An SC-Square Approach to the Minimum Kochen–Specker Problem

Zhengyu Li[1], Curtis Bright[2] and Vijay Ganesh[3]

[1]University of Toronto, Canada
[2]University of Windsor, Canada
[3]University of Waterloo, Canada

Abstract
One of the most fundamental results in the foundations of quantum mechanics is the Kochen–Specker (KS) theorem, a 'no-go' theorem which states that contextuality is an essential feature of any hidden-variable theory. The theorem hinges on the existence of a mathematical object called a KS vector system, and its minimum size in three dimensions has been an open problem worked on by renowned physicists and mathematicians for over fifty years. We improved the lower bound on the size of a three-dimensional KS system from 22 to 23 with a significant speed-up over the most recent computational approach. Our approach combines the combinatorial search capabilities of satisfiability (SAT) solvers, the isomorph-free exhaustive generation capabilities of computer algebra systems (CASs), and the nonlinear real arithmetic solving capabilities of SAT modulo theory (SMT) solvers. Our work therefore fits directly into the *Satisfiability Checking and Symbolic Computation* (SC-Square) paradigm.

Keywords
Satisfiability solving, symbolic computation, symmetry breaking, isomorph-free generation, Kochen–Specker systems

1. Introduction

Hidden-variable theory aims to model quantum phenomenons by speculating the existence of a theory with unobservable degrees of freedom. The formulation of such a theory has been attempted by many accomplished physicists, including Einstein, Podolsky, and Rosen [1], and Bohm [2]. However, there are theorems describing observable properties of quantum mechanics under assumptions of either locality or noncontextuality, asserting that any hidden-variable theory is subject to several constraints. Two examples of such theories are Bell's theorem [3] and the Kochen–Specker theorem [4]. Bell's theorem states that, given the principle of locality, certain predictions of quantum mechanics using any hidden-variable theory are incorrect. The Kochen–Specker (KS) theorem states that, given the principle of noncontextuality, it is impossible to assign values to all physical observables consistently.

A KS System is a set of 3-dimensional vectors that proves the KS theorem by demonstrating

SC[2] 2022: The 7th International Workshop on Satisfiability Checking and Symbolic Computation, August 12, 2022
brianli.li@mail.utoronto.ca (Z. Li); cbright@uwindsor.ca (C. Bright); vijay.ganesh@uwaterloo.ca (V. Ganesh)
https://404briannotfound.tech/ (Z. Li); http://curtisbright.com (C. Bright); https://ece.uwaterloo.ca/~vganesh/ (V. Ganesh)
0000-0002-0462-625X (C. Bright); 0000-0002-6029-2047 (V. Ganesh)

contextuality. In this paper, we refer to the KS problem as the problem of finding the minimum size of a three-dimensional KS system. Finding the minimum KS system not only has historical significance, but also paves the way for near-future applications in quantum information processing [5]. So far, the complexity of known KS systems has prevented physicists from using them for any application. Finding the minimum 3-dimensional KS system reduces its intrinsic complexity, and could enable applications in security of quantum cryptographic protocols based on complementarity [6], zero-error classical communication [7], and dimension witnessing [8].

Boolean satisfiability (SAT) is one of the most influential problems in computer science and mathematics, as it has been studied intensively since it was shown to be NP-complete [9]. Over the last two decades, the design and implementation of conflict-driven clause learning (CDCL) SAT solving algorithms has enabled the solution of instances with millions of variables [10]. Even more surprisingly, SAT solvers frequently outperform special-purpose algorithms designed for software engineering [11], verification [12], and AI planning [13].

Despite these fantastic achievements, SAT solvers struggle on certain problems such as those containing symmetries [14] or those requiring the usage of mathematical theories more advanced than propositional logic [15]. Much work has been done to remedy these drawbacks, including the development of sophisticated symmetry breaking techniques [16, 17] and the development of solvers that support richer logics [18] ("SAT modulo theories" or SMT solvers). However, the mathematical support of SMT solvers is quite primitive when compared with the vast mathematical functionality available in a modern computer algebra system (CAS). A new kind of solving methodology [19] was developed in 2015 that harnesses SAT solving in addition to the efficient mathematical algorithms of CASs [20, 21]. This "SAT+CAS" solving methodology has since been successfully applied to many diverse problems, including circuit verification [22, 23], automatic debugging [24], finding circuits for matrix multiplication [25], computing directed Ramsey numbers [26], finding sequences and matrices with special properties [27, 28], and solving Lam's problem from projective geometry [29]. In this paper, we use the SAT+CAS solving methodology to dramatically improve the performance of searching for KS systems when compared to an out-of-the-box SAT solver or an out-of-the-box CAS.

Our work provides a new lower bound on the size of a three-dimensional KS system and discovers missing candidates from Uijlen and Westerbaan's search [30] for KS systems of size 20. We search for KS systems using a SAT solver coupled with computer algebraic routines (to remove symmetry from the search) and an SMT solver (to solve nonlinear real systems). The approach is motivated by the observation that a great number of properties that a KS system must satisfy can be converted into Boolean logic.

2. Background

In quantum mechanics, spin is an intrinsic form of angular momentum carried by elementary particles. Its existence can be concluded by the Stern–Gerlach experiment [31]. In this context, a spin-1 particle is shot through a fixed inhomogeneous magnetic field and continues non-deflected, deflects up, or deflects down. Along any given axes or directions of measurement, the spin-1 particle has 3 possible angular momentum states, namely 0, 1, and -1. Thus, the squared result of such measurements along any direction is always 0 or 1. The **SPIN axiom** states that given three pairwise orthogonal directions of measurement, the squared spin components of a spin-1 particle are 1, 0, 1 in these three directions. Thus, the observable corresponding to the question “is the squared spin 0?” measured in three mutually orthogonal directions will always produce *yes* (or 1) in exactly one direction and *no* (or 0) in the other two directions. The SPIN axiom follows from the postulates of quantum mechanics and is experimentally verifiable [32].

A KS vector system can be represented in multiple ways—we describe it as a finite set of points on a sphere. As a consequence of the SPIN axiom, the squared-spin measurements along opposite directions must yield the same outcome, and we can restrict the domain to the northern hemisphere. To define a KS vector system, we formally define a vector system and the notion of 010-colourability.

Definition 1. *A **vector system** is a finite subset of the closed northern hemisphere.*

Definition 2. *A vector system is **010-colourable** if there exists an assignment of 0 and 1 to each vector such that:*

1. *No two orthogonal vectors are assigned 1.*
2. *Three mutually orthogonal vectors are not all assigned to 0.*

Definition 3. *A Kochen–Specker vector system is a vector system that is not 010-colourable.*

Exhibiting the existence of a KS vector system proves the KS theorem, which states that there is no function from the closed northern hemisphere to $\{0, 1\}$ that satisfies the 010-property. Each KS vector system has a corresponding KS graph, defined as follows.

Definition 4. *For a vector system $\mathcal{K}$, define its **orthogonality graph** $G_{\mathcal{K}} = (V, E)$, where $V = \mathcal{K}$, $E = \{\, (v_1, v_2) : v_1, v_2 \in \mathcal{K}$ and $v_1 \cdot v_2 = 0 \,\}$.*

A *KS graph* is the orthogonality graph of a KS system. Essentially, the vertices of $G_{\mathcal{K}}$ are the vectors in $\mathcal{K}$, and there exists an edge between two vertices if and only if their corresponding vectors are orthogonal. We can also translate the notion of 010-colourability from a vector system to a graph.

Definition 5. *A graph G is **010-colourable** if there is a $\{0, 1\}$-colouring of the vertices such that the following two conditions are satisfied simultaneously:*

1. *Adjacent vertices are not both coloured 1.*
2. *For each triangle in G, there is exactly one vertex that is coloured 1.*

It is not guaranteed that there is a corresponding vector system for an arbitrary graph. If a graph does have a corresponding vector system, we say that this graph is embeddable.

Definition 6. *A graph $G = (V, E)$ is **embeddable** if it is a subgraph of an orthogonality graph $G_{\mathcal{K}}$ for some vector system $\mathcal{K}$.*

Essentially, being embeddable implies the existence of a vector system $\mathcal{K}$ whose vectors have a one-to-one correspondence with the vertices of G in such a way that adjacent vertices are mapped to orthogonal vectors. It is not necessary for non-adjacent vertices to go to non-orthogonal vectors by the definition above, though it is necessary for distinct vertices to be mapped to distinct vectors. An example of a unembeddable graph would be the cycle graph of order 4, as the orthogonality constraints would force a pair of opposite vertices of C_4 to be mapped to the same point. A KS graph must be both embeddable and non-010-colourable. Every KS system corresponds to a KS graph, allowing us to translate a problem on KS systems into a problem on KS graphs.

Throughout the years, renowned mathematicians and physicists such as Roger Penrose, Asher Peres, and John Conway have attempted to find a minimum three-dimensional KS system. The current smallest known KS system contains 31 vectors and was discovered by John Conway and Simon Kochen around 1990 [33]. This was communicated to Peres [34], who found a more symmetric system of 33 vectors [35]. Shortly later, Penrose [36, 37] found another system of 33 vectors. In 2011, Arends, Ouaknine, and Wampler [38] proved several properties that any KS graph must have and applied them to computationally prove that a KS system must contain at least 18 vectors. Seven years later, Uijlen and Westerbaan [30] showed that a KS system must have at least 22 vectors. This computational effort used around 300 CPU cores for three months and relied on the nauty software package [39] to exhaustively search for KS vector systems. Pavičić, Merlet, McKay, and Megill [40] have also shown that a KS system in which each vector is part of a mutually orthogonal triple must have at least 30 vectors. However, despite these extensive searches, the gap between the lower and upper bounds remains significant and the minimum size of a 3-dimensional KS system remains unknown.

3. SAT Encoding of the KS Problem

A KS vector system $\mathcal{K}$ can be converted into a KS graph $G_{\mathcal{K}}$. Each vector in $\mathcal{K}$ can be assigned to a vertex in $G_{\mathcal{K}}$, so that if two vectors are orthogonal, then their corresponding vertices are connected. Therefore, to find a KS vector system, it is sufficient to find a Kochen–Specker graph. A KS graph is minimal if the only subgraph that is a KS graph is itself. Arends, Ouaknine, Wampler [38] proved that a minimal three-dimensional KS graph must satisfy the following properties:

1. The graph must not contain a subgraph isomorphic to C_4.
2. Each vertex of the graph must have minimum degree 3.
3. Every vertex is part of a triangle graph C_3.

We will encode these three properties above and the non-010-colourability of the KS graph in conjunctive normal form (CNF) in order to search for KS graphs using a SAT solver. If the solver produces solutions, these solutions are equivalent to graphs satisfying all four properties. A simple undirected graph of order n has $\binom{n}{2}$ potential edges, and we will represent each edge as a Boolean variable. The edge variable e_{ij} will be true exactly when vertices i and j are connected where $1 \leq i < j \leq n$. For convenience, we let both e_{ij} and e_{ji} denote the same variable. We also use the $\binom{n}{3}$ triangle variables t_{ijk} denoting that distinct vertices i, j, and k are mutually connected. In Boolean logic this is expressed as $t_{ijk} \leftrightarrow (e_{ij} \wedge e_{ik} \wedge e_{jk})$ which in conjunctive normal form is expressed via the four clauses $\neg t_{ijk} \vee e_{ij}$, $\neg t_{ijk} \vee e_{ik}$, $\neg t_{ijk} \vee e_{jk}$, and $\neg e_{ij} \vee \neg e_{ik} \vee \neg e_{jk} \vee t_{ijk}$. Again, the indices i, j, and k of the variable t_{ijk} may be reordered arbitrarily for notational convenience.

Encoding the Squarefree Constraint To encode the property that a KS graph must be squarefree, we construct encodings that prevent the existence of any possible squares in the graph. Three squares can be formed on four vertices. Therefore, for each choice of four vertices i, j, k, l, we use the three clauses $\neg e_{ij} \vee \neg e_{jk} \vee \neg e_{kl} \vee \neg e_{li}$, $\neg e_{ij} \vee \neg e_{jl} \vee \neg e_{lk} \vee \neg e_{ki}$, and $\neg e_{il} \vee \neg e_{lj} \vee \neg e_{jk} \vee \neg e_{ki}$. By enumerating over all possible choices of four vertices and constructing the above CNF formula, we force the graph to be squarefree.

Encoding the Minimum Degree Constraint For each vertex i, to ensure that i is connected to at least three other vertices, we take each subset S of the set $\{1, \ldots, i-1, i+1, \ldots, n\}$ with cardinality $n-3$ and construct the clause $\bigvee_{j \in S} e_{ij}$. By enumerating over all such subsets we enforce a minimum degree of 3 on vertex i. Thus, constructing similar formulae for all vertices $1 \leq i \leq n$ forces any vertex in the graph to have a degree of at least 3.

Encoding the Triangle Constraint We now encode the property that every vertex is part of a triangle. For each vertex i, we require 2 other distinct vertices to form a triangle and there are $\binom{n-1}{2}$ possible triangles containing i. At least one of those triangles must be present in the graph and this is ensured by the clause $\bigvee_{j,k \in S} t_{ijk}$ where S is $\{1, \ldots, i-1, i+1, \ldots, n\}$ and $j < k$. Using this clause for each $1 \leq i \leq n$ ensures that every vertex is part of a triangle.

Encoding the Colourability Constraint We generate clauses to block as many 010-colourable graphs as possible (ideally all of them, leaving only the non-010-colourable graphs). A graph is non-010-colourable if and only if for all $\{0, 1\}$-colourings of the graph a pair of colour-1 vertices is connected or a set of three colour-0 vertices are mutually connected. The idea is to consider many $\{0, 1\}$-colourings and construct clauses that block the graphs for which those colourings form a 010-colouring.

For each $\{0, 1\}$-colouring, we have a set of colour-0 vertices V_0 and a set of colour-1 vertices V_1. Given a specific such colouring, the clause

$$\bigvee_{\substack{i,j \in V_1 \\ i<j}} e_{ij} \vee \bigvee_{\substack{i,j,k \in V_0 \\ i<j<k}} t_{ijk}$$

enforces that the colouring is not a 010-colouring of the graph since either a pair of colour-1 vertices is connected or a set of three colour-0 vertices is mutually connected. Due to the large number of possible $\{0, 1\}$-colourings, we only consider colourings with less than or equal to $\lceil \frac{n}{2} \rceil$ colour-1 vertices. Colourings with more than $\lceil \frac{n}{2} \rceil$ colour-1 vertices are unlikely to be 010-colourings and in practice were not useful in blocking 010-colourable graphs.

4. Embeddability Checking

We verify the embeddability of a KS graph using an SMT approach. We refer to the solutions generated by the SAT solver as *KS candidates.* If a KS candidate is embeddable, then it is a KS system. Our embeddability checking algorithm consists of two parts. The first part is a direct integration of Uijlen and Westerbaan's vector assignment algorithm [30], which finds all possible interpretations to describe the orthogonal relations between the vectors. We define *free vectors* as vectors that have not been fixed as the cross product of two vectors. Of all possible interpretations, we choose the one with the least number of free vectors, since such an assignment will likely be solved in the least amount of time. The second part of the algorithm applies an SMT solver to determine the satisfiability of an intepretation. An interpretation generated by Uijlen and Westerbaan's algorithm can be converted into a set of cross and dot product equations, and we pass these equations into the SMT solver Z3 [41].

1. If v_i is connected to v_j and v_k, then $V_i = (V_j \times V_k)$ or $V_i = (V_k \times V_j)$.
2. If v_i are v_j are not connected, then V_i is not collinear to V_j, and $V_i \times V_j \neq \vec{0}$.
3. If v_i and v_j are connected, then V_i and V_j are orthogonal, and $V_i \cdot V_j = 0$.

To check whether a graph is embeddable, we use the Z3 theorem prover to determine whether such a system of equations is satisfiable over the real numbers. Z3 applies a CDCL-style algorithm to determine the satisfiability of non-linear arithmetic constraints [42]. Given a system of equations, Z3 will attempt to find a solution for all variables. If a solution is found, it is an assignment of vertices to vectors that satisfies all orthogonality constraints and the graph is therefore embeddable.

5. Implementation

Directly solving a SAT instance with the above encodings is only feasible for smaller orders, since the number of graphs in the search space increases exponentially with the order. We implement two effective techniques to reduce runtime. One is an orderly generation technique that generates graphs in the search space up to isomorphism, and the other is a parallelization technique.

5.1. Orderly Generation

We use a hybrid SAT and isomorphic-free generation approach. First, we introduce the orderly generation approach, developed independently by Igor Faradžev [43] and Ronald Read [44] in 1978. It uses the following canonical representation of a graph.

Definition 7. *An adjacency matrix M is canonical if every permutation of its rows produces a matrix lexicographically greater or equal to M, where the lexicographical order is defined by concatenating the above-diagonal entries of the columns of the adjacency matrix in order.*

The *parent* of an $n \times n$ matrix A is the upper-left $(n-1) \times (n-1)$ submatrix of A. The orderly generation method is based on the following two consequences of Definition 7:

1. Every isomorphic class of graphs has only one canonical representative.
2. If a matrix is canonical, then its parent is also canonical.

Note that the second property implies that if a matrix is not canonical, then all of its children are not canonical. Therefore, we can reject all intermediate noncanonical matrices, as they will not lead us to a canonical matrix in the search tree and we only want to generate canonical matrices. Orderly generation works by recording intermediate canonical objects and iteratively extending them a row and column at a time until the matrices have been extended to a full canonical matrix.

In our SAT+CAS implementation, when the SAT solver finds an intermediate matrix the canonicity of this matrix is determined by a canonicity-checking routine implemented in C++ and the MathCheck system [45]. If the matrix is noncanonical then a blocking clause is learned which removes this matrix (and all of its children) from the search. Otherwise, the matrix is canonical and the SAT solver proceeds as normal. We also combine this process with the symmetry breaking clauses of Codish et al. that canonical matrices can be shown to satisfy [46, Def. 8].

We simplify the SAT instance using the SAT solver CaDiCaL [47] before solving the instance using MapleSAT [48]. As a preprocessing step, we also run the orderly generation process on graphs with up to 12 vertices and add the generated blocking clauses directly into the instance provided to CaDiCaL—this allows the simplification to incorporate some of the knowledge derived from the orderly generation process.

5.2. Parallelization

For orders above 20, parallelization is applied by dividing the instance into smaller subproblems using the cube-and-conquer approach [49]. The approach applies a lookahead solver [50] to partition a hard problem into many subspaces, and offers very efficient solving time for some combinatorial problems. During the splitting, the lookahead solver tries to find the next variable that will split the search space the most evenly. We use the lookahead solver March_cu [51]. Each splitting variable will be added to the SAT instance as a new unit clause, generating two subproblems (one with a positive unit clause and one with a negative unit clause) that can be solved in parallel. We terminate the cubing process when a significant number of edge variables have been fixed in each subproblem.

6. Results

Given the CNF file with the encoded constraints, we use the aforementioned techniques combined with the SAT+CAS approach to verify all previous results on KS systems up to order 21 and improve the best known lower bound with a significant speedup factor. All computations are done on Intel E5-2683 CPUs @ 2.1GHz administrated by Compute Canada. Table 1 summarizes our results: our computation[1] on order 21 is over 1000 times faster than the previous computational search of Uijlen and Westerbaan [30]. We apply cube-and-conquer and naive parallel SAT solving on order 21 and 22 due to the combinatorial explosion caused by the large order. We eliminate 75 edge variables from subproblems in order 21 and 90 edge variables in order 22 during the cubing process. Some cubes of order 22 with 90 edge variables eliminated define instances that are not solved within 72 CPU hours, so we perform additional cubing on these instances until at least 125 edge variables have been eliminated.

[1] We provide an easy-to-use open source repository (https://github.com/BrianLi009/PhysicsCheck) for readers to reproduce our results.

Order	Candidates	Simplification	Cubing	Cube Simplification	Solving
17	1	0.02 hrs	N/A	N/A	0.02 hrs
18	0	0.02 hrs	N/A	N/A	0.13 hrs
19	8	0.31 hrs	N/A	N/A	2.46 hrs
20	147	0.54 hrs	N/A	N/A	39.71 hrs
21	2,497	1.50 hrs	38 hrs	19.4 hrs	1,019 hrs
22	88,282	2.54 hrs	953.7 hrs	253.3 hrs	46,079 hrs

Table 1
A summary of our results in the Kochen–Specker problem on orders $17 \leq n \leq 22$.

All 90,935 KS candidates of order less than 23 are unembeddable, so a KS system must contain at least 23 vectors. We compared our Kochen–Specker candidates with Uijlen and Westerbaan's results, and have verified their results up to order 21—though we obtained fewer candidates for each order because Uijlen and Westerbaan did not require every vertex of a candidate to be part of a triangle. However, we found four additional KS candidates in order 20 that are not present in Uijlen and Westerbaan's collection, indicating their search was incomplete. We have verified that these four additional graphs satisfy the constraints of a KS candidate and therefore would be KS systems were they embeddable. Note that not all KS candidates we discovered are minimal. Some KS candidates of larger order contain a KS candidate of smaller order as a subgraph.

7. Conclusion

In this paper we improved the lower bound on the size of a minimum three-dimensional KS vector system, improved the efficiency of searching for KS systems by orders of magnitude, and found KS candidates not present in the previous result of Uijlen and Westerbaan [30]. Compared to previous work, our approach is less error-prone and provides robust results, since it reduces the need for custom-purpose search algorithms. The SC-Square paradigm has resolved a number of problems from combinatorics, number theory, and geometry that were not solvable using either SAT solvers or CAS alone, and was proven once again to be an effective approach for combinatorial problems. Using a completely new approach we made substantial progress on the long-standing open problem of determining the smallest possible KS system. With this work we extend the reach of the SAT+CAS paradigm, for the first time, to resolving combinatorial questions in the realm of foundations of quantum mechanics.

References

[1] A. Einstein, B. Podolsky, N. Rosen, Can quantum-mechanical description of physical reality be considered complete?, Physical review 47 (1935) 777. doi:10.1103/PhysRev.47.777.
[2] D. Bohm, B. J. Hiley, The undivided universe: An ontological interpretation of quantum theory, Routledge, 2006.

[3] J. S. Bell, J. S. Bell, Speakable and unspeakable in quantum mechanics: Collected papers on quantum philosophy, Cambridge university press, 2004. doi:10.1017/CBO9780511815676.

[4] S. Kochen, E. P. Specker, The Problem of Hidden Variables in Quantum Mechanics, Journal of Mathematics and Mechanics 17 (1967) 59–87. doi:10.1007/978-94-010-1795-4_17.

[5] G. Canas, M. Arias, S. Etcheverry, E. S. Gómez, A. Cabello, G. B. Xavier, G. Lima, Applying the simplest Kochen-Specker set for quantum information processing, Physical review letters 113 (2014) 090404. doi:10.1103/PhysRevLett.113.090404.

[6] A. Cabello, V. D'Ambrosio, E. Nagali, F. Sciarrino, Hybrid ququart-encoded quantum cryptography protected by Kochen-Specker contextuality, Physical Review A 84 (2011) 030302. doi:10.1103/PhysRevA.84.030302.

[7] T. S. Cubitt, D. Leung, W. Matthews, A. Winter, Improving zero-error classical communication with entanglement, Physical Review Letters 104 (2010) 230503. doi:10.1103/PhysRevLett.104.230503.

[8] O. Gühne, C. Budroni, A. Cabello, M. Kleinmann, J.-Å. Larsson, Bounding the quantum dimension with contextuality, Physical Review A 89 (2014) 062107. doi:10.1103/PhysRevA.89.062107.

[9] S. A. Cook, The complexity of theorem-proving procedures, in: Proceedings of the third annual ACM symposium on Theory of computing, 1971, pp. 151–158. doi:10.1145/800157.805047.

[10] V. Ganesh, M. Y. Vardi, On the unreasonable effectiveness of SAT solvers, 2020. URL: https://www.cs.rice.edu/~vardi/papers/SATSolvers21.pdf.

[11] C. Cadar, V. Ganesh, P. M. Pawlowski, D. L. Dill, D. R. Engler, EXE: Automatically generating inputs of death, ACM Transactions on Information and System Security (TISSEC) 12 (2008) 1–38. doi:10.1145/1455518.1455522.

[12] E. Clarke, A. Biere, R. Raimi, Y. Zhu, Bounded model checking using satisfiability solving, Formal methods in system design 19 (2001) 7–34. doi:10.1023/A:1011276507260.

[13] H. A. Kautz, B. Selman, et al., Planning as satisfiability, in: ECAI, volume 92, Citeseer, 1992, pp. 359–363. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.9443&rep=rep1&type=pdf.

[14] C. Bright, J. Gerhard, I. Kotsireas, V. Ganesh, Effective problem solving using SAT solvers, in: Maple in Mathematics Education and Research, volume 1125 of *Communications in Computer and Information Science*, Springer, Cham, 2020, pp. 205–219. doi:10.1007/978-3-030-41258-6_15.

[15] E. Ábrahám, G. Kremer, Satisfiability checking: Theory and applications, in: International Conference on Software Engineering and Formal Methods, Springer, 2016, pp. 9–23. doi:10.1007/978-3-319-41591-8_2.

[16] F. A. Aloul, K. A. Sakallah, I. L. Markov, Efficient symmetry breaking for Boolean satisfiability, IEEE Transactions on Computers 55 (2006) 549–558. doi:10.1109/TC.2006.75.

[17] F. A. Aloul, A. Ramani, I. L. Markov, K. A. Sakallah, Solving difficult SAT instances in the presence of symmetry, in: Proceedings 2002 Design Automation Conference (IEEE Cat. No. 02CH37324), IEEE, 2002, pp. 731–736. doi:10.1109/DAC.2002.1012719.

[18] C. Barrett, P. Fontaine, C. Tinelli, The Satisfiability Modulo Theories Library (SMT-LIB), www.SMT-LIB.org, 2016.

[19] E. Ábrahám, Building bridges between symbolic computation and satisfiability checking, in: Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, 2015, pp. 1–6. doi:10.1145/2755996.2756636.

[20] C. Bright, I. Kotsireas, V. Ganesh, SAT solvers and computer algebra systems: A powerful combination for mathematics, in: Proceedings of the 29th International Conference on Computer Science and Software Engineering, 2019, pp. 323–328. URL: https://dl.acm.org/doi/abs/10.5555/3370272.3370309.

[21] C. Bright, I. Kotsireas, V. Ganesh, When satisfiability solving meets symbolic computation, Communications of the ACM 65 (2022) 64–72. doi:10.1145/3500921.

[22] D. Kaufmann, A. Biere, M. Kauers, Verifying large multipliers by combining SAT and computer algebra, in: 2019 Formal Methods in Computer Aided Design (FMCAD), IEEE, 2019. doi:10.23919/fmcad.2019.8894250.

[23] A. Mahzoon, D. Große, C. Scholl, A. Konrad, R. Drechsler, Formal verification of modular multipliers using symbolic computer algebra and Boolean satisfiability, in: 59th Design Automation Conference (DAC), 2022. URL: https://www.informatik.uni-bremen.de/agra/doc/konf/Formal_Verification_of_Modular_Multipliers.pdf.

[24] A. Mahzoon, D. Große, R. Drechsler, Combining symbolic computer algebra and Boolean satisfiability for automatic debugging and fixing of complex multipliers, in: 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, 2018. doi:10.1109/isvlsi.2018.00071.

[25] M. J. H. Heule, M. Kauers, M. Seidl, New ways to multiply 3×3-matrices, Journal of Symbolic Computation 104 (2021) 899–916. doi:10.1016/j.jsc.2020.10.003.

[26] D. Neiman, J. Mackey, M. J. H. Heule, Tighter bounds on directed Ramsey number $R(7)$, 2020. doi:10.48550/ARXIV.2011.00683.

[27] C. Bright, I. Kotsireas, V. Ganesh, Applying computer algebra systems with SAT solvers to the Williamson conjecture, Journal of Symbolic Computation 100 (2020) 187–209. doi:10.1016/j.jsc.2019.07.024.

[28] C. Bright, I. S. Kotsireas, A. Heinle, V. Ganesh, Enumeration of complex Golay pairs via programmatic SAT, in: Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16–19, 2018, 2018, pp. 111–118. doi:10.1145/3208976.3209006.

[29] C. Bright, K. K. H. Cheung, B. Stevens, I. Kotsireas, V. Ganesh, A SAT-based resolution of Lam's problem, in: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, 2021, pp. 3669–3676. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16483.

[30] S. Uijlen, B. Westerbaan, A Kochen-Specker system has at least 22 vectors, New Generation Computing 34 (2016) 3–23. doi:10.1007/s00354-016-0202-5.

[31] W. Gerlāch, O. Stern, Der experimentelle nachweis der richtungsquantelung im magnetfeld, Zeitschrift für Physik 9 (1922) 349–352. doi:10.1007/BF01326983.

[32] Y.-F. Huang, C.-F. Li, Y.-S. Zhang, J.-W. Pan, G.-C. Guo, Experimental test of the Kochen-Specker theorem with single photons, Physical Review Letters 90 (2003). doi:10.1103/physrevlett.90.250401.

[33] J. Conway, S. Kochen, The geometry of the quantum paradoxes, in: Quantum [Un]speakables, Springer Berlin Heidelberg, 2002, pp. 257–269. doi:10.1007/

978-3-662-05032-3_18.
[34] A. Peres, Two simple proofs of the Kochen–Specker theorem, Journal of Physics A: Mathematical and General 24 (1991) L175–L178. doi:10.1088/0305-4470/24/4/003.
[35] A. Peres, Quantum theory: concepts and methods, Springer, 2002. doi:10.1007/0-306-47120-5.
[36] R. Penrose, On Bell non-locality without probabilities: some curious geometry, Quantum Reflections, Cambridge University Press, Cambridge (2000) 1–27. URL: https://people.maths.ox.ac.uk/lmason/Tn/34/TN34-09.pdf.
[37] J. Zimba, R. Penrose, On bell non-locality without probabilities: More curious geometry, Studies in History and Philosophy of Science Part A 24 (1993) 697–720. doi:10.1016/0039-3681(93)90061-n.
[38] F. Arends, J. Ouaknine, C. W. Wampler, On searching for small Kochen-Specker vector systems, in: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 2011, pp. 23–34. doi:10.1007/978-3-642-25870-1_4.
[39] B. D. McKay, A. Piperno, Practical graph isomorphism, II, Journal of Symbolic Computation 60 (2014) 94–112. doi:10.1016/j.jsc.2013.09.003.
[40] M. Pavičić, J.-P. Merlet, B. McKay, N. D. Megill, Kochen–Specker vectors, Journal of Physics A: Mathematical and General 38 (2005) 1577–1592. doi:10.1088/0305-4470/38/7/013.
[41] L. de Moura, N. Bjørner, Z3: An efficient SMT solver, in: International conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2008, pp. 337–340. doi:10.1007/978-3-540-78800-3_24.
[42] D. Jovanović, L. de Moura, Solving non-linear arithmetic, in: International Joint Conference on Automated Reasoning, Springer, 2012, pp. 339–354. doi:10.1007/978-3-642-31365-3_27.
[43] I. Faradžev, Constructive enumeration of combinatorial objects, in: Problèmes combinatoires et théorie des graphes, 1978, pp. 131–135.
[44] R. C. Read, Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations, in: Annals of Discrete Mathematics, volume 2, Elsevier, 1978, pp. 107–120. URL: https://doi.org/10.1016/S0167-5060(08)70325-X.
[45] C. Bright, V. Ganesh, A. Heinle, I. Kotsireas, S. Nejati, K. Czarnecki, MathCheck2: A SAT+CAS verifier for combinatorial conjectures, in: International Workshop on Computer Algebra in Scientific Computing, Springer, 2016, pp. 117–133. doi:10.1007/978-3-319-45641-6_9.
[46] M. Codish, A. Miller, P. Prosser, P. J. Stuckey, Constraints for symmetry breaking in graph representation, Constraints 24 (2019) 1–24. doi:10.1007/s10601-018-9294-5.
[47] A. Biere, K. Fazekas, M. Fleury, M. Heisinger, CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020, in: T. Balyo, N. Froleyks, M. J. H. Heule, M. Iser, M. Järvisalo, M. Suda (Eds.), Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions, volume B-2020-1 of *Department of Computer Science Report Series B*, University of Helsinki, 2020, pp. 51–53.
[48] J. H. Liang, V. Ganesh, P. Poupart, K. Czarnecki, Learning rate based branching heuristic for SAT solvers, in: Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5–8, 2016, Proceedings, 2016, pp. 123–140. doi:10.1007/978-3-319-40970-2_9.

[49] M. J. H. Heule, O. Kullmann, S. Wieringa, A. Biere, Cube and conquer: Guiding CDCL SAT solvers by lookaheads, in: Haifa Verification Conference, Springer, 2011, pp. 50–65. doi:10.1007/978-3-642-34188-5_8.

[50] M. J. H. Heule, H. van Maaren, Look-ahead based SAT solvers, Handbook of satisfiability 185 (2009) 155–184. doi:10.3233/FAIA200988.

[51] P. van der Tak, M. J. H. Heule, A. Biere, Concurrent cube-and-conquer, in: Theory and Applications of Satisfiability Testing – SAT 2012, Springer Berlin Heidelberg, 2012, pp. 475–476. doi:10.1007/978-3-642-31612-8_42.