{C}^2$ will in general propagate around the cycle and impose a consistency constraint on v_1. Formally, denote $T_C = T_{\psi_m} \cdots T_{\psi_1} \in \mathcal{L}(\mathbb{C}^2)$ as the *cycle matrix* of C. Then, if the cycle matrix is not the zero matrix, it be shown that the satisfying assignments for the cycle are precisely the eigenvectors of T_C. (If $T_C = 0$, any assignment on v_1 will only propagate partially around the cycle, thus decoupling the cycle into two paths.) Thus, if G is a cycle, it has a product state solution.

Here, when we refer to "solving the path or cycle", we mean applying the transfer matrix techniques above to efficiently compute a product state solution to the path or cycle.

k-QSAT with bounded occurence of variables. We now prove Theorem 1.

Proof of Theorem 1. We begin by setting terminology. Let Π be an instance of k-QSAT with k-uniform interaction graph $G = (V, E)$. For any clause c, let Q_c denote the set of qubits acted on c, i.e. Q_c is the edge in G representing c. We say c is *stacked* if Q_c is contained in another clause $Q_{c'}$, i.e. if $\exists c' \neq c$ such that $Q_c \subseteq Q_{c'}$. For a qubit v, we use shorthand $|v\rangle$ to denote the current assignment from $\mathbb{C}^2$ to v. For a clause c, $|c\rangle$ denotes the bad subspace of c, i.e. clause c is given by rank-1 projector $I - |c\rangle\langle c|$. The set of clauses vertex v appears in is denoted C_v. For any assignment $|v\rangle$, let $S_{|v\rangle} = \{\langle v|c\rangle \mid c \in C_v\} \subseteq \bigcup_{i=0}^{k-1} \mathbb{C}^{2^i}$, where recall c can be a clause on $1, \ldots, k$ qubits, and we assume $\langle v|$ acts as the identity on the qubits of c which are not v. Thus, $S_{|v\rangle}$ is the set of constraints we obtain by taking the clauses in C_v, and projecting down qubit v in each clause onto assignment $|v\rangle$ (i.e. clauses in $S_{|v\rangle}$ do not act on v). Our algorithm will satisfy that the only possible element of $\mathbb{C}$ in $S_{|v\rangle}$ is 0, obtained by projecting a constraint $|c\rangle \in \mathbb{C}^2$ onto its orthogonal complement to satisfy it; thus, assume without loss of generality that $S_{|v\rangle} \subseteq \bigcup_{i=1}^{k-1} \mathbb{C}^{2^i}$. Finally, two 1-local clauses $|c\rangle, |c'\rangle \in \mathbb{C}^2$ *conflict* if $|c\rangle$ and $|c'\rangle$ are linearly independent.

Algorithm A. Let Π satisfy the conditions of our claim. We repeatedly "partially reduce" Π to a 2-QSAT instance, and use the transfer matrix techniques outlined above to solve this subproblem. Combining this with a new notion of fusing CRs, the technique can be applied iteratively to reduce k-local constraints to 2-local ones until the entire instance is solved. Note: If a CR on a path is broken by a transfer matrix T_ψ on edge (u, v), i.e. $T_\psi|u\rangle = 0$, we implicitly continue by choosing assignment $|0\rangle$ on v to induce a new CR on the path.

1. While there exists a 1-local constraint c acting on some qubit v:
   a. If c conflicts with another 1-local clause on v, reject. Else, set $|v\rangle = |c^\perp\rangle \in \mathbb{C}^2$. Set[5] $C_v = S_{|v\rangle}$, and remove v from Π.
2. While there exists a qubit v appearing only in clauses of size at least $k' \geq 3$:
   a. Set $|v\rangle = |0\rangle$ and $C_v = S_{|v\rangle}$. Remove v from Π.
3. While there exists a 2-local clause:
   a. If there exists a stacked 2-local clause c, i.e. $c' \neq c$ such that $Q_c \subseteq Q_{c'}$:
      i. If $Q_c = Q_{c'}$, remove the qubits c acts on, and set their values to satisfy c and c'.
      ii. Else, $Q_c \subset Q_{c'}$. Thus, c' is k'-local for $3 \leq k' \leq k$. Set the values of the qubits in Q_c so as to satisfy c. This collapses c' to a $(k'-2)$-local constraint on $Q_{c'} \setminus Q_c$.
         A. If $k' - 2 = 1$, then c' has been collapsed to a 1-local constraint on some vertex $v \in Q_{c'} \setminus Q_c$, creating a path rooted at v. Set v so as to satisfy c', and use a CR to solve the resulting path until either the path ends, or a k''-local constraint is hit for $3 \leq k'' \leq k'$. In the latter case (Figure 1, Left), the k''-local constraint is reduced to a $(k''-1)$-local constraint and we return to the beginning of Step 3.
   b. Else, pick an arbitrary 2-local clause c acting on variables v_1 and v_2. Then, v_1 (v_2) is the start of a path h_1 (h_2) (e.g., Figure 1, Middle).
      i. If the path forms a cycle from v_1 to v_2, use the cycle matrix to solve the cycle. Remove the corresponding qubits and clauses from Π.
      ii. Else, set v_1 and v_2 so as to satisfy c. Solve the resulting paths h_1 (h_2) until a k'-local (k''-local) constraint l_1 (l_2) is hit for $3 \leq k' \leq k$ ($3 \leq k'' \leq k$). If both l_1 and l_2 are found:
         A. If $l_1 = l_2$ (i.e. $k' = k''$) and $k' - 2 = 1$, then fuse the paths h_1 and h_2 into a new path beginning at the qubit in l_1 which is not in h_1 or h_2 (Figure 1, Right). Iteratively solve the resulting path until a k'-local constraint is hit for $3 \leq k' \leq k$.
4. If any qubits are unassigned, set their values to $|0\rangle$.

In the full version, we prove correctness, run algorithm A on a sample input, and discuss its general applicability to an entire family of non-trivial 3-QSAT instances. ◀

[5] Note there is one "global copy" of each clause c that is "shared" by all C_v.

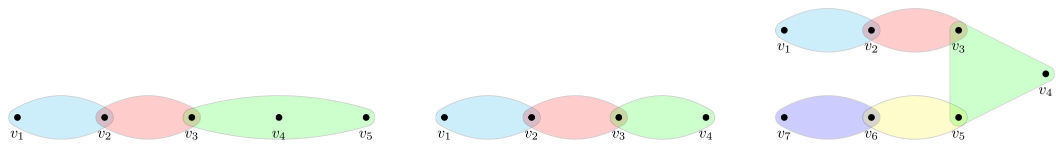

Figure 1 (Left) Solving the path rooted at v_1 via CR satisfies clauses (v_1, v_2) and (v_2, v_3), and projects clause (v_3, v_4, v_5) onto a 2-local residual clause on (v_4, v_5). The CR then stops. (Middle) Letting c denote the clause on (v_2, v_3), v_2 is the start of path $(v_2, v_1, \ldots)$, and v_3 is the start of path $(v_3, v_4, \ldots)$. (Right) Inducing CRs on v_1 and v_7, we assign values to v_3 and v_5. This collapses 3-local clause (v_3, v_4, v_5) into a 1-local clause on v_4 with a unique satisfying assignment, which induces a new CR starting at v_4. Thus, two CR's are "fused" into one CR.

3 Quantum SAT and parameterized algorithms

We next develop a parameterized algorithm for computing an explicit (product state) solution to a non-trivial class of k-QSAT instances (Theorem 23). Although the inspiration stems from algebraic geometry (AG), we generally avoid AG terminology to increase accessibility (see the full version for an overview in AG terms).

3.1 The transfer type of a hypergraph

▶ **Definition 5.** A hypergraph $G = (V, E)$ is of *transfer type* b if there exists a chain of subhypergraphs (denoted a *transfer filtration of type* b) $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_m = G$ and an ordering of the edges $E(G) = \{E_1, \ldots, E_m\}$ such that

1. $E(G_i) = \{E_1, \ldots, E_i\}$ for each $i \in \{0, \ldots, m\}$,
2. $|V(G_i)| \leq |V(G_{i-1})| + 1$ for each $i \in \{1, \ldots, m\}$,
3. if $|V(G_i)| = |V(G_{i-1})| + 1$, then $V(G_i) \setminus V(G_{i-1}) \subseteq E_i$,
4. $|V(G_0)| = b$, where we call $V(G_0)$ the *foundation*,
5. and each edge of G has at least one vertex not in $V(G_0)$.

In other words, a transfer filtration of type b builds up G iteratively by choosing b vertices as a "foundation", and in each iteration adding precisely one new edge E_i and *at most* one new vertex. If a new vertex is added in iteration i, condition (3) says it must be in edge E_i added in iteration i.

▶ **Example 6** (Running example)**.** We introduce a hypergraph G to serve as a running example in this section. Let $V(G) = \{1, 2, 3, 4\}$ with edges $E_1 = \{1, 2, 3\}$, $E_2 = \{1, 2, 4\}$, $E_3 = \{1, 3, 4\}$ and $E_4 = \{2, 3, 4\}$. By Definition 4, G is a 3-uniform cycle. Consider hypergraphs G_0, G_1, G_2, G_3 such that $V(G_0) = \{1, 2\}$, $V(G_1) = \{1, 2, 3\}$, $V(G_2) = V(G_3) = V(G_4) = V(G)$, $E(G_0) = \emptyset$ and $E(G_j) = \{E_1, \ldots, E_j\}$ for $j = 1, 2, 3$. Then $G_0 \subseteq G_1 \subseteq G_2 \subseteq G_3 \subseteq G_4 = G$ is a transfer filtration of type 2, G_2 is a chain, and G_3 is a semicycle.

▶ Remark 7. Let G be a hypergraph with transfer filtration $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_m = G$ of type b. Order the edges of G so that $E(G_i) = \{E_1, \ldots, E_i\}$ $\forall i \in \{1, \ldots, m\}$. Since by construction each edge contains at least one vertex not in $V(G_0)$, there exists a function $r : \{1, \ldots, m\} \to \{0, \ldots, m-1\}$ such that $r(i) < i$ and $|E_i \setminus V(G_{r(i)})| = 1$ for all $i \in \{1, \ldots, m\}$.

▶ **Example 8** (Running example)**.** Let G be the 3-uniform cycle of Example 6. Then one can choose $r : \{1, 2, 3, 4\} \to \{0, 1, 2, 3\}$ with $r(1) = r(2) = 0$, $r(3) = 1$ and $r(4) = 1$.

As the first step in our construction, we show how to map any k-uniform hypergraph G of transfer type b to a new k-uniform hypergraph G' of transfer type b whose transfer filtration

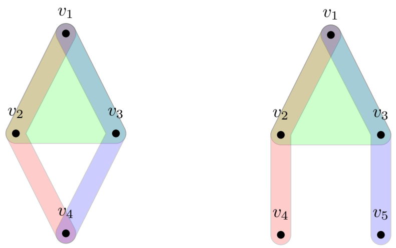

Figure 2 For the hypergraph on the left, consider the transfer filtration with foundation $G_0 = \{v_1, v_2\}$, and we iteratively add edges $\{v_1, v_2, v_3\}$, $\{v_1, v_2, v_4\}$, and $\{v_1, v_3, v_4\}$. The Decoupling Lemma maps this hypergraph to the one on the right, decoupling the intersection on vertex v_4. The surjective function p "undoes" the decoupling by mapping v_1, v_2, v_3 to themselves, and v_4, v_5 to v_4.

must add a vertex in each step (this follows directly from the relationship between $|V(G)|$ and $|E(G)|$ below). This has two effects worth noting: First, G' is guaranteed to have an SDR. Second, it *decouples* certain intersections in the hypergraph, as illustrated in Figure 2. For clarity, in the lemma below, for a function p acting on vertices, we implicitly extend its action to edges in the natural way, i.e. if $e = (v_1, v_2, v_3)$ then $p(e) = (p(v_1), p(v_2), p(v_3))$.

▶ **Lemma 9** (Decoupling lemma). *Given a k-uniform hypergraph G of transfer type b, there exists a k-uniform hypergraph $\widetilde{G}$ of transfer type b with $|E(G)| + b$ vertices and a surjective function $p : V(\tilde{G}) \to V(G)$ such $p(\widetilde{E}) \in E(G)$ for every $\widetilde{E} \in E(\widetilde{G})$.*

Proof. (Sketch) Let $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_m = G$ be a transfer filtration such that $V(G_0) = \{1, \ldots, b\}$, $E(G_i) = \{E_1, \ldots, E_i\}$ for every $i \geq 1$ and let $r : \{1, \ldots, m\} \to \{0, \ldots, m-1\}$ as in Remark 7. By Remark 7, there is a surjection $p : \{1, \ldots, m+b\} \to \{1, \ldots, n\}$ such that $p(i) = i$ for all $i \in \{1, \ldots, b\}$ and $\{p(i)\} = E_{i-b} \setminus V(G_{r(i-b)})$ for all $i \in \{b+1, \ldots, b+m\}$. For each $j \in \{1, \ldots, m+b\}$, let $\underline{j} = \min(p^{-1}(p(j)))$ and $\widetilde{E}_i = \{i+b\} \cup \{\underline{j} \mid j \in p^{-1}(E_i \setminus p(i+b))\}$ for each $i \in \{1, \ldots, m\}$. Setting $V(\widetilde{G_i}) = \{1, \ldots, b\}$ and $E(\widetilde{G_i}) = \{\widetilde{E}_1, \ldots, \widetilde{E}_i\}$ for each $i = \{0, \ldots, m\}$ we obtain a transfer filtration $\widetilde{G}_0 \subseteq \widetilde{G}_1 \subseteq \cdots \subseteq \widetilde{G}_m = \widetilde{G}$ of type b satisfying the requirements of the claim. ◀

▶ **Example 10** (Running example). Let G be the 3-uniform cycle of Example 6. The proof of Lemma 9 (full version) produces a 3-uniform hypergraph $\widetilde{G}$ with vertices $\{1, 2, 3, 4, 5, 6\}$ and edges $\widetilde{E}_1 = \{1, 2, 3\}$, $\widetilde{E}_2 = \{1, 2, 4\}$, $\widetilde{E}_3 = \{1, 3, 5\}$, $\widetilde{E}_4 = \{2, 3, 6\}$, and surjective function $p : \{1, 2, 3, 4, 5, 6\} \to \{1, 2, 3, 4\}$ defined by $p(1) = 1$, $p(2) = 2$, $p(3) = 3$, $p(4) = p(5) = p(6) = 4$. This choice is not unique: setting $\widetilde{E}_4 = \{2, 4, 6\}$ and $p(6) = 3$ also satisfies Lemma 9.

One of the "parameters" in our parameterized approach will be the *radius* of a transfer filtration, defined next. The concept is reminiscent of radii of graphs, and roughly measures "how far" an edge is from the foundation of b vertices with respect to the filtration.

▶ **Definition 11** (Radius of transfer filtration). Let G be a hypergraph admitting a transfer filtration $G_0 \subseteq \cdots \subseteq G_m = G$ of type b. Consider the function (whose existence is guaranteed by Remark 7) $r : \{0, \ldots, m\} \to \{0, \ldots, m-1\}$ such that $r(0) = 0$ and $r(i)$ is the smallest integer such that $|E_i \setminus V(G_{r(i)})| = 1$ $\forall i \in \{1, \ldots, m\}$. The *radius of the transfer filtration $G_0 \subseteq \cdots \subseteq G_m = G$ of type b* is the smallest integer β such that $r^\beta(i) = 0$ for all $i \in \{1, \ldots, m\}$ (r^β denotes composition of r with itself β times). The *type b radius of G* is the minimum value $\rho(G, b)$ of β over the set of all possible transfer filtrations of type b on G.

▶ **Example 12** (Running example). For G the 4-cycle from Example 6, since function r described in Example 8 is non-constant and $r(r(i)) = 0$ for all $i \in \{1, 2, 3, 4\}$, the transfer filtration of Example 6 has radius $\beta = 2$.

3.2 The main construction

Let W be a two dimensional vector space over a field $\mathbb{K}$. To discuss k-local constraints and product state solutions to k-QSAT instances, we now set up somewhat more general terminology than is standard in the literature. While this level of generality is natural given the geometric nature of our construction, for simplicity one may set $\mathbb{K} = \mathbb{C}$ and identify W with $\mathbb{C}^2$ if desired.

▶ **Definition 13.** A function $H_i : W^n \to \mathbb{K}$ is *k-local* if there exists a subset $E_i = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ and a nonzero functional $H_i^* : W^{\otimes k} \to \mathbb{K}$ such that $H_i(v_1, \dots, v_n) = H_i^*(v_{i_1} \otimes \dots \otimes v_{i_k})$ for all $v_1, \dots, v_n \in W$, i.e. H_i acts non-trivially only on a subset of k indices. A collection $H = (H_1, \dots, H_m)$ of k-local functions $H_1, \dots, H_n : W^n \to \mathbb{K}$ is *k-local*. The corresponding subsets E_i (i.e. on which H_i acts non-trivially) are the edges of a hypergraph G_H with vertices $\{1, \dots, n\}$, the *interaction graph of H*. The *product satisfiability set* of k-local collection H is the set $\mathcal{S}_H$ of all $(v_1, \dots, v_n) \in W^n$ such that $v_i \neq 0$ for all $i \in \{1, \dots, n\}$ and $H_j(v_1, \dots, v_n) = 0$ for all $j \in \{1, \dots, m\}$.

▶ Remark 14. Consider an isomorphism $\sharp$ between W and its dual $W^\vee$ that to each $v \in W$ assigns a functional $v^\sharp \in W^\vee$ such that $v^\sharp(v) = 0$. For instance, if a basis $\{w_1, w_2\}$ for W is chosen then we may define $\sharp$ by setting $((a_1 w_1 + a_2 w_2)^\sharp)(b_1 w_1 + b_2 w_2) = a_1 b_2 - a_2 b_1$ for all $a_1, a_2, b_1, b_2 \in \mathbb{K}$. Given any $v_1, v_2 \in W$, $v_1^\sharp(v_2) = 0$ if and only if $\exists \lambda \in \mathbb{K}$ such that $\lambda v_2 = v_1$.

▶ **Definition 15.** For $N \in \mathbb{Z}^+$, the *Fibonacci numbers of order N* are the entries of the sequence $(F_r^{(N)})$ such that $F_r^{(N)} = F_{r-1}^{(N)} + \ldots + F_{r-N}^{(N)}$ for all $r \geq N$, $F_{N-1}^{(N)} = 1$ and $F_r^{(N)} = 0$ for all $r \leq N - 2$. Note that there exists [27] a monotonically increasing sequence (ψ_N) with values in the real interval $[1, 2)$ such that, for each $N \geq 1$, $F_r^{(N)} \sim \psi_N^r$ as $r \to +\infty$.

▶ **Definition 16.** A function f on W^l with values in a $\mathbb{K}$-vector space has *degree* $(d_1, \dots, d_l)$ if $f(\lambda_1 v_1, \dots, \lambda_l v_l) = \lambda_1^{d_1} \cdots \lambda_l^{d_l} f(v_1, \dots, v_l)$ for every $\lambda_1, \dots, \lambda_l \in \mathbb{K}$ and every $v_1, \dots, v_l \in W$.

Applying the Decoupling Lemma to an input k-uniform hypergraph G with transfer type b, we obtain a k-uniform hypergraph $\widetilde{G}$ of type b with $m = n - b$, for m and n the number of edges and vertices, respectively. The next lemma shows that $\widetilde{G}$ is "easier to solve", in that any global (product) solution to the k-QSAT system can be derived from a set of assignments to the b foundation vertices, and conversely, any (product) assignment to the latter can be extended to a global (product) solution.

▶ **Lemma 17** (Transfer Lemma). *Let $H = (H_1, \dots, H_{n-b})$ be a k-local collection of functions $H_i : W^n \to \mathbb{K}$ whose interaction graph is a k-uniform hypergraph of transfer type b. There exist non-zero (non-constant) functions, "transfer functions" $g_1, \dots, g_n : W^b \to W$, s.t.:*

1. *(Global to local assignments) If $(v_1, \dots, v_n) \in \mathcal{S}_H$ (recall $v_i \neq 0$ by definition of $\mathcal{S}_H$) there exist nonzero $\lambda_1, \dots, \lambda_n \in \mathbb{K}$ such that, $\forall i \in \{1, \dots, n\}$, $\lambda_i v_i = g_i(v_1, \dots, v_b)$.*
2. *(Local to global assignments) For any nonzero $v_1, \dots, v_b \in W$ there exist $v_{b+1}, \dots, v_n \in W$ such that $(v_1, \dots, v_n) \in \mathcal{S}_H$ and $v_i = g_i(v_1, \dots, v_b)$ for every i such that $g_i(v_1, \dots, v_b) \neq 0$.*
3. *(Degree bounds) g_i has degree $(d_{i1}, \dots, d_{ib})$ such that $d_{ij} \leq F_i^{(b)}$ for all $j \in \{1, \dots, b\}$.*

Proof. (Sketch) We sketch the proof in the case $b = 2$ and $k = 3$. Define $g_1(v_1, v_2) = v_1$ and $g_2(v_1, v_2) = v_2$. Assume $G_0 \subseteq G_1 \subseteq \dots \subseteq G_{n-2} = G_H$ is a transfer filtration of type b, $V(G_i) = \{1, \dots, i + 2\}$ for all $i \in \{1, \dots, n - 2\}$. Assume $E(G_i) = \{E_1, \dots, E_i\}$ with

$E_i = \{i, i_1, i_2\}$ for some $i_1, i_2 < i$. We construct transfer functions inductively as follows. First define $(g_i^\sharp(v_1, v_2))(v) = H_{i-2}^*(g_{i_1}(v_1, v_2) \otimes g_{i_2}(v_1, v_2) \otimes v)$ for all $v_1, v_2, v \in W$. Then, given an isomorphism $\sharp$ between W and $W^\vee$ as in Remark 14, define $g_i : W^2 \to W$ such that $(g_i(v_1, v_2))^\sharp = g_i^\sharp(v_1, v_b)$ for all $v_1, v_2 \in W$. The properties of transfer functions stated in the lemma are proved by straightforward induction. We leave the details to the reader. ◀

▶ **Example 18** (Running example). Let $H = (H_1, H_2, H_3, H_4)$ be a 3-local collection of functions $H_i : W^6 \to \mathbb{K}$ whose interaction graph is the 3-uniform chain $\widetilde{G}$ described in Example 10 (obtained by plugging the 4-cycle G of Example 6 into the Decoupling Lemma). For clarity, H_i is defined on hyperedge $\widetilde{E}_i$, where the order of vertices in each edge is fixed by the transfer filtration chosen; in particular, use ordering $\widetilde{E}_1 = (1, 2, 3)$, $\widetilde{E}_2 = (1, 2, 4)$, $\widetilde{E}_3 = (1, 3, 5)$, $\widetilde{E}_4 = (2, 4, 6)$ with foundation $\{1, 2\}$. The proof of Lemma 17 constructs transfer functions $g_1, \ldots, g_6 : W^2 \to W$ which give assignments to qubits 1 through 6, respectively, as follows. Fixing a basis $\{w_1, w_2\}$ of W: $g_1(v_1, v_2) = v_1, g_2(v_1, v_2) = v_2, g_3(v_1, v_2) = H_1^*(v_1 \otimes v_2 \otimes w_2)w_1 - H_1^*(v_1 \otimes v_2 \otimes w_1)w_2, g_4(v_1, v_2) = H_2^*(v_1 \otimes v_2 \otimes w_2)w_1 - H_2^*(v_1 \otimes v_2 \otimes w_1)w_2, g_5(v_1, v_2) = H_3^*(v_1 \otimes g_3(v_1, v_2) \otimes w_2)w_1 - H_3^*(v_1 \otimes g_3(v_1, v_2) \otimes w_1)w_2, g_6(v_1, v_2) = H_4^*(v_2 \otimes g_4(v_1, v_2) \otimes w_2)w_1 - H_4^*(v_2 \otimes g_4(v_1, v_2) \otimes w_1)w_2$.

Thus far, we have seen how combining the Decoupling and Transfer Lemmas "blows up" an input k-QSAT system Π to a larger "decoupled" system Π^+ which is easier to solve due to its decoupled property. Now we wish to relate the solutions of Π^+ *back* to Π. This is accomplished by the next lemma, which introduces a set of "qualifier" constraints $\{h_s\}$ with the key property: Any solution to $\{h_s\}$ can be extended to one for Π^+, and then *mapped back* to a solution for Π. Importantly, the qualifier constraints *act only* on the b foundation vertices, as opposed to all n vertices!

▶ **Lemma 19** (Qualifier Lemma). *Let $H = (H_1, \ldots, H_m)$ be a k-local collection of functions $H_i : W^n \to \mathbb{K}$ whose interaction graph is a k-uniform hypergraph of transfer type b such that $m > n - b$. Then there exist non-zero (non-constant) functions, called* qualifiers, *$h_1, \ldots, h_{m-n+b} : W^b \to \mathbb{K}$ and $\pi : W^n \to W^b$ such that*
1. *$h_s(\pi(\mathcal{S}_H)) = 0$ for all $s \in \{1, \ldots, m - n + b\}$;*
2. *h_s has degree $(d_{s1}, \ldots, d_{sb})$ with $d_{sr} \le 2F^{(b)}_{\rho(G,b)+b+1}$ $\forall s \in [m + b]$ and $\forall r \in [b]$.*

Proof. (Sketch) We sketch the proof in the case $b = 2$. Given a transfer filtration $G_0 \subseteq \cdots \subseteq G_m = G_H$ of type 2 and radius $\rho(G_H, 2)$, the Decoupling Lemma yields a hypergraph $\widetilde{G_H}$ and a surjection p. Note that $\widetilde{G_H}$ is the interaction graph of a k-local collection $\widetilde{H} = (\widetilde{H}_1, \ldots, \widetilde{H}_m)$ of functions $\widetilde{H}_i : W^{m+2} \to \mathbb{K}$ such that $\widetilde{H}_i^* = H_i^*$ for each $i \in \{1, \ldots, m\}$. Let $\Delta : W^n \to W^{m+2}$ be such that $\Delta(v_1, \ldots, v_n) = (\widetilde{v}_1, \ldots, \widetilde{v}_{m+2})$, where $\widetilde{v}_i = v_{p(i)}$ for all $i \in \{1, \ldots, m + 2\}$. In particular $(v_1, \ldots, v_n) \in \mathcal{S}_H$ if and only if $\Delta(v_1, \ldots, v_n) \in \mathcal{S}_{\widetilde{H}}$. Applying the Transfer Lemma to $\widetilde{G_H}$ yields transfer functions $g_1, \ldots, g_{m+2} : W^2 \to W$. Borrowing notation from the proof of Lemma 9, let $\{i_1, \ldots, i_{m-n+2}\}$ be the subset of all $i \in \{1, \ldots, m+2\}$ such that $\underline{i} < i$. For each $s \in \{1, \ldots, m-n+2\}$, define qualifier $h_s : W^2 \to \mathbb{K}$ such that $h_s(v_1, v_2) = (g_{i_s}^\sharp(v_1, v_2))(g_{\underline{i_s}}(v_1, v_2))$ for all $v_1, v_2 \in W$. If $(v_1, \ldots, v_n) \in \mathcal{S}_H$, then for every $s \in \{1, \ldots, m - n + 2\}$ there exists $\lambda_{i_s}, \lambda_{\underline{i_s}} \in \mathbb{K}$ such that $\lambda_{i_s} v_{p(i_s)} = g_{i_s}(v_1, v_2)$ and $\lambda_{\underline{i_s}} v_{p(i_s)} = g_{\underline{i_s}}(v_1, v_2)$. Therefore $h_s(v_1, v_2) = \lambda_{i_s} \lambda_{\underline{i_s}} v^\sharp_{p(i_s)}(v_{p(i_s)}) = 0$ for every $s \in \{1, \ldots, m - n + 2\}$. Upon defining π as the composition of Δ with the projection onto the first two entries, this proves the first statement of the lemma. The second statement follows from the Transfer Lemma. ◀

▶ Remark 20. To recap, the construction in the proof of Lemma 19 implies that to solve the k-QSAT instance Π, we: (1) Apply the Decoupling Lemma to blow up Π to decoupled instance Π^+. (2) Apply the transfer functions from the Transfer Lemma to $v_1, \ldots, v_b$ to obtain a solution on all $m+b$ vertices for Π^+. Crucially, the qualifier constraints ensure that all decoupled copies of a vertex v receive the *same* assignment. (3) Map this solution back to one on n vertices for Π by "merging" decoupled copies of vertices.

▶ **Example 21** (Running example). Let $H = (H_1, H_2, H_3, H_4)$ be a 3-local collection of functions $H_i : W^4 \to \mathbb{K}$ whose interaction graph is the 3-uniform cycle of transfer type 2 from Example 6. If p is chosen as in Example 10, then the two qualifier functions are $h_1(v_1, v_2) = (g_5^\sharp(v_1, v_2))(g_4(v_1, v_2))$ of degree $(3,2)$ and $h_2(v_1, v_2) = (g_6^\sharp(v_1, v_2))(g_3(v_1, v_2))$ of degree $(2,3)$, where g_3, g_4, g_5, g_6 so that $d_{sr} \le 3 \le 10 = 2F_5^{(2)}$ for each $s, r \in \{1, 2\}$.

3.3 Generic constraints

Remark 20 outlined the high-level strategy for computing a (product-state) solution to an input k-QSAT system Π. For this strategy to work, however, we require an assignment to the foundation of the transfer filtration which (1) satisfies the qualifier functions from the Qualifier Lemma, and (2) causes the transfer functions g_i from the Transfer Lemma to output non-zero vectors. When are (1) and (2) possible? We now answer this question affirmatively for a non-trivial class of k-QSAT instances, assuming constraints are chosen generically.

▶ Remark 22 (Generic constraints). The set of k-local constraints H on k-uniform interaction hypergraph G is canonically identified with the projective variety $\mathcal{X}_G(\mathbb{K}) = (\mathbb{P}^{2^k-1}(\mathbb{K}))^m$. (See also [17].) We say a property holds for the *generic constraint with interaction graph* G if it holds for every k-local constraint on a Zariski open set of $\mathcal{X}_G(\mathbb{K})$. In the important case $\mathbb{K} = \mathbb{C}$, this implies in particular that such a property holds for almost all choices of constraints (with respect to the natural measure on $\mathcal{X}_G(\mathcal{C})$ induced by the Fubini-Study metric).

We now show the main theorem of this section (whose proof requires a few other definitions and a Surjectivity Lemma; see full version). The theorem applies to k-uniform hypergraphs of transfer type $b = n - m + 1$, which includes non-trivial instances such as the semi-cycle and the "fir tree" (full version). In words, the theorem says that for any k-uniform hypergraph of transfer type $b = n-m+1$ (i.e. there is one qualifier function h_1), if the constraints are chosen generically, then any zero of h_1 is the image under the map π (defined in Qualifier Lemma) of a satisfying assignment to the corresponding k-QSAT instance. The key advantage to this approach is simple: To solve the k-QSAT instance, instead of solving a system of equations, we are reduced to solving for the roots of just one polynomial – h_1. Moreover, if both the foundation size b and the radius of the transfer filtration of G are at most logarithmic in m and n, then h_1 has polynomial degree in m and n.

▶ **Theorem 23.** *Let $\mathbb{K}$ be algebraically closed, and let $\mathcal{F}$ denote the set of k-uniform hypergraphs with n vertices, m edges, and transfer type $b = n - m + 1$. If H is a generic k-local constraint with interaction graph $G \in \mathcal{F}$ and h_1 and π are as in the Qualifier Lemma (Lemma 19), then $(h_1 \circ \pi)^{-1}(0) \cap \mathcal{S}_H$ is nonempty.*

▶ **Example 24** (Running example). We illustrate the proof of Theorem 23 by specializing the construction to the 3-uniform semicycle G_3 from Example 6. Then $\widetilde{G}_3$ is the hypergraph with vertices $\{1, 2, 3, 4, 5\}$ and edges $\widetilde{E}_1 = \{1, 2, 3\}$, $\widetilde{E}_2 = \{1, 2, 4\}$, $\widetilde{E}_3 = \{1, 3, 5\}$. Moreover, the transfer functions $g_1, \ldots, g_5 : W^2 \to W$ can be chosen as in Example 18. Let h_1 be as in Example 21 and suppose $v_1, v_2 \in W$ are such that $h_1(v_1, v_2) = 0$. If none of the $g_i(v_1, v_2)$ are zero, then a solution of the form (v_1, v_2, v_3, v_4) can be found by Remark 20. Else, suppose (say) $g_3(v_1, v_2) = 0$ (generically, only one $g_i(v_1, v_2)$ will be zero in this case) so that v_3 is not constrained by v_1 and v_2. With respect to a fixed basis $\{w', w''\}$ of W, we need to show that v_3

can be chosen in such a way that (v_1, v_2, v_3, v_4), where (according the Transfer Lemma) $v_4 = H_2^*(v_1 \otimes v_2 \otimes w'')w' - H_2^*(v_1 \otimes v_2 \otimes w')w''$, is a solution. The idea is to modify $\widetilde{G_3}$ by removing the edge $\widetilde{E}_1$ and adding the vertex labeled by 3 to the foundation. With this modification, the Transfer Lemma yields $g_5(v_1, v_2, v_3) = H_3^*(v_1 \otimes v_3 \otimes w'')w' - H_3^*(v_1 \otimes v_3 \otimes w')w''$. By the Qualifier Lemma, we conclude that $g_5(v_1, v_2, v_3)$ is a non-zero multiple of v_4 if and only if $H_3^*(v_1 \otimes v_3 \otimes w')H_2^*(v_1 \otimes v_2 \otimes w'') - H_3^*(v_1 \otimes v_3 \otimes w'')H_2^*(v_1 \otimes v_2 \otimes w') = 0$. Introducing a coordinate $v_3 = w' + xw''$, this last condition is equivalent to the vanishing of a polynomial in x. While this particular example the polynomial is linear, it is in general of high degree and the assumption that $\mathbb{K}$ is algebraically closed is required in order to guarantee the existence of a root.

▶ Remark 25 (Reduction to univariate polynomials). Theorem 23 reduces us to solving a single polynomial equation, $h_1(v_1, \ldots, v_b) = 0$, which is multi-variate. In this case, we can reduce it further to a univariate polynomial by fixing arbitrary vectors $w_1, \ldots, w_b \in W$ and $w'_b \in W$ linearly independent from w_b. Then $P(x) = h_1(w_1, \ldots, w_b + xw'_b)$ is a *univariate* polynomial in $\mathbb{K}[x]$, which has a root $x \in \mathbb{K}$ since $\mathbb{K}$ is algebraically closed.

▶ Remark 26 (Runtimes, and complexity of solving for roots). By Theorem 23 and Remark 25, solving k-QSAT instances on hypergraphs in $\mathcal{F}$ with generic constraints reduces to solving for the roots of a single univariate polynomial, $P(x) \in \mathbb{K}[x]$. This can be accomplished by combining Theorem 2.7 of [25] and the algorithm of Schönhage [24] (Section 3.4 therein), which yields numerical approximations to all the roots of P within additive inverse exponential error in time exponential only in r and b. More specifically, in the full version, we give an explicit statement of the algorithm and a formal runtime analysis. We find k-QSAT instances with generic constraints and $b = n - m + 1$ require total time at most (for radius r, foundation size b, degree $d \leq 2^{r+b+2}$, m the number of constraints, n the number of qubits, $k \in \Theta(1)$ the locality of the constraints, and p a fixed polynomial which determines the additive accuracy $2^{-p(n)}$ to which we solve for roots of polynomials)

$$O(mn) + O\left(2^{2kb(r+b)}\right) + O\left(d^3 \log d + d^2 \log\left(9^d 2^{p(n)d}\right)\right) + O((r+b)2^{b(r+b+2)}). \qquad (1)$$

Thus, the algorithm is polynomial in m, n, and p, and exponential in k (the locality of the constraints), r (the radius), and b (foundation size).

Before discussing exponential speedups, we tie Theorem 23 back to SDRs:

▶ **Corollary 27.** *If G is a k-uniform hypergraph of transfer type $b = |V(G)| - |E(G)| + 1$, then G has an SDR.*

Thus, Theorem 23 constructively recovers the result of [17] (that any k-QSAT instance with an SDR has a (product-state) solution) in the case when the additional conditions of Theorem 23 are met (recall [17] works on all graphs with an SDR, but is not constructive). More generally, we can prove

▶ **Theorem 28.** *If G is a k-uniform hypergraph of transfer type $b \leq |V(G)| - |E(G)| + k - 1$, then G has an SDR.*

On exponential speedups via Theorem 23. Recall Theorem 23 applies to k-uniform hypergraphs of transfer type $b = n - m + 1$, such as the semicycle. From a parameterized complexity perspective, however, most interesting are hypergraphs for which the foundation size b and filtration radius r satisfy $b, r \in o(n + m)$, for which we might obtain an asymptotic speedup over brute force diagonalization of the Quantum SAT system (note the semicycle has $b \in \Theta(k), r \in \Theta(n)$). In the full version, we discuss the triangular tiling of the torus and

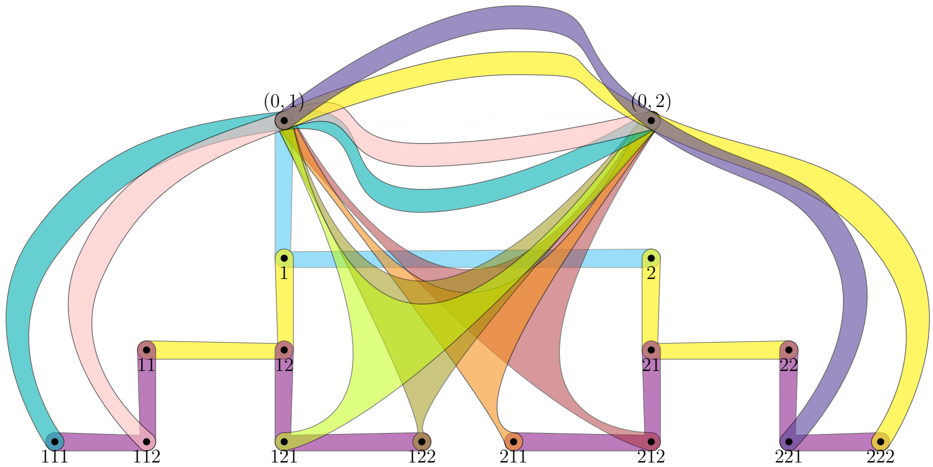

Figure 3 Depiction of 3-uniform crash hypergraph $C_{3,3}$. Generally, $C_{t,k}$ has an exponential separation between the filtration radius and foundation size versus number of vertices and edges.

the fir tree as examples with a quadratic separation $b, r \in \Theta(\sqrt{n} + \sqrt{m})$. (Note that for the runtime of Equation (1), however, a quadratic separation is unfortunately not enough for an asymptotic speedup.) Here, however, we give a hypergraph with a stronger, *exponential*, separation. Namely, we introduce the hypergraph *Crash* (Figure 3), with $r \in \Theta(t)$ and $b \in O(k)$, but $n, m \in \Theta((k-1)^t)$ for $k \geq 3$. On such hypergraphs, our parameterized algorithm hence runs in polynomial time, whereas brute force diagonalization would require time exponential in m and n.

We define k-uniform hypergraph family *Crash*, denoted $C_{t,k}$, as follows. For $k \geq 2$, let $\Sigma = \{1, 2, \ldots, k-1\}$. For $t \geq 1$, $C_{t,k}$ has vertices $V(C_{t,k}) = \bigcup_{j=0}^{t} V_j$ where $V_0 = \{(0, x) \,|\, x \in \Sigma\}$ and $V_j = \Sigma^{t-j+1}$ for all $1 \leq j \leq s$. The edge set of $C_{t,k}$ is the union of all edges of the following three forms:

1. For every $x \in V_1$, $E_x = \{x\} \cup V_0$;
2. for every $2 \leq j \leq t$ and every $x \in V_j$, $E_x = \{x\} \cup \{xa \,|, a \in \Sigma\}$;
3. $E_0 = \{(0, 1)\} \cup V_t$.

Then $C_{t,k}$ has a transfer filtration with foundation V_0 obtained by first adding all the edges E_x with $x \in V_1$, then adding all the edges E_x with $x \in V_2$, etc, with E_0 added last. This transfer filtration has radius t and type $k - 1 = |V(C_{n,k})| - |E(C_{n,k})| + 1$, whereas $|V(C_{n,k})|, |E(C_{n,k})| \in \Theta((k-1)^t)$ (full version).

References

1 Itai Arad, Miklos Santha, Aarthi Sundaram, and Shengyu Zhang. Linear Time Algorithm for Quantum 2SAT. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

2 M. Bellare and S. Goldwasser. The complexity of decision versus search. *SIAM J. Comput.*, 23(1):97–119, 1994.

3 S. Bravyi. Efficient algorithm for a quantum analogue of 2-SAT. Available at arXiv.org e-Print quant-ph/0602108v1, 2006.

4 Niel de Beaudrap and Sevag Gharibian. A Linear Time Algorithm for Quantum 2-SAT. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:21, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

5 Rodney G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Publishing Company, Incorporated, 2012.

6 E. Fischer, J.A. Makowsky, and E.V. Ravve. Counting truth assignments of formulas of bounded tree-width or clique-width. *Discrete Applied Mathematics*, 156(4):511–529, 2008. Third Haifa Workshop on Interdisciplinary Applications of Graph Theory, Combinatorics & Algorithm.

7 Robert Ganian, Petr Hliněný, and Jan Obdržálek. Better algorithms for satisfiability problems for formulas of bounded rank-width. *Fundam. Inf.*, 123(1):59–76, 2013.

8 Serge Gaspers, Christos H. Papadimitriou, Sigve Hortemo Sæther, and Jan Arne Telle. On satisfiability problems with a linear structure. In *IPEC*, 2016.

9 Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum Hamiltonian complexity. *Foundations and Trends® in Theoretical Computer Science*, 10(3):159–282, 2014.

10 D. Gosset and D. Nagaj. Quantum 3-SAT is QMA1-complete. In *Proceedings of the 54th IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 756–765, 2013.

11 L. R. Ford Jr. and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

12 S. Jukna. *Extremal Combinatorics With Applications in Computer Science*. Springer, second edition, 2011.

13 R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. New York: Plenum, 1972.

14 Gyula Y. Katona and Péter G.N. Szabó. Bounds on the number of edges in hypertrees. *Discrete Mathematics*, 339(7):1884–1891, 2016. 7th Cracow Conference on Graph Theory, Rytro 2014.

15 J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

16 A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002.

17 C. R. Laumann, A. M. Läuchli, R. Moessner, A. Scardicchio, and S. L. Sondhi. Product, generic, and random generic quantum satisfiability. *Physical Review A*, 81:062345, 2010.

18 C. R. Laumann, R. Moessner, A. Scardicchio, and S. L. Sondhi. Phase transitions and random quantum satisfiability. *Quantum Information & Computation*, 10:1–15, 2010.

19 Ranjan N. Naik, S.B. Rao, S.S. Shrikhande, and N.M. Singhi. Intersection graphs of k-uniform linear hypergraphs. *European Journal of Combinatorics*, 3(2):159–172, 1982.

20 T. J. Osborne. Hamiltonian complexity. *Reports on Progress in Physics*, 75(2):022001, 2012.

21 Daniel Paulusma, Friedrich Slivovsky, and Stefan Szeider. Model counting for cnf formulas of bounded modular treewidth. *Algorithmica*, 76(1):168–194, 2016.

22 Sigve Hortemo Sæther, Jan Arne Telle, and Martin Vatshelle. Solving #sat and maxsat by dynamic programming. *J. Artif. Int. Res.*, 54(1):59–82, sep 2015.

23 Marko Samer and Stefan Szeider. Algorithms for propositional model counting. *Journal of Discrete Algorithms*, 8(1):50–64, 2010.

24 A. Schönhage. Equation solving in terms of computational complexity. In *Proceedings of the International Congress of Mathematicians*, pages 131–153, 1986.

25 Arnold Schönhage. Quasi-GCD computations. *Journal of Complexity*, 1(1):118–137, 1985.

26 Stefan Szeider. *On Fixed-Parameter Tractable Parameterizations of SAT*, pages 188–202. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

27 D. A. Wolfram. Solving generalized fibonacci recurrences. *Fibonacci Quart.*, 36(2):129–145, 1998.