RESEARCH STATEMENT

KENNETH GILL

My research is in computability theory (recursion theory), a subfield of mathematical logic related to theoretical computer science. Computability theory is broadly concerned with quantifying the level of computing power needed to describe a given mathematical object. My current work is divided into two main projects, each exploring this general idea in different ways. The first introduces a way to measure the complexity of finite strings via probabilistic finite-state automata. The second involves gauging the intrinsic uniform computational and combinatorial difficulty of solving a given problem using the framework of Weihrauch complexity, focusing on certain principles in infinite Ramsey theory.

1. Probabilistic automatic complexity

The Kolmogorov complexity of a finite string x is, loosely speaking, the least number of states of a Turing machine which outputs x given no input. This quantity is well-known to be noncomputable and defined only up to an additive constant, and this has motivated several authors to investigate complexity measures based on the difficulty of describing x via a less powerful model of computation. In 2001, Jeffry Shallit and Ming-wei Wang introduced the deterministic finite-state automatic complexity $A_D(x)$, which they defined to be the least number of states of a DFA which uniquely accepts x among strings of the same length [19]. This is computable, well-defined, and there is a polynomial-time algorithm to recover x given $|x|$ and a witness for $A_D(x)$. Kayleigh Hyde defined an analogous quantity $A_N(x)$ based on nondeterministic finite automata (NFAs), which has some additional desirable properties [12].

In this project, I introduce a new complexity measure in the same spirit as A_D and A_N but based on probabilistic finite-state automata (PFAs). A PFA is an automaton where the state transitions are given probabilities, and each word x is given a probability of acceptance $\rho(x)$ rather than a binary acceptance or rejection. (In the machine learning literature, what I call PFAs are often referred to as probabilistic or stochastic acceptors, and the term PFA more commonly denotes a transducer with both input and output behavior.)

The point of view adopted here is that a PFA is a good description of x if x is the most likely string to be accepted among those of the same length:

Definition 1.1 ([8])**.** The PFA complexity of a string x, $A_P(x)$, is the least number of states of a PFA accepting x with the unique highest probability among strings of the same length.

We have $A_P(x) \leq A_N(x) + 1$ for all x. No example is known where equality holds aside from the constant strings. Indeed, $A_P(x) \leq 3$ for all binary x with $|x| \leq 9$, which is far from the case for A_N. The most significant result obtained about A_P so far is a complete classification of the binary strings with $A_P = 2$ [8]:

Theorem 1.2. *For a binary string x, $A_P(x) = 2$ if and only if x is of the form*

$$i^n j^m, \qquad i^n j^m i, \qquad i^n (ji)^m, \qquad \textit{or} \quad i^n j(ij)^m \qquad \textit{for some } n, m \geq 0.$$

This represents a striking level of compression over the NFA complexity. As shown by Hyde in [12], for any string x, $A_N(x) = 2$ iff x is of the form

$$(ij)^m, \qquad i^m j, \qquad \text{or} \quad ij^m.$$

Observe that the theorem implies in particular that the set of binary strings with $A_P = 2$ is a regular language. Its proof also demonstrates that for any 2-state PFA M over a binary alphabet, the set of strings whose complexity is witnessed by M is a regular (usually infinite) language. This stands in contrast to the accepted language of a PFA with specified cut-point $\lambda \in [0, 1]$, i.e., the set of all strings x with $\rho(x) > \lambda$, which depending on the PFA and the cut-point can be at any level of the Chomsky hierarchy, and may not even be recursively enumerable.

The proof of Theorem 1.2 exploits a correspondence between PFAs and iterated function systems (IFSs). It involves an exhaustive analysis of the dynamics of the IFSs associated with two-state automata, which would almost certainly be infeasible to imitate for PFAs with more than two states, or even for 2-state PFAs over a larger alphabet. The original motivation of the classification theorem was to show that A_P is not constant, but it is still unknown whether it is unbounded:

Question 1.3. Is A_P unbounded? If not, what is its maximum value (for binary strings or over any other alphabet)?

A possible approach to showing A_P unbounded would be to attempt a similar analysis of the dynamics of an IFS associated to a PFA as in the proof of Theorem 1.2, but with much less granularity. One would aim to prove that for any k, the binary strings whose complexity is witnessed by a k-state PFA must be of a certain general form which does not fit every string. I intend to investigate combinatorial ideas which could make such a proof tractable, or if none present themselves, to pursue Question 1.3 through other means.

PFA complexity with required gap: In general it could be that a witness for $A_P(x)$ does not distinguish x very well from other strings in the sense that the acceptance probability of x is not much larger than that of others of the same length. To rectify this, we can define another measurement $A_{P,\delta}(x)$ that builds in a required minimum gap δ in probabilities, where δ is a real-valued parameter:

Definition 1.4 ([8])**.** The PFA complexity of x with gap δ, $A_{P,\delta}(x)$, is the least number of states of a PFA such that $\rho(x) - \delta \geq \rho(y)$ whenever $|y| = |x|$ and $y \neq x$. Here δ may be any element of $(0, 1]$.

We have $A_P(x) \leq A_{P,\delta}(x) \leq A_D(x)$ for all x and $\delta \in (0, 1]$. In general it seems that $A_{P,\delta}$ is harder to work with than A_P: the proof of Theorem 1.2, for example, is not recoverable in any obvious way for $A_{P,\delta}$ since it relies in part on completely disregarding the actual sizes of probabilities, only looking at the pattern of maximum and minimum probabilities among strings of each length. Due to the stabilization of all orbits of any IFS associated with a generic PFA, the acceptance probabilities of longer and longer words with respect to the same PFA tend to cluster together. Thus for any given δ, a generic PFA is expected to witness $A_{P,\delta}(x)$ for only finitely many x. The proof of Theorem 1.2 shows this is far from the case for A_P, e.g., the same PFA may witness the complexity of the strings $0^n 1^m$ for a fixed n and for all $m \geq 0$. This seems to confer a descriptive advantage to A_P since a single PFA can be viewed as describing an infinite family of strings with similar structure.

However, $A_{P,\delta}$ does have one distinct advantage over A_P, or at least appears to at the time of writing. The computability of $A_D(x)$ was a primary motivation for its introduction.

One would hope that A_P also turns out to be computable, but so far this is unknown. On the other hand, while $A_{P,\delta}(x)$ is not everywhere continuous, it is continuous on a co-countable set and also computable there [8]:

Theorem 1.5. *For any finite alphabet Σ, the function from $[0,1]\times\Sigma^*$ to $\mathbb{N}$ given by $(\delta, x) \mapsto A_{P,\delta}(x)$ is computable where it is continuous. It is discontinuous only on a countable set whose elements can be computed by a single algorithm. In particular, for every x, $A_{P,\delta}(x)$ is computable for all but at most $A_D(w) - 1$ many values of δ.*

The proof of this theorem does not extend in any evident way to a proof that A_P is computable. I currently suspect that A_P is in fact noncomputable, and would like to study possible routes to a proof. For example, one could view the computation of A_P as a decision problem and reduce it to another decision problem known to be undecidable. A number of such decision problems related to PFAs have been studied [5], but it is for now unclear whether any of them are amenable to such a reduction.

My research into $A_{P,\delta}$ in the near term will focus on finding out what general structure $A_{P,\delta}$ appears to capture in finite strings as compared with A_P, especially as δ varies. It would also be interesting to see whether $A_{P,\delta}$ is unbounded, even if A_P is not. Finally, a few other ideas for modifying the definition of A_P are presented in [8], and I would like to investigate their properties further, along with further possible complexity measures. In particular, one idea involves a real-valued notion of complexity $A_\mu(x)$, where μ is a Borel probability measure on the space of PFAs, that takes into account the μ-measure of the set of witnesses for $A_P(x)$ in an attempt to refine the numerical measurement itself. A quantity $A_{\mu,\delta}$ analogous to $A_{P,\delta}$ can be defined as well.

Question 1.6. What is a natural measure μ which would make a result analogous to Theorem 1.5 hold for $A_{\mu,\delta}$?

2. Indivisibility and Weihrauch reducibility

My other project addresses the rather broad question of “how difficult is it to solve a mathematical problem?” A little less vaguely, “given a hypothetical machine capable of solving any instance of some problem, how powerful must the machine be?” To quantify this we formalize a problem as a partial multivalued function on $\mathbb{N}^\mathbb{N}$ viewed as mapping instances to solutions; an example would be the problem RT^1_k, representing the pigeonhole principle with countably many pigeons and k pigeonholes, which maps a k-coloring of $\mathbb{N}$ to an infinite monochromatic set. We then measure the intrinsic difficulty of solving a problem with *Weihrauch reducibility*, a partial order on problems. If P and Q are problems, then P is Weihrauch reducible to Q (written $P \leq_\mathrm{W} Q$) if every instance of P can be solved by a single application of Q, together with the application of a single algorithm to translate between the problems (see [1] for a precise definition).

This notion can be viewed as capturing the uniform computational content of a problem, because the algorithm in the definition cannot depend on the particular instances or solutions of P or Q and so must reflect the intrinsic combinatorics of P and Q. Weihrauch reducibility is thus a uniform, resource-sensitive refinement of the general program of reverse mathematics [20], which seeks to classify mathematical statements according to which other theorems are needed to prove them, and which other statements they imply. In reverse mathematics, principles P and Q are compared only in terms of provability over a weak base system for mathematics, without regard to the uniform computability of the proof or of the number of applications of Q needed to prove P. Weihrauch reducibility can reveal subtle distinctions between combinatorial principles that are not otherwise apparent through a reverse-mathematical lens.

I have focused in particular on indivisibility problems in this project. A structure is *indivisible* if for any coloring of the structure with finite range, there is a monochromatic subset isomorphic to the whole structure. Indivisibility is naturally part of Ramsey theory, the study of objects which cannot be "too disordered" in that any large subset must recover the structure of the whole object. For instance, the infinite pigeonhole principle (Ramsey's theorem for singletons) is the statement that $\mathbb{N}$ is indivisible. $\mathbb{Q}$ is also indivisible, as are the Rado graph and many other objects. Any indivisible structure $\mathcal{S}$ corresponds to an *indivisibility problem* $\mathsf{Ind}\,\mathcal{S}$. For example, $\mathsf{Ind}\,\mathbb{Q}$ is the problem whose instances are colorings c of $\mathbb{Q}$ with finite range, and whose solutions to c are all c-monochromatic sets order-isomorphic to $\mathbb{Q}$. We can also consider the problems $\mathsf{Ind}\,\mathbb{Q}_k$ whose instances are restricted to k-colorings.

The rational numbers: $\mathsf{Ind}\,\mathbb{Q}$ is a problem which seems to lie "off to the side" in the Weihrauch lattice: it has little apparent uniform strength itself, yet seems to require a fair amount of uniform strength from other problems which solve it. For a start, limit computable problems are insufficiently strong:

Proposition 2.1. *$\mathsf{Ind}\,\mathbb{Q}_2$ cannot be uniformly solved by any limit computable problem. That is, $\mathsf{Ind}\,\mathbb{Q}_2 \not\leq_{\mathrm{W}} P$ for any P for which there is an algorithm computing, from any instance of P, a sequence in Baire space $\mathbb{N}^{\mathbb{N}}$ whose limit is a solution.*

An example of a limit computable problem would be the problem mapping a sequence of real numbers to its supremum. The general idea that $\mathsf{Ind}\,\mathbb{Q}$ is too difficult to solve from finitary information, in a sense, is further demonstrated by the following result. I call a problem P *pointwise c.e. traceable* (p.c.e.t.) if there is an algorithm that, given any instance p of P, outputs a finite list of guesses for solutions to p, at least one of which is correct [7]. The class of p.c.e.t. problems includes any problem with a fixed finite set of possible natural-number solutions, such as a problem which can answer any finite list of yes/no questions of a given level of complexity.

Theorem 2.2 ([7])**.** *No p.c.e.t. problem can uniformly solve $\mathsf{Ind}\,\mathbb{Q}_2$.*

On the other hand, $\mathsf{Ind}\,\mathbb{Q}$ seems to have little uniform strength. The strongest problem $\mathsf{Ind}\,\mathbb{Q}_k$ is currently known to solve is RT^1_k, and the reduction follows trivially by identifying rationals with naturals, not using any of the apparent additional power of $\mathsf{Ind}\,\mathbb{Q}_k$. Indeed, there are problems computable with finitely many mind changes which $\mathsf{Ind}\,\mathbb{Q}$ cannot solve. This property is exactly characterized by Weihrauch reducibility to the problem $\mathsf{C}_{\mathbb{N}}$, which outputs an element of a nonempty subset of $\mathbb{N}$ given an enumeration of its complement, and we have

Theorem 2.3 ([7])**.** *For all $k \geq 2$, $\mathsf{C}_{\mathbb{N}} \not\leq_{\mathrm{W}} \mathsf{Ind}\,\mathbb{Q}_k$.*

This theorem can be used to relate $\mathsf{Ind}\,\mathbb{Q}$ to another family of problems involving colorings of the rationals. The problem $\mathsf{iShuffle}_k$ maps a k-coloring of $\mathbb{Q}$ to a code for an interval in which every color appearing is dense [18]. $\mathsf{iShuffle}$ performs the same task but can handle any given number of colors k, rather than being restricted to a fixed k. It is known that $\mathsf{iShuffle}_2$ is strong enough to solve $\mathsf{C}_{\mathbb{N}}$ and hence also any problem computable with finitely many mind changes. Then as a corollary of the last theorem, $\mathsf{Ind}\,\mathbb{Q}$ cannot solve $\mathsf{iShuffle}$: finding a monochromatic densely ordered set is not sufficient to uniformly compute an interval where every color is dense. By further results of [18], in fact, $\mathsf{Ind}\,\mathbb{Q}$ is not strong enough to find an interval where even *one* color is dense. And indeed $\mathsf{iShuffle}_2 \not\leq_{\mathrm{W}} \mathsf{Ind}\,\mathbb{Q}_k$ for all k, meaning that even if it is limited to two colors, $\mathsf{iShuffle}$ cannot be solved by adding more colors to $\mathsf{Ind}\,\mathbb{Q}$.

Putting these and other known results together, we obtain the following diagram, in which an arrow from P to Q means that $P <_{\mathrm{W}} Q$. No reducibilities hold other than those implied by transitivity of $\leq_{\mathrm{W}}$.

$$\begin{tikzcd}
\mathsf{iShuffle}_2 \arrow[r] & \mathsf{iShuffle}_3 \arrow[r] & \mathsf{iShuffle}_4 \arrow[r] & \mathsf{iShuffle}_5 \arrow[r] & \cdots \\
\mathsf{Ind}\,\mathbb{Q}_2 \arrow[r] \arrow[u] & \mathsf{Ind}\,\mathbb{Q}_3 \arrow[r] \arrow[u] & \mathsf{Ind}\,\mathbb{Q}_4 \arrow[r] \arrow[u] & \mathsf{Ind}\,\mathbb{Q}_5 \arrow[r] \arrow[u] & \cdots \\
\mathsf{RT}^1_2 \arrow[r] \arrow[u] & \mathsf{RT}^1_3 \arrow[r] \arrow[u] & \mathsf{RT}^1_4 \arrow[r] \arrow[u] & \mathsf{RT}^1_5 \arrow[r] \arrow[u] & \cdots
\end{tikzcd}$$

Solutions of $\mathsf{Ind}\,\mathbb{Q}$ are rather nebulous: for example, if x and y are two rational numbers in a solution S, one can delete the whole interval $[x, y]$ from S and still obtain a solution. The seeming weakness of $\mathsf{Ind}\,\mathbb{Q}$ may be a consequence of this nebulosity, and it is unclear how much power (if any) $\mathsf{Ind}\,\mathbb{Q}$ derives from the fact that it outputs a second-order object, i.e., a set of rationals. An investigation of these properties can be put on precise footing with the following notions [9, 3]:

- The *first-order part* of a problem P is the strongest problem reducible to P that has codomain $\mathbb{N}$.
- The *deterministic part* of P is the strongest problem reducible to P for which every instance has a unique solution.

Question 2.4. How strong exactly are the first-order and deterministic parts of $\mathsf{Ind}\,\mathbb{Q}$?

The countable equivalence relation: Let $\mathscr{E}$ be, up to isomorphism, the equivalence relation with countably many equivalence classes each having countably many elements. Then $\mathscr{E}$ is indivisible (folklore). The Weihrauch degree of $\mathsf{Ind}\,\mathscr{E}_2$ turns out to lie strictly between those of SRT^2_2 and RT^2_2, two heavily studied problems in reverse mathematics whose full logical separation was a major open problem until recently [17]. In fact the same holds for any number of colors [7]:

Theorem 2.5. *$\mathsf{SRT}^2_k <_{\mathrm{W}} \mathsf{Ind}\,\mathscr{E}_k <_{\mathrm{W}} \mathsf{RT}^2_k$ for all $k \geq 2$.*

The proof shows in particular that $\mathsf{RT}^2_2 \not\leq_c \mathsf{Ind}\,\mathscr{E}_k$ for all k, where $\leq_c$ denotes the weaker nonuniform computable reducibility. (See for example [11] for a definition of $\leq_c$.) In addition, $\mathsf{Ind}\,\mathscr{E}_2 \not\leq_{\mathrm{W}} \mathsf{SRT}^2_k$ for all k. This produces a hierarchy of strict reducibilities very much like the one shown above for RT^1_k, $\mathsf{Ind}\,\mathbb{Q}_k$, and $\mathsf{iShuffle}_k$.

For problems P and Q, $P \leq_{\mathrm{W}} Q$ implies $P \leq_c Q$, but not conversely. To more fully understand the nature of the combinatorial distinction between $\mathsf{Ind}\,\mathscr{E}_k$ and SRT^2_k, one could then ask

Question 2.6. Is the reducibility $\mathsf{SRT}^2_k \leq_c \mathsf{Ind}\,\mathscr{E}_k$ strict (for any or all k)?

One of the most celebrated results of reverse mathematics is that RT^2_2 has a sort of "logical orthogonal decomposition" into $\mathsf{SRT}^2_2 + \mathsf{COH}$, where COH is the so-called cohesive principle [2]. In particular, COH and SRT^2_2 are Weihrauch incomparable and both are Weihrauch reducible to RT^2_2. It is possible to view $\mathsf{Ind}\,\mathscr{E}_2$ as a variant of COH and I want to know what the precise relationship between the two problems is.

The Henson graphs and Turing degrees of solutions: For each $n \geq 3$, the Henson graph H_n is the universal countable K_n-free graph, where we write K_n for the complete graph on n vertices [10]. Both the Rado graph and the Henson graphs are known to be indivisible [16, 4]. The Rado graph is computably indivisible, in the sense that a solution always exists which is computable in the instance; the same is true for $\mathbb{Q}$ (both statements are folklore). It turns out that the Henson graphs do not share this property [6]:

Theorem 2.7. *For every $n \geq 3$, there is a computable 2-coloring of H_n for which no c.e. monochromatic subcopy of H_n exists.*

No study has yet been made of the Weihrauch degrees of $\mathsf{Ind}(H_n)_k$ for any n or k. But the above theorem naturally raises the question of how difficult it is in general to even *nonuniformly* compute a solution to $\mathsf{Ind}(H_n)$. The proof by Komjáth and Rödl of the indivisibility of H_3 [16] shows that there is always a solution to $\mathsf{Ind}(H_3)_2$ computable in the first jump of the instance. The general indivisibility proof of El-Zahar and Sauer [4] gives a strictly greater upper bound for $n = 3$, and it is unknown whether a similar gap exists for $n \geq 4$. Furthermore, whether or not any noncomputable information can be uniformly encoded in solutions of $\mathsf{Ind}(H_n)$, for any n, remains open, and I would like to systematically investigate this question starting with $n = 3$.

Other directions: Out of the many directions I would like to take this project in, I single out the idea of polynomial-time Weihrauch reduction, first explored by Kawamura and Cook [13, 14] in the context of computable analysis. Although their framework does not translate exactly to the problems I have mentioned due to extra stipulations they make about the domains of problems, I am generally interested in how the added requirement of an efficient algorithm implementing the reduction might reveal further distinctions between the intrinsic combinatorics of these and other problems.

References

[1] V. Brattka, G. Gherardi, and A. Pauly, Weihrauch complexity in computable analysis. In *Handbook of computability and complexity in analysis* (V. Brattka and P. Hertling, eds.), Springer, 2021, pp. 367–417. arXiv: `1707.03202`.

[2] P. Cholak, C. G. Jockusch Jr., and T. Slaman, On the strength of Ramsey's theorem for pairs, *J. Symbolic Logic* **66** (2001), 1–55. Corrigendum in *J. Symbolic Logic* **74** (2009), 1438–1439.

[3] D. Dzhafarov, R. Solomon, and K. Yokoyama, On the first-order parts of problems in the Weihrauch degrees, preprint (2023). arXiv: `2301.12733`.

[4] M. El-Zahar and N. Sauer, The indivisibility of the homogeneous K_n-free graphs, *J. Comb. Theory B* **47** (1989), 162–170.

[5] N. Fijalkow, Undecidability results for probabilistic automata, in *ACM SIGLOG NEWS* **4**(4) (Oct. 2017).

[6] K. Gill, A note on the indivisibility of the Henson graphs, preprint (2023). arXiv: `2310.20097`.

[7] K. Gill, Indivisibility and uniform computational strength, preprint (2023). arXiv: `2312:03919`.

[8] K. Gill, Probabilistic automatic complexity of finite strings, in preparation (2023).

[9] J. L. Goh, A. Pauly, and M. Valenti, Finding descending sequences through ill-founded linear orders, *J. Symbolic Logic* **86**(2) (2021), 817–854. `doi:10.1017/jsl.2021.15`.

[10] C. Ward Henson, A family of countable homogeneous graphs, *Pacific J. Math.* **38**(1) (1971), 69–83.

[11] D. Hirschfeldt and C. G. Jockusch Jr., On notions of computability-theoretic reductions between Π^1_2 principles, *J. Math. Logic* **16**(1) (2016). Available at `http://www.math.uchicago.edu/~drh/Papers/Papers/upaper.pdf`.

[12] K. Hyde, *Nondeterministic finite state complexity*, MA thesis, University of Hawai'i, Manoa, 2013.

[13] A. Kawamura, Lipschitz continuous ordinary differential equations are polynomial-space complete, *Computational Complexity* **19**(2) (2010), 305–332. arXiv: `1004.4622`.

[14] A. Kawamura and S. Cook, Complexity theory for operators in analysis, arXiv: `1305.0453v1` (2013). Preliminary version in *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, 495–502. New York: ACM, 2010.

[15] B. Kjos-Hanssen, Maximal automatic complexity and context-free languages, preprint (2022). arXiv: `2206.10130`.

[16] P. Komjáth and V. Rödl, Coloring of universal graphs, *Graphs and Combinatorics* **2** (1986), 55–60.

[17] B. Monin and L. Patey, SRT^2_2 does not imply RT^2_2 in ω-models, *Advances in Mathematics* **389** (2021). arXiv: `1905.08427`.

[18] A. Pauly, C. Pradic, and G. Soldà, On the Weihrauch degree of the additive Ramsey theorem, preprint (2023). arXiv: `2301.02833`.

[19] J. Shallit and M.-w. Wang, Automatic complexity of strings, *J. Autom. Lang. Comb.* **6**(4) (2001), 537–554.

[20] S. Simpson, *Subsystems of second-order arithmetic*, 2nd ed. Cambridge University Press, 2009.