
Computational proof techniques for combinatorics on words

(Org: **James Currie** (University of Winnipeg), **Narad Rampersad** (University of Winnipeg) and/et **Jeffrey Shallit** (University of Waterloo))

CURTIS BRIGHT, University of Windsor

SAT solvers and combinatorics problems

Solvers for the Boolean satisfiability (SAT) problem have been increasingly used to resolve problems in combinatorics. This talk will outline how SAT solvers can be used to effectively search for combinatorial objects as well as produce computer-verifiable proofs of nonexistence. Some simple examples of using a SAT solver will be presented and some noteworthy SAT-based results will be discussed—such as a recent resolution of Lam's problem which produced certificates demonstrating the nonexistence of a projective plane of order ten.

JOEL D. DAY, Loughborough University

Computational methods for solving word equations

Word equations are equations in which each side is a word consisting of terminal symbols (constants) and variables. Solutions are substitutions of the variables for words containing only terminal symbols which result in both sides becoming identical. E.g. $X=bab$ is a solution to $Xab=baX$.

Motivated by applications in software analysis, several tools called string solvers have been developed for automatically deciding the truth of various statements concerning strings, which often involve combinations of word equations with other constraints (regular language membership, comparing lengths of words...). I will discuss some approaches for solving word equations in this setting.

STEPAN HOLUB, Charles University, Prague

Proof assistants in combinatorics on words

Combinatorics of (finite) words is an area where computer assisted formalization may be very helpful. Proofs of even fairly simple results tend to be tedious and repetitive, featuring complicated analysis of cases, which makes them hard (both for referees and readers) to verify. Moreover, despite the short history of the field, important results are sometimes forgotten and rediscovered, or simply repeatedly proven in many papers. Some easily stated problems are vast classification tasks.

We shall discuss how proof assistants can help to tackle these issues and illustrate it by an ongoing formalization project in Isabelle/HOL.

REED OEI, University of Illinois

Design and use of the Pecan system

Pecan is an automated theorem prover for reasoning about properties of automatic sequences, most notably including Sturmian words. It is capable of efficiently proving non-trivial mathematical theorems about all Sturmian words using a decision procedure based on Büchi automata. We give an overview of the Pecan system and show how to use it to prove some interesting theorems about Sturmian words.

JEFFREY SHALLIT, University of Waterloo

Proving theorems in combinatorics on words with Walnut

Walnut is free software, written by Hamoon Mousavi and recently modified by Aseem Baranwal, for proving (or disproving) statements about combinatorics on words. Specifically, it can prove or disprove any first-order claim about automatic sequences

s that is phrased in terms operations like addition, subtraction, comparison, boolean connectives, and indexing into s . In this talk, I will briefly give the theory behind Walnut and give many examples of the kinds of things one can prove with it. Both existing results and new results will be covered.