MathCheck2: A SAT+CAS Verifier for Combinatorial Conjectures

Curtis Bright[1], Vijay Ganesh[1], Albert Heinle[1],
Ilias Kotsireas[2], Saeed Nejati[1], Krzysztof Czarnecki[1]

[1]University of Waterloo, [2]Wilfred Laurier University

July 2, 2016

To appear at CASC 2016 as an invited paper in the
Satisfiability Checking + Symbolic Computation (SC^2) Track

Motivation

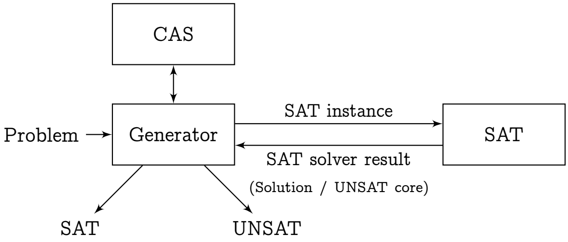
The research areas of SMT solving and symbolic computation are quite disconnected. On the one hand, SMT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining solving techniques, and developing dedicated heuristics, but its current focus lies on easier theories and it makes use of symbolic computation results only in a rather naive way.

Erica Ábrahám[1]

[1] Building bridges between symbolic computation and satisfiability checking. *Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation.*

The MathCheck2 System

- Uses SAT and CAS functionality to finitely verify or counterexample conjectures in mathematics.
- Used to study conjectures in combinatorial design theory about the existence of Hadamard and Williamson matrices.



Experimental Results

MathCheck2 was able to show that...

- Williamson matrices of order 35 do not exist.
  - First shown by Đoković[2], who requested an independent verification.
- Williamson matrices exist for all orders $n < 35$.
  - Even orders were mostly previously unstudied.
- Found over 160 Hadamard matrices which were not previously in the library of the CAS Magma.
  - Orders up to 168×168.

[2] Williamson matrices of order $4n$ for $n = 33, 35, 39$. *Discrete Mathematics.*

Hadamard matrices

- square matrix with ± 1 entries
- any two distinct rows are orthogonal

Hadamard matrices

- square matrix with ± 1 entries
- any two distinct rows are orthogonal

Example

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}$$

Conjecture

An $n \times n$ Hadamard matrix exists for any n a multiple of 4.

Naive Hadamard Encoding

Each entry of H will be represented using a *Boolean variable* encoding with $\mathrm{BV}(1) = \text{true}$ and $\mathrm{BV}(-1) = \text{false}$.

Multiplication becomes XNOR under this encoding, i.e.,

$$\mathrm{BV}(x \cdot y) = \neg(\mathrm{BV}(x) \oplus \mathrm{BV}(y)) \qquad \text{for } x, y \in \{\pm 1\}.$$

Naive Hadamard Encoding

Arithmetic formula encoding

$$\sum_{k=0}^{n-1} h_{ik} \cdot h_{jk} = 0 \qquad \text{for all } i \neq j.$$

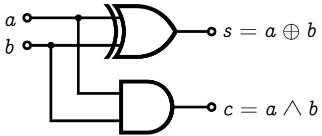
Boolean variable encoding

Using 'product' variables $p_{ijk} \coloneqq \mathrm{BV}(h_{ik} \cdot h_{jk})$ this becomes the cardinality constraints

$$\sum_{\substack{k=0 \\ p_{ijk} \text{ true}}}^{n-1} 1 = \frac{n}{2} \qquad \text{for all } i \neq j.$$

Naive Hadamard Encoding

A *binary adder* consumes Boolean values and produces Boolean values; when thought of as bits, the outputs contain the binary representation of how many inputs were true.



To encode the cardinality constraints we use a network of binary adders with:

- n inputs (the variables $\{p_{ijk}\}_{k=0}^{n-1}$)
- $\lfloor \log_2 n \rfloor + 1$ outputs (counting the number of input variables which are true)

Williamson Matrices

- $n \times n$ matrices A, B, C, D
- entries ± 1
- symmetric, circulant
- $A^2 + B^2 + C^2 + D^2 = 4nI_n$

Symmetric and Circulant Matrices

Such matrices are defined by their first $\lceil \frac{n+1}{2} \rceil$ entries so we may refer to them as if they were sequences.

Examples ($n = 5$ and 6)

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_2 & a_1 \\ a_1 & a_0 & a_1 & a_2 & a_2 \\ a_2 & a_1 & a_0 & a_1 & a_2 \\ a_2 & a_2 & a_1 & a_0 & a_1 \\ a_1 & a_2 & a_2 & a_1 & a_0 \end{bmatrix}$$

symmetric conditions

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_1 & a_2 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_1 & a_2 & a_3 \\ a_3 & a_2 & a_1 & a_0 & a_1 & a_2 \\ a_2 & a_3 & a_2 & a_1 & a_0 & a_1 \\ a_1 & a_2 & a_3 & a_2 & a_1 & a_0 \end{bmatrix}$$

circulant conditions

Symmetric and Circulant Matrices

Such matrices are defined by their first $\left\lceil \frac{n+1}{2} \right\rceil$ entries so we may refer to them as if they were sequences.

Examples ($n = 5$ and 6)

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_2 & a_1 \\ a_1 & a_0 & a_1 & a_2 & a_2 \\ a_2 & a_1 & a_0 & a_1 & a_2 \\ a_2 & a_2 & a_1 & a_0 & a_1 \\ a_1 & a_2 & a_2 & a_1 & a_0 \end{bmatrix}$$

symmetric conditions

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_1 & a_2 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_1 & a_2 & a_3 \\ a_3 & a_2 & a_1 & a_0 & a_1 & a_2 \\ a_2 & a_3 & a_2 & a_1 & a_0 & a_1 \\ a_1 & a_2 & a_3 & a_2 & a_1 & a_0 \end{bmatrix}$$

circulant conditions

Williamson ~~Matrices~~ Sequences

- sequences A, B, C, D of length $\lceil \frac{n+1}{2} \rceil$
- entries ± 1
- $\mathrm{PAF}_A(s) + \mathrm{PAF}_B(s) + \mathrm{PAF}_C(s) + \mathrm{PAF}_D(s) = 0$ for $s = 1, \ldots, \lceil \frac{n-1}{2} \rceil$.

The PAF[3] here is defined

$$\mathrm{PAF}_A(s) \coloneqq \sum_{k=0}^{n-1} a_k a_{(k+s) \bmod n}.$$

[3] Periodic Autocorrelation Function

Compression

The *m-compression* of a sequence $A = [a_0, \ldots, a_{n-1}]$ of length $n = dm$ is a new sequence of length d whose jth entry is the sum of a_{j+kd} for $k = 0, \ldots, m-1$.

Example

The sequence $A = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9]$ has the 5-compression

$$A^{(2)} = \left[a_0 + a_2 + a_4 + a_6 + a_8, \quad a_1 + a_3 + a_5 + a_7 + a_9\right].$$

Compression

The *m-compression* of a sequence $A = [a_0, \ldots, a_{n-1}]$ of length $n = dm$ is a new sequence of length d whose jth entry is the sum of a_{j+kd} for $k = 0, \ldots, m-1$.

Example

The sequence $A = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9]$ has the 2-compression

$$A^{(5)} = \left[a_0 + a_5, \quad a_1 + a_6, \quad a_2 + a_7, \quad a_3 + a_8, \quad a_4 + a_9\right].$$

Useful Properties of Compressed Sequences

Lemma 1

The entries of an m-compression of a ± 1-sequence of length dm:

- have absolute value at most m
- have the same parity as m

Lemma 2

The compression of a symmetric sequence is also symmetric.

Technique 1: Sum-of-squares Decomposition

Đoković–Kotsireas[4] theorem

Williamson sequences satisfy

$$\mathrm{PAF}_A(0) + \mathrm{PAF}_B(0) + \mathrm{PAF}_C(0) + \mathrm{PAF}_D(0) = 4n$$

and this still holds even if A, B, C, D are compressed.

With maximal compression, this becomes

$$\mathrm{rowsum}(A)^2 + \mathrm{rowsum}(B)^2 + \mathrm{rowsum}(C)^2 + \mathrm{rowsum}(D)^2 = 4n.$$

[4] Compression of periodic complementary sequences and applications. *Designs, Codes and Cryptography*

Technique 1: Sum-of-squares Decomposition

Why is this useful?

CAS functions exist which can determine all possible solutions of

$$w^2 + x^2 + y^2 + z^2 = 4n \qquad w, x, y, z \equiv n \pmod 2.$$

This tells us all possibilities for the rowsums of A, B, C, D.

We can encode the constraints

$$\text{rowsum}(A) = w \qquad \text{rowsum}(B) = x$$
$$\text{rowsum}(C) = y \qquad \text{rowsum}(D) = z$$

and pass that information to the SAT solver.

Technique 1: Sum-of-squares Decomposition

Example

When $n = 35$, there are exactly three ways to write $4n$ as a sum of four positive odd squares in ascending order:

$$1^2 + 3^2 + 3^2 + 11^2 = 4 \cdot 35$$
$$1^2 + 3^2 + 7^2 + 9^2 = 4 \cdot 35$$
$$3^2 + 5^2 + 5^2 + 9^2 = 4 \cdot 35$$

Williamson Equivalences

Williamson sequences A, B, C, D can be re-ordered and negated without affecting the Williamson conditions.

Given this, we may enforce the constraint

$$0 \leqslant \text{rowsum}(A) \leqslant \text{rowsum}(B) \leqslant \text{rowsum}(C) \leqslant \text{rowsum}(D).$$

Technique 2: Divide-and-conquer

For efficiency reasons, we want to partition the search space into subspaces. An effective way to do this is to have each subspace contain one possibility for the compressions of A, B, C, D.

- Determine all possible compressions with Lemmas 1 and 2.
- Use the ĐK theorem to remove possibilities which are necessarily invalid (for example, because their *power spectral density* is too large).

Power Spectral Density

The *power spectral density* of a sequence A is

$$\mathrm{PSD}_A(s) \coloneqq |\mathrm{DFT}_A(s)|^2$$

where DFT_A is the *discrete Fourier transform* of A.

Example

The power spectral density of $A = [1, 1, -1, -1, 1]$ is given by:

$$\begin{aligned}
\mathrm{PSD}_A(0) &= 1^2 &&= 1\\
\mathrm{PSD}_A(1) &\approx 3.236^2 &&= 10.472\\
\mathrm{PSD}_A(2) &\approx (-1.236)^2 &&= 1.528\\
\mathrm{PSD}_A(3) &\approx (-1.236)^2 &&= 1.528\\
\mathrm{PSD}_A(4) &\approx 3.236^2 &&= 10.472
\end{aligned}$$

Đoković–Kotsireas Theorem

For all $s \in \mathbb{Z}$, Williamson sequences satisfy

$$\mathrm{PSD}_A(s) + \mathrm{PSD}_B(s) + \mathrm{PSD}_C(s) + \mathrm{PSD}_D(s) = 4n$$

and these still hold if A, B, C, D are compressed.

Corollary

If $\mathrm{PSD}_X(s) > 4n$ then X is not a Williamson sequence or the compression of a Williamson sequence.

Technique 2: Divide-and-conquer

For $n = 35$ with 7-compression, the following is one of 119 compressions which satisfy the ĐK conditions:

$$\begin{aligned}
A^{(5)} &= [\;\;\,5,\;\;\,1,-3,-3,\;\;\,1\,] \\
B^{(5)} &= [-3,\;\;\,3,-3,-3,\;\;\,3\,] \\
C^{(5)} &= [-3,\;\;\,1,-1,-1,\;\;\,1\,] \\
D^{(5)} &= [\;\;\,1,-3,-3,-3,-3\,]
\end{aligned}$$

Technique 2: Divide-and-conquer

If n has more than one nontrivial factor it is possible to perform compression by both factors.

Example

Using 5 and 7-compression on $n = 35$ lead to the following number of instances for each decomposition type:

Instance type	# subspaces
$1^2 + 3^2 + 3^2 + 11^2$	6960
$1^2 + 3^2 + 7^2 + 9^2$	8424
$3^2 + 5^2 + 5^2 + 9^2$	6290

Technique 3: UNSAT Core

The instances generated are very similar and a short reason why one instance is unsatisfiable may apply to other instances.

Example

The $n = 35$ instances contained 3376 variables but only 168 were set differently between instances.

Experimental Results

Timings[5] for Williamson orders $25 \leqslant n \leqslant 35$ are below. The number of SAT calls which successfully returned a result is in parenthesis. A hyphen denotes a timeout after 24h.

Order	Base	Sum-of-squares	Divide-and-conquer	UNSAT Core
25	317s (1)	1702s (4)	408s (179)	408s (179)
26	865s (1)	3818s (3)	61s (3136)	34s (1592)
27	5340s (1)	8593s (3)	1518s (14994)	1439s (689)
28	7674s (1)	2104s (2)	234s (13360)	158s (439)
29	-	21304s (1)	N/A	N/A
30	1684s (1)	36804s (1)	139s (370)	139s (370)
31	-	83010s (1)	N/A	N/A
32	-	-	96011s (13824)	95891s (348)
33	-	-	693s (8724)	683s (7603)
34	-	-	854s (732)	854s (732)
35	-	-	31816s (21674)	31792s (19356)

[5] on 64-bit AMD Opteron processors running at 2.2 GHz

Conclusions and Future Work

- We have demonstrated the power of the SAT+CAS combination by...
  - performing a requested verification of a nonexistence result
  - generating new matrices for MAGMA's Hadamard database.
- We are working on extending the system...
  - to search for other types of combinatorial objects
  - to integrate the CAS calls into the inner loop of the SAT solver.