Kenneth W. Regan **Research Statement** **July 7, 2021**

My research in computational complexity has been diverse but is animated by three strands with ambition of fundamental science more than puzzle-solving:

1. Emphasis on explaining why it is hard to prove that problems are hard—and what it will take to do so;
2. Combining ideas from different disciplines—and often coding them;
3. Trying to build on cleverer or more elegant solutions to sub-problems.

My interest in the first began with my dissertation work and early papers on complexity-related undecidability results. Among work in this area that predated my thesis were papers (co-)written by Juris Hartmanis and John Hopcroft on independence results and ones by Rich de Millo and Richard Lipton on the inability of weak systems of number theory to prove $\mathsf{P} \neq \mathsf{NP}$ [26, 27, 9, 10]. These results are now understood as orbiting around the combinatorial properties that really drive the present barriers to lower bounds, as captured by the "Natural Proofs" and "Algebrization" frameworks [41, 2] amid the general hardness-versus-randomness paradigm.

It is easy to code up particular Turing machines M that accept particular NP-complete languages, such that your favorite strong formal system F cannot disprove the formal statement of the language of M belonging to polynomial time. This is done by bolting on to M a routine that spends a modicum of time searching for an inconsistency in F and rejects the input x if so, such that as $n = |x|$ increases so does the allotted search time. Then proving that $L(M)$ is infinite, let alone being not in P and equal to a known hard language in NP such as SAT, entails proving the consistency of F, which by Gödel's theorems, F cannot do. My dissertation [43] probed the more-substantial case of languages $A \notin P$ for which that fact cannot be proved regardless of the Turing machine M_A by which A is represented. I gave a most-general form of Uwe Schöning's "uniform diagonalization theorem" [60], a tool of structural complexity that was popular in the early 1980s, that yields a "factory" for such results, and I related this to a natural topology on the space of languages that is Hausdorff but not metrizable [44, 46]. I also used topological ideas to streamline a result of Alan Selman (with his student Jochen Grollman, [20]) that I had independently obtained [45]. One non-logical result that might be thought surprising is that every recursive enumeration of NP by languages L_k has infinitely many k such that L_k equals an encoding of the language of undirected graphs that have a clique of size k [48]. Alas, my thesis did more to "explain away" this second vein of independence results than to build powerful new techniques upon them. So after it, I re-tooled myself to learn and apply more combinatorial approaches.

I was involved in the earliest after-discussions of both Seinosuke Toda's celebrated theorem and the "Natural Proofs" phenomenon. I and independently Thomas Schwentick realized that Toda's theorem requires access only to one bit—wlog. the middle bit—of one $\#\mathsf{P}$ function value. We formulated a complexity class MP around the idea as one idea of a tightest upper bound for the polynomial hierarchy [57]. I devoted a lot of time to whether MP equals polynomial space, and I still do not know of an oracle that separates them. Schwentick and I joined forces with

Fred Green, Johannes Köbler, and Jacobo Torán. who made a more-combinatorial application of these ideas to Boolean circuits, in the journal paper [19].

I noted an affinity between the "Natural Proofs" machinery and the resource-bounded measure framework of Jack Lutz in the weekly research meetings led by Alan, and D. Sivakumar joined me to develop it (and later Jin-Yi Cai) [58]. For awhile this seemed to propagate into a real way to separate BPP from exponential time via a modification of Lutz's martingales. Harry Buhrman, Dieter van Melkebeek, and Martin Strauss joined in after this was discussed after a 1997 Schloss Dagstuhl meeting and this became a 5-author paper [6] in *SIAM J. Comput.*; I can relate that at one point the latter two were really high on pushing it further but like much else in complexity theory it ground down.

A third area of my work generalized examples like the above languages of graphs with a clique of fixed size k into what I called "finitary substructure languages" [47]. This flowed into the river of Fixed-Parameter Complexity Theory as developed mostly by Rod Downey and Mike Fellows. My main contribution across two papers with them [12, 13] was a characterization of the *W-hierarchy* by descriptive logic that has been referred to as the fixed-parameter analogue of a famous characterization of NP by Ronald Fagin (in the textbook [18]). The W-hierarchy is analogous to the polynomial hierarchy. I used what I felt was a kludgey limitation on the logical quantifiers that come after a leading $\exists$ and was annoyed that I couldn't make the proof work without it. Two decades later, this remains an unresolved problem; relaxing the kludge in various ways defines alternative analogues of the W-hierarchy (also covered at length in [18]).

My main effort in the 1990s, however, was to develop a richer theory of linear and quasi-linear time. With Jie Wang and Sivakumar and Alan's student Ashish Naik, I made some pieces of the Berman-Hartmanis isomorphism conjecture and Toda's theorem work under the latter time bound, including an application of optimal-rate error-correcting codes [59, 37]. The barriers to lower bounds really begin right at linear time. I made a more-robust version of linear time [50] to which techniques in Kolmogorov complexity could apply for lower bounds—very modest, but at least super-linear. I tightened the characterization of the NC^k classes by machines [49] and found lower bounds on Boolean circuits whose graphs have limited expansion [51]. The latter paper attracted Lipton's attention for a talk invite to Princeton, as it extended techniques he used with Robert Tarjan on graph separator theorems.

However, this was all diverted by the prospect of a novel algebraic attack on the core polynomial-time questions. The simple takeaway from the Razborov-Rudich phenomenon is that a hardness predicate $R(f)$ capable of separating Boolean functions $f(x_1, \ldots, x_n)$ from polynomial-sized circuits either must hold for negligibly few functions f (thus contravening the reality that a random Boolean function is hard) or must have decision complexity more than singly exponential in n (which is quasi-polynomial in the size $N = 2^n$ of the truth table of f). The subtext of [42] is that super-exponential hardness is "unnatural" because it is humanly hard to control mathematical methods at such high complexity levels. Yet in *algebraic geometry*, in particular *polynomial ideal theory*, the simplest and most natural predicates are exponential space complete (per seminal results of Ernst Mayr and Albert Meyer [35]), yet have been understood and applied by mathematicians going back to David Hilbert and his one-time co-advised student: the world chess champion Emanuel Lasker. I was hooked: this was a vein for "super-natural proofs"—plus concrete fun using Gröbner basis algorithms to solve prob-

lems. Indeed, I thought deep theorems in this area might already have done the heavy lifting for a proof of $\mathsf{NP} \neq \mathsf{P}$—in a way that might be realized by extending the "algebraic-geometric dictionary" to embrace complexity classes. I formulated a concrete hardness predicate $R(f)$ involving the number of minimal monomials in the Jacobian or Hessian ideal of some algebraic analogue f' of f. When f' is the determinant function the number is *zero*, but when f' is the permanent there are many minimal monomials. Separating the complexity of the permanent and determinant is considered the algebraic analogue of $\mathsf{P} \neq \mathsf{NP}$. A computation on the Hessian of the 5×5 permanent polynomial lasting 38-1/4 days yielded over 128,000 minimal monomials in a reduced Gröbner basis of size 257,576. (I had estimated that doing so for the Jacobian would take over 100 years.) This pointed toward known cases of doubly exponential count, which is maximum by the concrete bound on Hilbert's finiteness proof.

Alas, Teo Mora of the University of Genoa, Italy found such a case where f has uniform linear-sized circuits, so my $R(f)$ is not "effective" against P. I subsequently realized that the mechanism by which this approach overcomes the exponential-time limitation of Razborov-Rudich is self-defeating with regard to effectiveness against P. It was hard to salvage much from that, though a review [52] I wrote of the similar motivation and related mathematics of Ketan Mulmuley's Geometric Complexity Theory programme (with Milind Sohoni and others) was much appreciated.[1] This ran into an illenss in 2003–2005 that was evidently caused by a known principal side effect of the drug Reglan and the events of 2006 in chess described below. Where my work emerged was in polynomials associated to quantum circuits, in a different way from the polynomial method for quantum query lower bounds launched by Buhrman among others. The paper [8], including Michael Nielsen of the famous Nielsen and Chuang text on quantum computing [38], translated circuits using the universal but limited basis of Hadamard, CNOT, and Toffoli gates into polynomials over the binary field $\mathbb{Z}_2$. They discussed an extension to the quantum T-gate using a mashup of mod-2 and mod-8 arithmetic. I worked out not only how to do this cleanly into $\mathbb{Z}_8$ but how to extend the algebra naturally for virtually the entire universe of commonly-used gates into a wide choice of target algebras, both multiplicatively and additively. In 2007, we were visited by Amlan Chakrabarti while he was still a PhD student in practical quantum circuit engineering, and we hatched a plan to use my translation (fed to heuristic polynomial equation solvers such as those using Gröbner-based algorithms) as one engine of his circuit simulator. I completed a long draft paper while on sabbatical in Montreal in early 2009 and tried to interest their quantum group, but the work did not reach critical mass until it was augmented by a parallel translation into Boolean logic by my PhD graduate Chaowen Guan. This became the paper [55]; some of this material is in my textbook with Lipton [33, 34]. We programmed the logic part so that SAT-counters including *sharpSAT* by Marc Thurley and *Cachet* by Henry Kautz can be interfaced for heuristic simulation of the quantum circuits. They may be successful in many application domains but not yet this one: they need to be kicked even just to deduce that two adjacent Hadamard gates cancel.

Guan worked also on a lager motive for this work. Its springboard is the mechanism of the $\Omega(n \log n)$ lower bound on algebraic circuits for certain (numerous) polynomial functions $f(X)$ by Volker Strassen and Walter Baur [5], which is sometimes called the only *general* nonlinear lower bound known in complexity theory. That is based on an algebraic invariant called the

[1] I had someone say he couldn't begin Mulmuley's work without reading my review, and Scott Aaronson paid it the silent compliment of citing it inline with the original as "Mulmuley and Sohoni [39,57]" in [1].

geometric degree of the Jacobian ideal of f or rather its mapped version $\langle y_i - \frac{\partial f}{\partial x_i} : i = 1, \ldots, n \rangle$, which is irreducible. The polynomial translation of an n-qubit quantum circuit involves the *partition function*

$$Z(f) = \sum_x \omega^{f(x)}$$

(where ω is a principal 2^n-th root of unity), which describes the behavior of the circuit as a physical system. This is embodied in work by Gibbs in statistical physics even before the 1900 birth of quantum theory. I believe that algebraic invariants associated to f ought to have physical meaning. In particular, it may yield nonlinearities that may explain why the simple linear gate-counting measures of quantum circuits have not matched the effort and obstacles that have been encountered while trying to engineer quantum computations. Our effort to apply Strassen's method directly has been hindered by a step that looks simple—akin to completing a commutative diagram—but has not yielded. It is also possible that a nonlinearity with direct physical meaning may exist but be no large than the $O(\log n)$ factor already involved in the quantum fault tolerance theorem (as also in Baur-Strassen). Similar unpublished work by Bacon, van Dam, and Russell [4] also merits further attention.

Instead, Guan and I improved the running time for simulating the main classically polynomial-time subcase of quantum circuits, called stabilizer circuits, from $O(n^3)$ to $O(n^\omega)$, where now ω means the exponent of matrix multiplication. This likewise improves the time for counting solutions to quadratic equations over $\mathbb{Z}_2$ from [14], a fact that was new to its second author in recent communication. The improvement is not really practical because even Strassen's improvement from the simple $O(n^3)$ matrix multiplication algorithm requires large matrix size to realize. The paper [21] fell short of FOCS, partly on ground that stabilizer circuits were more interesting circa 2005 and our improvement is less surprising now given recent developments in linear algebra—some of which we applied, of course. I have been diverted by the "chess cheating pandemic" from repackaging the paper. We also have the published paper [22] on relations to Tutte polynomials and matroid theory that were also developed in posts on the GLL blog.

My most recent work is with Ronald Fagin and Jonathan Lenchner of IBM and Nikhil Vyas of MIT [15] on an attempt to revive the logic approach to lower bounds. It defines and applies a variant of the classic Ehrenfeucht-Fraïssé games that makes the second player (called Duplicator) more powerful. After having already conversed with Neil Immerman about this, we found that Neil had defined the games in a passage added to a 1979 conference paper in a 1981 journal version [30] we had missed. Neil's motivation then speaks ours: "We urge others to study [the variant game], hoping that [it] may become a viable tool for ascertaining some of the lower bounds which are 'well believed' but have so far escaped proof." We came up with new quantitative results on linear orders, and now on distance in general graphs in conversations with Ryan Williams. Will this fare better than 40 years ago? One hope I've thought of is that by saturating Duplicator, we may set up the same kind of "flip" duality as occurs in Mulmuley's GCT programme, whereby the absence of a proof by one means would imply the presence of a shorter witness (an "obstruction" in GCT) for the opposite. I will mention a further ambitious and synthetic idea after describing my chess research.

Chess Research

Most important to say first about my model of human decision making at chess is that it is not *about* detecting cheating with strong computer chess programs. It is a full *predictive analytic* model. This means that it generates probabilities p_i for specified events m_i, which in my case are the legal moves in a chess position. It also computes variances for the resulting distributions and hence generates confidence intervals for various predicates. Previous work [23, 25, 24, 16, 17] stopped at evaluating such predicates, not judging their likelihoods—such as the likelihood of finding and playing yea-many moves that match the choices of some chess program. The only other published work I know that is close to predictive analytics is the Markov-chain model of Alliott [3]. My model is frequentist. Guy Haworth of the University of Reading (UK) formulated a Bayesian version and we shared the initial papers [11, 29, 56], but I observed that maximum-likelihood estimation (which the Bayesian update rule approaches) gives inferior results [54]. That MLE disagrees with my simple training of the main predicates as unbiased estimators is disquieting, but has persisted through all iterations of the model.

The core equation uses a utility function along lines of the classic multinomial logit model ([36] and many more) but is loglog-linear rather than log-linear. That is, the utility, which is non-positive relative to the value of the best move m_1, is equated not to $\log p_i$ or $\log p_i - \log p_1$ but rather to $\log\log(1/p_1) - \log\log(1/p_i)$. The values of moves as judged by strong programs are the only inputs to the equation; I do not even use the time the player consumed on each game turn or the time budget left because the timing data is not available reliably in bulk for training. The surrounding mathematics has elements common to psychometrics/item-response theory, in particular the theory of standardized tests—which I regard as the main conduit for applications outside chess.[2] The model has been trained on vast amounts of data: over ten million positions from several hundred thousand games by players of all strengths, each analyzed with five chess programs. *All* archived games since 2015 have been analyzed in a quicker mode that generates test sets for the predicates. This totals 2TB (uncompressed) of text data gathered using free time provided by the UB Center for Computational Research (CCR).

For cheating detection, it provides z-scores under the normal approximation to multinomial distribution. The conformance of the z-scores to the bell curve (in the region above $z = 2.00$, in particular) over large populations has been validated both in field tests of thousand-player tournaments and on a million scale by randomized resampling applied at all skill levels from beginner to champion. The training and resampling runs on CSE departmental machines accessing the data on CCR via an SSH transfer tunnel. The software corpus is almost 35,000 lines of C++ for the model and about 15,000 lines of Perl scripts (parts of both written by my 2016 PhD graduate Tamal Biswas, plus some scripts duplicated in Python) for data gathering and collation. Mine has been the only system recognized and employed by FIDE for in-person chess since 2014. Online playing platforms such as Chess.com, Lichess, the Internet Chess Club, Playchess, and Tornelo have their own cheating detection systems, which avail extra information about player behavior, including window focus and timing rhythm data, that has been correlated with known cheating instances. These systems employ neural nets and other

[2]The humorous blog post http://angrystatistician.blogspot.com/2013/03/baseball-chess-psychology-and.html makes a related point about the chess Elo rating system, which is bound up in my work.

classifiers. The scale online is vast: over 100 million games per month *in toto*, compared with the 10 million games in the main ChessBase library, representing the entire recorded history of (mostly in-person) chess. Dreadful to relate, the frequency of cheating online is vastly higher than for in-person chess; consensus estimates that accord with my observations place the latter between 1-in-5,000 and 1-in-10,000, whereas during the pandemic I've observed 2% as a most-case lower bound and upwards of 10% in youth tournaments. This has often created more cases in a day than I encounter in a year of in-person chess.

Several points of contrast in my system were brought out a recent webinar hosted by the Washington, DC-based Center for AI and Digital Policy on June 30:

1. The other classifiers are trained on data from both honest and cheating players. My model needs only honest data that is freely available, whereas cheating data is secret and either scant or proprietary.[3]

2. My model is not compromised by divulging details of a judgment and the information on which it was based. The z-scores are explainable in ways that neural-net determinations are not. During the pandemic, my results have often been communicated in cases where the official judgment was made by the online platform—this has happened with all platforms except Playchess (which is run by Chessbase GMBH).

3. Both my model's z-score and its Intrinsic Performance Rating (IPR) are single numbers—in the matter of a credit score or loan rating—but they provide dimensions along which an accused player can argue at stages after my initial report. My model has two main parameters, which correspond to the classic dichotomy of strategic and tactical ability, plus a third parameter that indirectly reflects a player's habitual depth of thinking. This creates a 2:1 or 3:1 correspondence to the player's rating; that is, chess ratings R are isobars on the model's nonlinear landscape. The mean parameter values create a central cut through the isobars that is close to linear in R, and this is what I use for the initial test. A player can try to defend by claiming to be far from the main sequence. This can then be tested by "profiling" earlier games by the player as described in [28]. Speaking more simply, my system seeks to empower stakeholders with information rather than supersede judgment with one authoritative number.

4. My model has numerous cross-checks on its judgments. The IPR feature gives the playing quality in a way that supplements the z-scores. My model can compute its own mean prediction error. My graduate seminar in Fall 2019 developed a second z-test based on David Spiegelhalter's prediction statistic; the separate calculation of mean prediction error (a multiplicative 0.04) enabled calibrating it as an unbiased estimator. This z-test is not stronger as originally hoped: on known/sanctioned cheating cases it has averaged the same score as the regular test. But it is usually within 0.20 of the main z-score and serves to second it.

[3]The presence of a small amount of uncaught cheating in my data is more than offset by errors in the recorded moves of games causing phantom blunders, not all of which my scripts catch and clean. The honest chess data is exempt from human-subjects restrictions for several reasons: game records are public, are taken under natural conditions where players expect to be observed, and are not compromising apart from the embarrassment of seeing one's blunders and losses in print.

Not only do these four properties constitute plus-points from the perspective of the new field of *algorithmic fairness*, the cross-checks have been called upon to correct an actual massive amount of bias resulting from the pandemic. FIDE recognizes only in-person chess as valid for its mainstay ratings, but there has been only a trace of in-person play for 15 months. Most players' official ratings have effectively been frozen since the April 1, 2020, rating list. Online platforms have their own rating systems but they are not indexed to FIDE's and generally have inflated and unstable rating values. However, the players' brains have not stayed frozen, especially those of young, developing players who are keen enough to contend in national and regional championships. I was first able to discern significant lag in young players' ratings in September 2020, and my initial estimate was solidified by my monitoring the entire World Youth Rapid Championship in November and early December: 15 Elo $\times$ months of the pandemic for teenage players, more for preteens (originally 50% more, i.e., 22.5 Elo per month, since raised to 25 per month after large overseas scholastic tournaments in March and April and the US Scholastics in May). This estimate has been accurate to within 30–40 Elo when comparing the average adjusted rating to the aggregate IPR performance measures of the players. This has worked even for tournaments with players of all ages, once I've been told the numbers of teens and preteens (or are given the FIDE player IDs so my scripts can look up the birth years automatically), in Greece, Iran, Kazakhstan, Indochina, South America, the US, everywhere. Thus I have effectively assumed responsibility on the part of FIDE to provide accurate ratings for its players, likewise for US scholastic tournaments using US Chess Federation ratings, in accusations of unfair play and perhaps further cases where the rating affects qualifications.

Another juncture of sudden need and human research has been determining the exact curve by which playing quality is impacted by having less thinking time. For in-person chess the only sources of large data besides usual slow time controls have been the 25 minutes allotted (to reach the standard milestone of turn 60) by the World Rapid Championships and the 5 minutes of the World Blitz Championships, plus some smaller tournaments using the same formats. But online chess has been played at all manner of paces. Last month's FIDE Women's Speed Championships had segments played at 2-minute "Bullet" blitz, 4-minute blitz, and 6-minute blitz; while the Paris Grand Chess Tour Blitz allowed 7 minutes while blitz tiebreaks in the Magnus Carlsen Chess Tour give 8 minutes. This spring alone I've been called upon for Rapid at 10, 12, 15, 17, 20, 25, 30, 35, 37, 40, and 45-minute paces—plus many game-60 and game-75 events, which count for slow-chess ratings but afford half the thinking time. I undertook to interpolate and found that two disparate methods converged to the same curve. I put this out on the GLL blog as the third section of my Election Eve article[4], which had over 1,200 views. My Intrinsic Performance Rating calculations on clean events (or after cleaning them) show the curve's projections to be accurate within 30–40 Elo, weakening to 50–75 only for chess at 3-minutes to turn 60 or less, for players at all levels not just the elite players of my in-person data. I am the first to produce such a curve of quality versus thinking time, but before I can even think of writing a paper, I have been pressed to employ it in my cheating tests in "post-normal science" fashion—as with much else during the pandemic. Even when I have had to combine these two adjustments, such as for 15-minute Rapid chess in K-7 vis-à-vis K-12 scholastic sections, they have been accurate within an order of magnitude less than the size of the adjustments themselves.

[4]https://rjlipton.wpcomstaging.com/2020/11/03/the-election-night-time-warp/

My model has two main principles. The first has been on my website in boldface unchanged since 2007, and I have referenced it many times while expounding innocence in cases where others have noticed the high computer concordance of a series of moves—including a high-profile instance last week. The second was core content of Biswas's 2016 thesis but took three-plus years to implement in a stable manner amid the model's double-log/double-exponential digital dynamics.

1. A move that is given a *clear standout evaluation* by a program is much more likely to be found by a strong human player than one with alternatives of nearly equal value.

2. Weaker players do not expressly prefer weaker moves, but rather are more often "diverted by shiny objects."

I had hoped to make my third main model parameter denote depth of thinking directly (or gullibility on the flip side), but I could only make it indirectly reflect Biswas's adaptation of Herbert Simon's idea of *satisficing* to analyze *when* a player stops thinking and acts by making a move. This improved prediction accuracy by 2-to-3 percentage points, which may not sound like much but has increased z-scores by about 0.50 in typical cheating cases. My model now represents every cognitive trait I know to affect skill at chess. It does not represent any chess-specific elements. In principle, it can be built in like manner for any strategy game for which the values of a player's options can be authoritatively estimated by computer. The fact of computers finally reigning at Go should enable my model to be built to predict fallible human play at Go in "from-zero" fashion—that is, given only a large database of games played by humans at all skill levels and the computer values of moves (as they change at different depths of search), and with the rules and nature of the game serving only to exclude illegal moves from the input.

Given that the only inputs are values under functionally perfect rationality, the scientific essence is human performance under *bounded rationality*—which in turn is a facet of computational complexity. Thus my work engages with cognitive science in the large, and carries a similar import to *AlphaZero* that minimal input without any case-based knowledge representation or supervision of learning suffices to predict human brain functions with deployable accuracy. My invited contribution "Rating Computer Science Via Chess" [53] to the Springer Lecture Notes in Computer Science 10,000 anniversary issue covered developments in computer chess since the start of LNCS, but used my model at the end to measure the progress intrinsically. This yields a parallel to Moore's Law that embraces software as well as hardware advances. The bottleneck I face is how to transfer this deployment outside chess. For comparison, it is not yet clear whether the "-Zero" paradigm has started achieving better success in general fields than IBM's *Watson* has. In any event, if the doing of science is the making of falsifiable projections that come true over 90% of the time, I could attest this by a lot of annotated spreadsheets and before-and-after communications with chess officials around the globe.

A Still-Higher Ambition

I'll end with speculation that conveys more of my scientific outlook and reaching after fundamentals. I regard complexity theory as having a second holy grail besides $\mathsf{P} \neq \mathsf{NP}$ and lower bounds in general. This is to develop a concretely tractable measure of (pseudo-)randomness. To exemplify concreteness, let us take strings of 50,000 bits and try to distinguish those that can be generated from 150-bit seeds by a highly succinct rule from those that are truly random—or require 300 bits at least. The numbers 150 and 300 flank the square root of 50,000.

Theory holds that there are efficient pseudorandom generators for which the distinguishing task scaled for large n in place of 50,000 is hard asymptotically. An algorithm succeeding at the concrete case would have to be exponential time in principle. But many exponential-time algorithms work concretely well at this scale much of the time—consider the aforementioned *sharpSAT* and *Cachet* and Gröbner basis algorithms used for scientific computation quite in general. The field has not developed concrete analysis that can tell which concrete tasks are tractable, while tasks that matter all the time in cryptography keep having to raise the applicable value of n for security. There is nothing that opposes the possibility that a highly tuned exponential-time algorithm can work here.

I have been imbued with one genre of highly tuned exponential-time algorithms that have arguably received many more person-hours of development and competitive commercial testing than the likes of *sharpSAT* or *Cachet*: chess-playing programs. They all employ the same tabulation hashing scheme originally developed for Go-playing programs by Zobrist [61] and analyzed generally by Patrascu and Thorup [39, 40]. For chess this requires just under 50,000 bits to fix a set of 781 hash codes of 64 bits each. Those bits should be truly random—this is not a situation of consuming millions of bits per second for molecular simulations where pseudorandom generators (PRGs) are needed for scale—but many chess programs still generate them via PRGs. The ambitious question is, *can the difference between random and pseudorandom be detected in the behavior of the chess program?*

I laid out a testable mechanism for this detection via hash collisions of the kind that have caused programs to err in actual tournaments, and gave a chess position that generates reproducible ones.[5] I have found a few other such positions, including one constructed by the artist and master player Marcel Duchamp.[6] It is not enough to count collisions—the test must leverage the exponential search power of the chess program in order not to contravene theory about both complexity and PRGs that pass all linear tests. The errors caused by bad values propagating to the top level of the search can be automatically verified by majority vote of other programs. This is an *in situ* test of a kind considered generally desirable but hard to implement; it addresses the most subtle kinds of unwanted regularities that could emerge from the output of PRGs.

The obstacles to executing the experiment in full are the need to find many more critical positions, the computing time needed to run a large suite of tests, and perhaps most of all, the prospect of needing to establish that a tiny deviation from expectation is significant. I have not been able to take time to try it. I did once undertake computational experiments

[5] https://rjlipton.wpcomstaging.com/2012/05/04/digital-butterflies-and-prgs/

[6] https://rjlipton.wpcomstaging.com/2018/02/16/a-coupe-of-duchamp/

with resource-bounded information complexity [31], but while the results were termed positive in a later edition of the noted text by Li and Vitanyi [32], I felt them inconclusive, and later regarded the larger predicate we were testing as countermanded by Mitzenmacher et al. [7].

To come all the way back to logic, the idea falls across the theorem that a formal system whose axioms have 150 bits of entropy cannot prove any string to have appreciably more than 150 bits of entropy. But I believe the question well deserves to be posed. It may be unmatched by results but speaks scientific earnestness, original synthesis, and diversity of source experience. At least it belongs to the stream of ideas that Lipton generates even more, and exemplifies how the GLL blog helps to promote research even if only one or two of the ideas ultimately catch fire.

References

[1] S. Aaronson. Is P versus NP formally independent? *Bull. EATCS*, pages 109–136, 2003. Computational Complexity Column 81 edited by L. Fortnow, viewable at http://people.cs.uchicago.edu/~fortnow/beatcs/column81.pdf.

[2] S. Aaronson and A. Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computing Theory*, 1(1), 2009.

[3] J.-M. Alliot. Who is the master? *ICGA Journal*, 39:1–40, 2017.

[4] D. Bacon, W. van Dam, and A. Russell. Analyzing algebraic quantum circuits using exponential sums. http://www.cs.ucsb.edu/ vandam/LeastAction.pdf, November 2008.

[5] W. Baur and V. Strassen. The complexity of partial derivatives. *Theor. Comp. Sci.*, 22:317–330, 1982.

[6] H. Buhrman, D. van Melkebeek, K. Regan, D. Sivakumar, and M. Strauss. A generalization of resource-bounded measure, with application to the BPP vs. EXP problem. *SIAM J. Comput.*, 30:576–610, 2001.

[7] K.-M. Chung, M. Mitzenmacher, and S. Vadhan. Why simple hash functions work: Exploiting the entropy in a data stream. *Theory of Computing*, 9:897–945, 2013.

[8] C. Dawson, H. Haselgrove, A. Hines, D. Mortimer, M. Nielsen, and T. Osborne. Quantum computing and polynomial equations over the finite field Z_2. *Quantum Information and Computation*, 5:102–112, 2004.

[9] R. DeMillo and R. Lipton. Some connections between mathematical logic and complexity theory. In *Proc. 11th Annual ACM Symposium on the Theory of Computing*, pages 45–57, 1979.

[10] R. DeMillo and R. Lipton. The consistency of "P=NP" and related problems with fragments of number theory. In *Proc. 12th Annual ACM Symposium on the Theory of Computing*, pages 45–57, 1980.

[11] G. DiFatta, G.M^{c}C. Haworth, and K. Regan. Skill rating by Bayesian inference. In *Proceedings, 2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09), Nashville, TN*, pages 89–94, March 30–April 2 2009.

[12] R. Downey, M. Fellows, and K. Regan. Parameterized circuit complexity and the w hierarchy. *Theor. Comp. Sci., Ser. A*, 191:97–115, 1998.

[13] R. Downey, M. Fellows, and K. Regan. Threshold dominating sets and an improved characterization of W[2]. *Theor. Comp. Sci., Ser. A*, 209:123–140, 1998.

[14] A. Ehrenfeucht and M. Karpinski. The computational complexity of (XOR, AND)-counting problems. Technical Report TR-90-032, Mathematical Sciences Research Institute, University of California at Berkeley, 1990.

[15] R. Fagin, J. Lenchner, K. Regan, and N. Vyas. Multi-structureal games and number of quantifiers. In *Proc. 36th Annual IEEE Conference on Logic in Computer Science*, 2021. ArXiv: https://arxiv.org/abs/2104.14709.

[16] D. Ferreira. Determining the strength of chess players based on actual play. *ICGA Journal*, 35(1):3–19, March 2012.

[17] D. Ferreira. The impact of search depth on chess playing strength. *ICGA Journal*, 36(2):67–80, June 2013.

[18] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer Verlag, 2006.

[19] F. Green, J. Köbler, K. Regan, T. Schwentick, and J. Torán. The power of the middle bit of a #P function. *J. Comp. Sys. Sci.*, 50:456–467, 1995.

[20] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems (preliminary report). In *Proc. 25th Annual IEEE Symposium on Foundations of Computer Science*, pages 495–503, 1984.

[21] C. Guan and K. Regan. Stabilizer circuits, quadratic forms, and computing matrix rank. https://arxiv.org/abs/1904.00101, 2019.

[22] C. Guan and K. Regan. A new graph polynomial and generalized tutte-grothendieck invariant from quantum circuits. In Chaki et al., editor, *Advanced Computing and Systems for Security*, pages 3–16. Springer Verlag, 2020.

[23] M. Guid and I. Bratko. Computer analysis of world chess champions. *ICGA Journal*, 29(2):65–73, 2006.

[24] M. Guid and I. Bratko. Using heuristic-search based engines for estimating human skill at chess. *ICGA Journal*, 34(2):71–81, 2011.

[25] M. Guid, A Pérez, and I. Bratko. How trustworthy is Crafty's analysis of world chess champions? *ICGA Journal*, 31(3):131–144, 2008.

[26] J. Harmanis and J. Hopcroft. Independence results in computer science. *SIGACT News*, 8:13–24, 1976.

[27] J. Hartmanis. Relations between diagonalization, proof systems, and complexity gaps. *Theor. Comp. Sci.*, 8:224–233, 1979.

[28] G. Haworth, T. Biswas, and K. Regan. A comparative review of skill assessment: Performance, prediction and profiling. In *Proceedings of the 14th ICGA* Advances in Computer Games *conference, Leiden, Netherlands, July 2015*, Lect. Notes Comp. Sci. Springer-Verlag, 2015. to appear.

[29] G.M^{c}C. Haworth, K. Regan, and G. DiFatta. Performance and prediction: Bayesian modelling of fallible choice in chess. In *Proceedings, 12th ICGA Conference on Advances in Computer Games, Pamplona, Spain, May 11–13, 2009*, volume 6048 of *Lecture Notes in Computer Science*, pages 99–110. Springer-Verlag, 2010.

[30] N. Immerman. Number of quantifiers is better than number of tape cells. *J. Comp. Sys. Sci.*, 22:384–406, 1981.

[31] A. Jagota and K. Regan. Performance of neural net heuristics for Maximum Clique on diverse highly compressible graphs. *Journal of Global Optimization*, 10:439–465, 1997. A shorter version appeared in the proceedings of the 1996 International Conference on Neural Information and Processing (ICONIP'96), Hong Kong, September 1996.

[32] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications, Third Edition*. Springer Verlag, 2008.

[33] R. Lipton and K. Regan. *Quantum Algorithms via Linear Algebra*. MIT Press, 2014.

[34] R.J. Lipton and K.W. Regan. *Introduction to Quantum Algorithms Via Linear Algebra, Second Edition*. MIT Press, 2021.

[35] E. Mayr and A. Meyer. The complexity of the word problem for commutative semigroups and polynomial ideals. *Advances in Math.*, 46:305–329, 1982.

[36] D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, 1973.

[37] A. Naik, K. Regan, and D. Sivakumar. On quasilinear time complexity theory. *Theor. Comp. Sci.*, 148:325–349, 1995.

[38] M.A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[39] M. Patrascu and M. Thorup. On the k-independence required by linear probing and minwise independence. In *Proc. 37th Annual International Conference on Automata, Languages, and Programming*, volume 6198 of *Lect. Notes in Comp. Sci.*, pages 715–726. Springer Verlag, 2010.

[40] M. Patrascu and M. Thorup. The power of simple tabulation hashing. *Journal of the Association of Computing Machinery*, 59:1–50, 2012.

[41] A. Razborov and S. Rudich. Natural proofs. In *Proc. 26th Annual ACM Symposium on the Theory of Computing*, pages 204–213, 1994.

[42] A. Razborov and S. Rudich. Natural proofs. *J. Comp. Sys. Sci.*, 55:24–35, 1997.

[43] K. Regan. On the Separation of Complexity Classes, 1986. Dissertation, Oxford University.

[44] K. Regan. A topology of provability in complexity theory. In *Proc. 1st Annual IEEE Conference on Structure in Complexity Theory*, volume 223 of *Lect. Notes in Comp. Sci.*, pages 291–310. Springer Verlag, 1986.

[45] K. Regan. A uniform reduction theorem, extending a result of J. Grollmann and A. Selman. In *Proc. 13th Annual International Conference on Automata, Languages, and Programming*, volume 226 of *Lect. Notes in Comp. Sci.*, pages 324–333. Springer Verlag, 1986.

[46] K. Regan. Unprovably intractable languages. In *Proc. 2nd Annual IEEE Conference on Structure in Complexity Theory*, pages 69–80, 1987.

[47] K. Regan. Finitary substructure languages. In *Proc. 4th Annual IEEE Conference on Structure in Complexity Theory*, pages 87–96, 1989.

[48] K. Regan. Diagonalization, uniformity, and fixed-point theorems. *Inform. and Comp.*, 98:1–40, 1992.

[49] K. Regan. A new parallel vector model, with exact characterizations of NC^k. In *Proc. 11th Annual Symposium on Theoretical Aspects of Computer Science*, volume 778 of *Lect. Notes in Comp. Sci.*, pages 289–300. Springer Verlag, 1994.

[50] K. Regan. Linear time and memory-efficient computation. *SIAM J. Comput.*, 25:133–168, 1996.

[51] K. Regan. Polynomial vicinity circuits and nonlinear lower bounds. In *Proceedings, 12th IEEE Conference on Computational Complexity (formerly Structure in Complexity Theory), Ulm, Germany, June*, pages 61–68, New York, 1997. IEEE.

[52] K. Regan. Understanding the Mulmuley-Sohoni approach to P vs. NP. *Bull. EATCS*, 78:86–97, October 2002. Invited contribution to Lance Fortnow's Computational Complexity Column.

[53] K. Regan. Rating computer science via chess. In *Computing and Software Science*, volume 10,000 of *Lect. Notes in Comp. Sci.*, pages 200–216. Springer Verlag, 2019.

[54] K. Regan and T. Biswas. Psychometric modeling of decision making via game play. In *Proceedings,IEEE Conference on Computational Intelligence in Games, Niagara Falls, Canada*, August 2013.

[55] K. Regan, A. Chakrabarti, and C. Guan. Algebraic and logical emulations of quantum circuits. *Transactions on Computational Science*, 10,730:41–76, 2018.

[56] K. Regan and G. Haworth. Intrinsic chess ratings. In *Proceedings of AAAI 2011, San Francisco*, 2011.

[57] K. Regan and T. Schwentick. On the power of one bit of a #P function. In *Proc. 4th Annual Italian Conference on Theoretical Computer Science*, pages 317–329. World Scientific, Singapore, 1992.

[58] K. Regan, D. Sivakumar, and J.-Y. Cai. Pseudorandom generators, measure theory, and natural proofs. In *Proc. 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 26–35, 1995.

[59] K. Regan and J. Wang. The quasilinear isomorphism challenge. *SIGACT News*, 25:106–113, September 1994.

[60] U. Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theor. Comp. Sci.*, 18:95–103, 1982.

[61] A. Zobrist. A new hashing method with application for game playing. Technical Report 88, Computer Sciences Department, University of Wisconsin, Madison, 1969.