Verified encodings for SAT solvers

Cayden Codel
Carnegie Mellon University

Abstract

Satisfiability (SAT) solvers are versatile tools that can solve a wide array of problems, and the models and proofs of unsatisfiability emitted by SAT solvers can be checked by simple, verified software. In this way, the SAT toolchain is trustworthy. However, many applications are not expressed natively in SAT and must instead be *encoded* into SAT. These encodings are often subtle, and it is easy to make errors when implementing them. Formal correctness proofs are needed to ensure that implementations are bug-free.

In my talk, I will present recent work on a library for formally verifying SAT encodings, written using the Lean interactive theorem prover. Our library currently contains verified encodings for the parity, at-most-one, and at-most-k constraints. It also contains methods of generating fresh variable names and combining sub-encodings together to form more complex ones, such as one for encoding a valid Sudoku board. Our library is general, extensible, and easy to use, and we hope it will provide the foundation for future encoding verification efforts, such as verifying the encodings of combinatorial problems.

Heesch numbers of unmarked polyforms

Craig S. Kaplan
University of Waterloo

Abstract

A shape's Heesch number is the number of layers of copies of the shape that can be placed around it without gaps or overlaps. Experimentation and exhaustive searching have turned up examples of shapes with finite Heesch numbers up to six, but nothing higher. The computational problem of classifying simple families of shapes by Heesch number can provide more experimental data to fuel our understanding of this topic. I present a technique for computing Heesch numbers of nontiling polyforms using a SAT solver, and the results of exhaustive computation of Heesch numbers up to 19-ominoes, 17-hexes, and 24-iamonds.

Using Walnut: Recent results in combinatorics on words and number theory

Narad Rampersad
University of Winnipeg

Abstract

Walnut is an implementation of a decision procedure for proving theorems concerning a large class of interesting integer sequences from combinatorics and number theory. Some of these sequences include the Thue-Morse sequence, the Rudin-Shapiro sequence, the infinite Fibonacci word, etc. Walnut is capable of verifying a variety of interesting combinatorial properties of such sequences, such as whether or not the sequence is periodic, whether or not the sequence ever contains the same pattern of values repeated twice in a row, etc. Indeed, Walnut can verify any property of such sequences that can be expressed in a certain first-order logic (an extension of Presburger arithmetic). Walnut can even enumerate certain quantities of interest, such as how many distinct blocks of length n appear in the sequence, and many more.

We will present here some recent results that J. Shallit and I have been able to obtain with Walnut about the sequence of Catalan numbers modulo p, for certain primes p. We will also show how to use Walnut to quickly re-prove (and improve) some classical results of Brillhart and Morton on the Rudin-Shapiro sequence.

Rational cubics through non-generic points

Taylor Brysiewicz
University of Western Ontario

Abstract

It is well-known that there are exactly 12 rational irreducible cubics through a generic configuration of 8 points in the plane. In this talk, I will discuss the non-generic cases. Specifically, I will outline the conditions for genericity, and discuss the stratification of the space of configurations of 8 points in the plane via these conditions. I will outline how we combine symbolic and numerical software to compute these 779777 strata as well as the number of cubics through a generic configuration on each stratum. This is joint work with Avi Steiner and Fulvio Gesmundo.

Enumeration and classification of single change covering designs

Brett Stevens
Carleton University

Abstract

Single change covering designs were initially studied in 1969 as a means to optimize magnetic tape access to fill core memory. In a series of ten papers from 1993 to 2001 by Constable, McSorley, Phillips, Preece, van Rees, Wallis, Yucas and Zhang, the spectrum of SCCDs with block size 2 and 3 was completely solved, progress was made for block sizes 4 and higher, and the investigation of "circular" SCCD was begun. In 2018, A. Chafee developed the first recursive construction for circular SCCD which prompted the search and enumeration of small ingredient designs. We describe a canonical augmentation search to enumerate SCCD and generalize Phillips' "end-permutation" and "minor variant" classification schemes. We report the initial findings of the algorithm.