Outline of Proposed Research

My short and sweet research goal is to *develop a search tool that mathematicians and engineers can use to solve larger combinatorial problems*. The method I propose to do this is by combining satisfiability (SAT) solvers with computer algebra systems (CAS).

This "SAT+CAS" method is still in its infancy, having only been first proposed in 2015. Despite this, it has already had some great successes, including solving special cases of the Ruskey–Savage and Norine conjectures (JAR 2017), automatically generating a counterexample to the Williamson conjecture (CASC 2016), finding a huge number of new Williamson matrices (AAAI 2018), and verifying the complex Golay conjecture (ISSAC 2018); see my contributions attachment for more details. The varied conjectures and problems in which the SAT+CAS method has already found application speaks to its versatility and its great potential to push the state-of-the-art in many more applications for years to come.

Briefly, the reason the SAT+CAS approach is so effective is because it uses the respective strengths of both tools. SAT solvers are powerful "automated reasoning" search tools that can solve many important and difficult problems such as verifying the correctness of a microprocessor's design. Despite this power, SAT solvers do not perform well on problems that have a lot of symmetry. On the other hand, a CAS is an effective tool for detecting and working with symmetries but they are not optimized for searching.

Many combinatorial problems that require *both* powerful search and symmetry detection have been out of reach of present algorithms—but the SAT+CAS method has the potential to make such problems feasible. For example, I'm excited to apply the SAT+CAS method to the problem of optimizing Boolean circuits, i.e., finding a circuit with the fewest number of logic gates that computes a given Boolean function. This problem is considered so difficult that it often isn't even attempted but the consequences of having a system that could effectively optimize Boolean circuits would be massive. For example, engineers who are designing electronic circuits could use the system to produce more efficient designs and this would be hugely impactful considering the enormous number of circuits that are produced and used.

Recently SAT solvers have made some progress on this problem and I believe incorporating CAS functionality will push the technology even farther and be useful to a lot of people. More generally, I believe SAT+CAS methods will become more and more common in the future and examples of SAT+CAS systems being applied to problems will be useful to those who want to use the method for themselves.

Justification for Eligibility of Proposed Research

My proposal falls under the mandate of NSERC as the research I am proposing is in the field of computer science and has applications to mathematics and engineering. I will be developing and implementing improved combinatorial search methods using satisfiability (SAT) solvers and computer algebra systems. These methods will allow large combinatorial spaces to be searched faster than is currently possible, allowing mathematicians to verify or find counterexamples to conjectures in combinatorics. It will also have application to engineering as many engineering problems reduce to a combinatorial search for which SAT solvers are currently used (e.g., in microprocessor verification).

Curtis Bright

Thesis Information

My PhD thesis studied a number of difficult problems from combinatorics that have large search spaces. In my thesis I developed new methods for solving those problems, produced optimized implementations of those methods, and used the implementations to verify and extend previous work. In particular, I was able to solve larger problem instances than had previously been solved.

As a concrete example, I studied Williamson matrices from combinatorial design theory. These matrices are objects that were originally defined in 1944 and have since been widely studied for their combinatorial applications. In particular, I provided the first enumeration of Williamson matrices in the even orders up to 44 (only previously done up to 18) and provided an independent verification of the 1993 result that Williamson matrices do not exist in order 35.

These results were achieved using the system MathCheck, the first piece of software that combined satisfiability (SAT) solvers with computer algebra systems (CAS) to effectively solve large combinatorial problems. I was the lead developer of this system and in my thesis I showed that the SAT+CAS paradigm was highly effective at searching for Williamson matrices and could perform the search much more efficiently than applying either SAT solvers or CAS methods in isolation. I analyzed why this was the case and gave guidelines for the kinds of problems that are suitable for solving with the SAT+CAS paradigm. Furthermore, I described how to effectively apply the paradigm to other problems.

Curtis Bright

Justification for Location of Tenure

I'm fortunate to have multiple researchers interested in collaborating with me and extending the research that I've done so far on the MATHCHECK system and combining satisfiability (SAT) solvers with computer algebra systems (CAS).

The most enthusiastic researcher that I've personally met who has expressed interest in extending our work is professor Kevin Cheung of Carleton University. In fact, at the *Ottawa Mathematics Conference* last year he gave an invited talk where he spent several slides outlining our work on the Williamson conjecture despite having no connection to our research group. He later reached out to me and I gave a talk at the Ottawa–Carleton Combinatorics and Optimization seminar series. There I met his research group and several members also stated their interest in MATHCHECK and applying SAT+CAS methods to their own research.

When I met with professor Cheung we discussed four concrete problems that are of interest to both of us and in which we think progress can be made by employing SAT+CAS methods:

1. Enumerating projective planes of small order. Very important in projective geometry, their non-existence in order 10 has never been independently verified and their existence in order 12 is open.

2. Improving 3×3 matrix multiplication. A hugely important problem in computer science whose best known algorithm hasn't been improved since 1975, though a second algorithm with the same complexity was found (using a SAT solver) in 2011.

3. Optimizing Boolean circuits. Solutions to this problem would have the huge potential to make computers more efficient by improving the design of fundamental digital electronic circuits.

4. Finding disjunct matrices. Such matrices are useful for group testing, a procedure that has applications to many practical fields including computer science, engineering, and statistics.

Additionally, earlier this year I met several times with Marijn Heule, a computer science professor at the University of Texas at Austin and one of the leading experts in using SAT solvers to solve extremely large combinatorial problems. We discussed several problems and he expressed interest in using SAT+CAS methods in his work on the Collatz conjecture and the Hadwiger–Nelson problem of computing unit-distance graphs with the largest possible chromatic number.

I also had a phone conversation with Joël Ouaknine, a computer science professor at Oxford University. He was very enthusiastic about using SAT+CAS methods to search for combinatorial objects used in the proof of Conway and Kochen's "Free Will Theorem" called Kochen–Specker systems. He strongly believed that an optimized SAT+CAS system would be able to make progress on finding the smallest possible Kochen–Specker system.

Contributions / Statements

I. Contributions to research and development

a. Articles published or accepted in peer-reviewed journals

E. Zulkoski, **C. Bright**, A. Heinle, I. Kotsireas, K. Czarnecki, V. Ganesh. (2017) Combining SAT Solvers with Computer Algebra Systems to Verify Combinatorial Conjectures. Submitted to the Journal of Automated Reasoning in July 2016, accepted November 2016. 58: 313–339. (PhD work)

C. Bright, R. Devillers, J. Shallit. (2016) Minimal Elements for the Prime Numbers. Submitted to the Journal of Experimental Mathematics in January 2015, accepted June 2015. 25: 321–331. (PhD work)

b. Articles submitted to peer-reviewed journals

C. Bright, I. Kotsireas, V. Ganesh. (2018) Applying Computer Algebra Systems and SAT Solvers to the Williamson Conjecture. Submitted to the Journal of Symbolic Computation in February 2018, submission number 1, 26 pages. (Postdoc work)

c. Other peer-reviewed contributions

C. Bright, D. Đoković, I. Kotsireas, V. Ganesh. (2018) A SAT+CAS Approach to Finding Good Matrices: New Examples and Counterexamples. Submitted to the AAAI Conference on Artificial Intelligence in September 2018, submission number 2699, 8 pages. (Postdoc work, international conference)

C. Bright*, I. Kotsireas, A. Heinle, V. Ganesh. (2018) Enumeration of Complex Golay Pairs via Programmatic SAT. Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. 111–118. (Postdoc work, oral presentation, international conference)

C. Bright*, I. Kotsireas, V. Ganesh. (2018) The SAT+CAS Paradigm and the Williamson Conjecture. Submitted to ACM Communications in Computer Algebra in May 2018, accepted May 2018, 4 pages. (Postdoc work, poster presentation, international conference)

C. Bright*, I. Kotsireas, V. Ganesh. (2018) A SAT+CAS Method for Enumerating Williamson Matrices of Even Order. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. 6573–6580. (Postdoc work, oral and poster presentation, international conference)

C. Bright*, V. Ganesh, A. Heinle, I. Kotsireas, S. Nejati, K. Czarnecki. (2016) MATHCHECK2: A SAT+CAS Verifier for Combinatorial Conjectures. Proceedings of Computer Algebra in Scientific Computing. 117–133. (PhD work, oral presentation, international conference)

C. Bright*, A. Storjohann. (2011) Vector Rational Number Reconstruction. Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation. 51–58. (Master's work, oral presentation, international conference)

II. Most significant contributions to research and development

The work I am most proud of is *Applying Computer Algebra Systems and SAT Solvers to the Williamson Conjecture* that was submitted to the *Journal of Symbolic Computation* this year and was presented as a poster at this year's ISSAC conference.

In this work we used our satisfiability solver + computer algebra (SAT+CAS) hybrid system MATHCHECK to show that the Williamson conjecture is true for all even orders up to 70 and explicitly constructed over 100,000 new sets of Williamson matrices that are inequivalent—a result that surpassed even our expectations, considering that Williamson matrices have been extensively studied since 1944. For example, some of the first researchers to study Williamson matrices were from NASA's Jet Propulsion Laboratory. In 1961 they found a set of Williamson matrices of order 23 while developing codes for communicating with spacecraft.

Despite attracting sustained interest for about 75 years, previous computational searches were not able to produce anywhere near 100,000 Williamson matrices, nor were they able to scale up to order 70. Prior to MATHCHECK, complete searches had only been done in the even orders up to 18 (in 2006) and the odd orders up to 59 (in 2008) and had found fewer than 150 inequivalent sets of Williamson matrices in those orders. We were able to greatly push the state-of-the-art by using the SAT+CAS paradigm and proving some new properties of Williamson matrices.

This research was a collaboration between myself, my supervisor Vijay Ganesh (a researcher in satisfiability checking), and professor Ilias Kotsireas (a researcher in computer algebra). I wrote the scripts used to perform the search, ran the scripts on a high-performance computing cluster, and proved the new properties of Williamson matrices on my own. I also wrote the content of the paper and poster, incorporating feedback on drafts from Vijay, Ilias, and professor Dragomir Đoković. Vijay supplied the direction for the project (where to focus our efforts) as well as expertise on SAT solvers. Ilias supplied expertise on Williamson matrices and the very large literature on complementary sequences that he is intimately familiar with.

As a concrete example of this collaboration, I would write scripts for searching for Williamson matrices and then meet with Vijay and Ilias to discuss ideas for how their performance could be improved. I would implement those ideas, and the process would iterate. The roots of the work start in 2015—at that time MATHCHECK could only run searches in orders around 10. By the start of 2016 we could run searches up to around order 35 (which verified a counterexample and resulted in a paper at CASC 2016) and by the time I finished my PhD in early 2017 we could run searches up to around order 45. We chose to submit to the *Journal of Symbolic Computation* because it is the leading journal in computer algebra and they are publishing a special issue on combinations of satisfiability checking and symbolic computation which perfectly captures the topic of our work.

The second paper I would like to highlight is *Minimal Elements for the Prime Numbers* that was published in *Experimental Mathematics* in 2016. The roots of this paper stem from a course I took with Jeffrey Shallit as a PhD student. During the course Jeff distributed a list of about a dozen open problems and one of these problems was if a number of the form $80\cdots0111$ in base 13 could be prime. Based on some heuristic reasoning I expected infinitely many primes of the form in question and I wrote a primality checking script that found such a prime.

This solved the problem (open since 2000) of determining the set of "minimal primes" in base 13—the smallest set of primes which appear as a subword of *every* prime expressed in base 13. Computing the set of minimal primes even in small bases is a very hard problem; even though the set of minimal primes is provably finite the problem may actually be undecidable, i.e., impossible to resolve. Despite this, Raymond Devillers, a professor at the Université Libre in Brussels, Belgium had solved a number of small orders. Jeff connected us and we exchanged a number of emails about his techniques. I simplified and extended his method, proved some new theorems that were helpful, and wrote scripts that solved the problem in a number of additional bases and was able to push the search much further than Raymond had taken it; for example, my scripts found a probable prime with over a million digits—at the time the tenth largest known probable prime number ever discovered.

The writing was a collaborative effort between Jeff and myself; Jeff wrote the introduction and described why the problem was interesting and I wrote the sections describing the theorems, search method, and results. We decided to publish in *Experimental Mathematics* because we believed (or at least hoped) that the result was of sufficient interest for the readership of such a well-respected journal. It was a bit risky, as some mathematicians look down on these "prime number games" as not serious work.

The work was not intended to have any application beyond generating interest in mathematics. It's at least plausible that it may never have any application beyond that but it's not a bet I'd personally be willing to make—for example, computing the "minimal strings" of strings of DNA has applications to DNA strand design.

The last paper I would like to highlight is *Enumeration of Complex Golay Pairs via Programmatic SAT* that was published at ISSAC this year. In this paper we verified the 2002 conjecture that complex Golay pairs of order 23 do not exist. Unfortunately, after we had done this we discovered a little-known paper from 2013 that claimed the same result. However, the paper provided no code or implementation details and only an outline of their method. We felt an independent verification using code that was made freely available was worthwhile, therefore we submitted the paper to ISSAC, the leading conference in symbolic computation and computer algebra.

The paper is also interesting in that it demonstrates the versatility of the SAT+CAS method as it uses the SAT solver and CAS differently than in the Williamson case study. It also demonstrates the power of the SAT+CAS approach—beforehand I implemented a CAS-only approach and took months to solve a problem a problem that the SAT+CAS method solved in a day.

III. Applicant's statement

Research experience. In addition to the work I've done on MATHCHECK I've been fortunate to have had two internships with Maplesoft, the developers of the computer algebra system MAPLE. During these internships I worked on three projects: I improved the efficiency of MAPLE's satisfiability checking routines, I used SAT solvers to improve the efficiency of some other combinatorial routines, and I created worksheets to showcase MAPLE's SAT solving functionality.

In the first project, I replaced MAPLE's default SAT solver MINISAT (which hasn't been updated since 2010) with the modern state-of-the-art SAT solver MAPLESAT. I also modified the SAT solver internally to accept native MAPLE expressions. This avoids the overhead of converting those expressions into the normal format that SAT solvers use, resulting in the initialization process being about 1000 times faster.

In the second project, I improved the efficiency of some of MAPLE's combinatorial routines (computing the chromatic number and maximum clique of a graph) by translating the given problem into a SAT instance and calling MAPLESAT. The SAT approach was able to solve more instances in less time, including solving one benchmark in seconds that the previous version of MAPLE couldn't solve in an hour.

In the third project, I wrote articles for Maplesoft's application center demonstrating how to solve problems of practical interest by translating the problem into a SAT instance. In particular, I wrote articles on the *n*-queens problem, Sudoku puzzles, Einstein's logic riddle, maximum clique finding, and the 15-puzzle.

This experience has been very beneficial because I improved my understanding of how SAT solvers and CASes work and how to apply them to solve real problems. This knowledge should be invaluable in my future research, especially considering my goal is to improve the connection between computer algebra systems and SAT solvers to solve real problems.

Relevant activities. I have been fortunate to have a number of leadership opportunities. One of the first was in the summer of 2014 when I was a mentor for Google's *Summer of Code* program. I supervised an undergraduate student who spent the summer implementing a highly optimized lattice basis reduction algorithm. I answered questions and provided guidance about lattice theory, implementation issues, and various troubleshooting. The project was successful and the code that was produced is available in the latest version of the C library FLINT.

I have also been fortunate to have multiple opportunities to teach computer science. As a PhD student I often led tutorials in undergraduate logic and did significant extra work behind the scenes; for example, I would write and distribute my own assignment solutions if they were not provided by the instructors. I received an outstanding TA award in the summer of 2013 and following that I was an instructor of three first-year courses in computer science. In the fall of 2014 I coordinated with 5 other instructors to teach 876 students, in the summer of 2015 I was the sole instructor of 145 students, and in the fall of 2015 I taught 258 students with one other instructor.

Socially, I was a member of the Computer Science Graduate Student Association and administered their website in 2014. I also started salsa dancing in 2013 despite no background in dance or much of any physical activity. Predictably, I was terrible—but I persisted for years and slowly watched my competence improve to the point that I now often teach salsa as a volunteer. I was also on the executive team of the club KW Salseros as an instructor from Sep. 2015 to Aug. 2016, as a treasurer in the fall of 2015, and as a secretary in the winter of 2016.

Special circumstances. The primary circumstance that hindered my research productivity in the past was serious social anxiety. The problem was severe enough that I would avoid human contact as much as I could and only meet people when absolutely necessary. Needless to say, this was not healthy, not sustainable, and not conducive to excelling as a researcher. Realizing that something needed to change, at the beginning of 2013 I vowed to start being social and started salsa dancing and eating lunch in the computer science lounge every day. It took years of working on my social skills before the anxiety stopped but the result has been nothing short of transformative. I am comfortable at navigating social situations to the point that I enjoy it and without this change none of the research after my first ISSAC paper would have been possible.

The anxiety also caused a lack of direction: for example, after completing my master's degree I wanted to continue with research but didn't talk to professors about that. I ended up taking courses as a non-degree student before telling my first supervisor of my research plans. Unfortunately my anti-social tendencies (coupled with differing interests) made collaboration difficult and he asked me to find another supervisor. I taught courses in the meantime to keep busy and started working with Vijay Ganesh at the end of 2015. He officially became my new supervisor (along with Krzysztof Czarnecki) in 2016—and the rest is history.