Unverifiability, Unexplainability & Unpredictability

Roman V. Yampolskiy
Computer Science and Engineering
University of Louisville
roman.yampolskiy@louisville.edu

Abstract
Optimistic plans of mathematicians to automatically uncover all truths have been thwarted by Gödel's Incompleteness and Turing's Undecidability among many other impossibility results. In this essay we describe a more general limitation on mathematical proofs, Unverifiability, along with Unpredictability and Unexplainability of powerful knowledge discovery agents. We conclude with analysis of limits to what we can prove, predict or understand on physics and science in general, as well as safety of artificial intelligence in particular.

1. On Observers and Verifiers

The concept of an 'observer' shows up in contexts as diverse as physics (particularly quantum and relativity), biophysics, neuroscience, cognitive science, artificial intelligence, philosophy of consciousness, and cosmology [1], but what is an equivalent idea in mathematics? We believe it is the notion of the proof verifier. Consequently, the majority of open questions recently raised [1] by the Foundational Questions Institute related to the physics of the observer could be asked about proof verifiers. In particular, the mathematical community may be interested in studying different types of proof verifiers (people, programs, oracles, communities, superintelligences, etc.) as mathematical objects, ways they can be formalized, their power and limitations (particularly in human mathematicians), minimum and maximum complexity, as well as self-verification and self-reference in verifiers[1].

Proof Theory has been developed to study proofs as formal mathematical objects consisting of axioms from which, by rules of inference, one can arrive at theorems [2]. However, the indispensable concept of the verifier has been conspicuously absent from the discussion, particularly with regards to its formalization and practical manifestation. A *verifier* in the context of mathematics is an agent capable of checking a given proof, step-by-step, starting from axioms to make sure that all intermediate deductions are indeed warranted, that the final conclusion follows, and consequently, that the claimed theorem is indeed true.

2. Historical Perspective

The field of mathematics progresses by proving theorems, which in turn serve as building blocks for future proofs of yet more interesting and useful theorems. To avoid introduction of costly errors in the form of incorrect theorems, proofs typically undergo an examination process, usually as a part of a peer-review. Traditionally, human mathematicians have been employed as proof verifiers; however, history is full of examples of undetected errors and important omissions even in the most widely examined proofs [3-7]. It has been estimated that at least a third of all mathematical publications contain errors [8]. To avoid errors and make the job of human verifiers as easy as

[1] Essay based on https://arxiv.org/abs/1609.00331

possible "a single step in a deduction has been required … [t]o be simple enough, broadly speaking, to be apprehended as correct by a human being in a single intellectual act. No doubt this custom originated in the desire that each single step of a deduction should be indubitable, even though the deduction as a whole may consist of a long chain of such steps" [9].

Despite such stringent requirements, it has long been realized that a single human verifier is not reliable enough to ascertain validity of a proof with a sufficient degree of reliability. In fact, it is known that humans are subject to hundreds of well-known "bugs"[2], and probably many more unknown ones. To reduce the number of potential mistakes and to increase our confidence in the validity of a proof, a number of independent human mathematicians should examine an important mathematical claim. As Calude puts it "A theorem is a statement which could be checked individually by a mathematician and confirmed also individually by at least two or three other mathematicians, each of them working independently. But already we can observe the weakness of the criterion: how many mathematicians are to check individually and independently the status of [a conjecture] to give it a status of a theorem?" [4].

Clearly, the greater the number of independent verifiers, the higher is our confidence in the validity of a theorem. We can say that "a theorem is validated if it has been accepted by a general agreement of the mathematical community" [4]. Krantz agrees and says: "it is the mathematics profession, taken as a whole, that decides what is correct and valid, and also what is useful and is interesting and has value" [10]. Wittgenstein expresses similar views, as quoted in [11]: "who validates the 'mathematical knowledge'? … the acceptability ultimately comes from the collective opinion of the social group of people practising mathematics." So, for many practitioners of mathematics, proof verification is a social and democratic process in which "[a]fter enough internalization, enough transformation, enough generalization, enough use, and enough connection, the mathematical community eventually decides that the central concepts in the original theorem, now perhaps greatly changed, have an ultimate stability. If the various proofs feel right and the results are examined from enough angles, then the truth of the theorem is eventually considered to be established" [12].

While the mathematical community as a whole constitutes a powerful proof verifier, a desire for ever greater accuracy has led researchers to develop mechanized verification systems capable of handling formal proofs of great length. The prototype for such verifiers has its roots in *formal systems* [13] proposed by David Hilbert and which "contain an algorithm that mechanically checks the validity of all proofs that can be formulated in the system. The formal system consists of an alphabet of symbols in which all statements can be written; a grammar that specifies how the symbols are to be combined; a set of axioms, or principles accepted without proof; and rules of inference for deriving theorems from the axioms" [14]. However, there is a tradeoff when one switches from using human verifiers to utilizing automated ones, namely: "People are usually not very good in checking formal correctness of proofs, but they are quite good at detecting potential weaknesses or flaws in proofs" [15]. " 'Artificial' mathematicians are far less ingenious and subtle than human mathematicians, but they surpass their human counterparts by being infinitely more patient and diligent" [4]. In other words, while automated verifiers are excellent at spotting incorrect deductions, they are much worse than humans at seeing the "big picture" outlined in the proof.

[2] https://en.wikipedia.org/wiki/List_of_cognitive_biases

Additionally, to maintain a consistent standard of verification for all accepted theorems, a significant effort would need to be applied to reexamine already-accepted proofs. "to do so would certainly entail going back and rewriting from scratch all old mathematical papers whose results we depend on. It is also quite hard to come up with good technical choices for formal definitions that will be valid in the variety of ways that mathematicians want to use them and that will anticipate future extensions of mathematics. … [M]uch of our time would be spent with international standards commissions to establish uniform definitions and resolve huge controversies" [15].

Such criticism of automated verifiers is not new and has been expressed in the past, particularly from a human centric point of view: "No matter how precise the rules (logical and physical) are, we need human consciousness to apply the rules and to understand them and their consequences. Mathematics is a human activity" [4]. Additionally, "[m]echanical proof-checkers have indeed been developed, though their use is currently limited by the need or the proof to be written in precisely the right logical formalism" [16].

Despite such criticism, there is also a lot of hope in terms of what automated verification can offer mathematics. "[M]athematical knowledge is far too vast to be understood by one person, moreover, it has been estimated that the total amount of published mathematics doubles every ten to fifteen years… Perhaps computers can also help us to navigate, abstract and, hence, understand … proofs. Realising this dream of: computer access to a world repository of mathematical knowledge; visualising and understanding this knowledge; reusing and combining it to discover new knowledge" [17].

3. Unverifiability

Unverifiability, an idea frequently discussed in philosophy [18-20], has been implicitly present in mathematics since the early days of the field. In this section, we survey literature that deals with the limits of proof verifiability caused by infinite regress of verifiers, and provides analysis of the concept of unverifiability. We believe that such explicit discussion will be useful to researchers interested in being able to cite this important idea, which so far has been relegated to the status of mathematical folklore [21] and only alluded to in the literature, despite being a more general result than incompleteness [22, 23].

Unverifiability is a fundamental limitation on verification of mathematical proofs, computer software, behavior of intelligent agents, and all formal systems. It is an ultimate limit to our ability to know certain information and is similar to other major "impossibilities" to acquiring knowledge in our universe such as: uncertainty [24], randomness [25, 26], incompleteness [22, 23], undecidability [27], undefinability [28], unprovability [29], incompressibility [14], noncomputability [30], and relativity [31]. Many paths can lead us to arrive at the concept of unverifiability, but in this essay we concentrate specifically on the infinite regress of verifiers.

For example, Calude et al. state: "what if the 'agent' human or computer checking a proof for correctness makes a mistake (agents are fallible)? Obviously, another agent has to check that the agent doing the checking did not make any mistakes. Some other agent will need to check that

agent and so on. Eventually one runs out of agents who could check the proof and, in principle, they could all have made a mistake!" [32]. Later, Calude and Muller emphasize: "one cannot prove the correctness of the formal prover itself" [33]. Similarly, MacKenzie observes: "Indeed, if one was to apply the formal, mechanical notion of proof entirely stringently, might not the software of the automated theorem prover itself have to be verified formally? … The formal, mechanized notion of proof thus prompted a modern day version of Juvenal's ancient question, *quis custodiet ipsos custodes*, who will guard the guards themselves?" [34]. Others have expressed similar sentiments [11].

Our trust in a formal proof is only as strong as our trust in the verifier used to check the proof; as the verifier itself needs to be verified, and so on *ad infinitum*, we are never given a 100% guarantee of correctness, only asymptotically increasing probability of correctness. Worse yet, at the end of the chain of verifiers there is typically a single human, whose internal mechanism is simply not verifiable with our current technology and possibly not verifiable in principle. Additionally, problems other than infinite regress of verifiers may significantly reduce our ability to verify proofs. Such obstacles include: splicing and skipping [35], hidden lemmas [36], exponential size proofs [37] (with recent publication of a 200 terabyte computer proof [38] being only a current record which is unlikely to stand for long), impenetrable proofs [39], hardware failures [40, 41], Rice's theorem [42], and Gödel's Incompleteness theorem [22].

After the advent of probabilistic proofs by Rabin [43], "[s]ome have argued that there is no essential difference between such probabilistic proofs and the deterministic proofs of standard mathematical practice. Both are convincing arguments. Both are to be believed with a certain probability of error. In fact, many deterministic proofs, it is claimed, have a higher probability of error" [44]. "… the authenticity of a mathematical proof is not absolute, but only probabilistic. … Proofs cannot be too long, else their probabilities go down and they baffle the checking process. To put it in another way: all really deep theorems are false (or at best unproved or unprovable). All true theorems are trivial" [3]. "A derivation of a theorem or a verification of a proof has only probabilistic validity. It makes no difference whether the instrument of derivation or verification is man or a machine. The probabilities may vary, but are roughly of the same order of magnitude" [3]. All proofs have a certain level of "proofness" [45], which can be made arbitrarily deep via expending necessary verification resources, but "in no domain of mathematics is the notion of provability a perfect substitute for the notion of truth [46]." To conclude, we reiterate Knuth's famous warning: "Beware of bugs in the above code: I have only proved it correct, not tried it."

4. Unverifiability of Software

Unverifiability has important consequences not just for mathematicians and philosophers of knowledge; more recently it has become an important issue in software and hardware verification, which can be seen as special cases of proof verification [47, 48]. Just like a large portion of published mathematical proofs, software is known to contain massive amounts of bugs [49], perhaps as many as fifty per thousand lines of code[3], but maybe as few as 2.3 [50]. Similarly, just like with mathematical proofs, the issue of infinite regress of verifiers is making software only probabilistically verifiable. For example, Fetzer writes: "There are no special difficulties so long

[3] http://www.theengineer.co.uk/issues/may-2013-online/verification-system-aims-to-guarantee-software-function/

as [higher-level machines'] intended interpretations are abstract machines. When their intended interpretations are target machines, then we encounter the problem of determining the reliability of the verifying programs themselves ("How do we verify the verifiers?"), which invites a regress of relative verifications" [51].

This notion of unverifiability of software has been a part of the field since its early days. Smith writes: "For fundamental reasons - reasons that anyone can understand - there are inherent limitations to what can be proven about computers and computer programs. … Just because a program is "proven correct" …, you cannot be sure that it will do what you intend" [52]. Rodd agrees and says: "Indeed, although it is now almost trite to say it, since the comprehensive testing of software is impossible, only very vague estimates of any program's reliability seem ever to be possible" [53]. Currently, most software is released without any attempt to formally verify it in the first place.

4.1 Unverifiability of Artificial Intelligence

One particular type of software, known as Artificial Intelligence (AI) (and even more so superintelligence), differs from other programs by its ability to learn new behaviors, adjust its performance, and act semi-autonomously in novel situations. Given the potential impact from intelligent software, it is not surprising that the ability to verify future intelligent behavior is one of the grand challenges of modern AI research [54-57].

It has been observed that science frequently discovers so called "conjugate (complementary) pairs", "a couple of requirements, each of them being satisfied only at the expense of the other … . It is known as the Principle of Complementarity in physics. Famous prototypes of conjugate pairs are (position, momentum) discovered by W. Heisenberg in quantum mechanics and (consistency, completeness) discovered by K. Gödel in logic. But similar warnings come from other directions. According to Einstein …, 'in so far as the propositions of mathematics are certain, they do not refer to reality, and in so far as they refer to reality, they are not certain', hence (certainty, reality) is a conjugate pair" [32]. Similarly, in proofs we are "[t]aking rigour as something that can be acquired only at the expense of meaning and conversely, taking meaning as something that can be obtained only at the expense of rigour" [32]. With respect to intelligent agents, we can propose an additional conjugate pair - (capability, control). The more generally intelligent and capable an entity is, the less likely it is to be predictable, controllable, or verifiable.

It is becoming obvious that just as we can only have probabilistic confidence in correctness of mathematical proofs and software implementations, our ability to verify intelligent agents is at best limited. As Klein puts it: "if you really want to build a system that can have truly unexpected behaviour, then by definition you cannot verify that it is safe, because you just don't know what it will do."[4] Muehlhauser writes: "The same reasoning applies to [Artificial General Intelligence] AGI 'friendliness.' Even if we discover (apparent) solutions to known open problems in Friendly AI research, this does not mean that we can ever build an AGI that is 'provably friendly' in the strongest sense, because … we can never be 100% certain that there are no errors in our formal reasoning. … Thus, the approaches sometimes called 'provable security,' 'provable safety,' and 'provable friendliness' should not be misunderstood as offering 100% guarantees of security,

[4] https://intelligence.org/2014/02/11/gerwin-klein-on-formal-methods

safety, and friendliness."[5] Jilk, writing on limits to verification and validation in AI, points out that "language of certainty" is unwarranted in reference to agentic behavior [58]. He also states: "there cannot be a general automated procedure for verifying that an agent absolutely conforms to any determinate set of rules of action."

Seshia et al., describing some of the challenges of creating Verified Artificial Intelligence, note: "It may be impossible even to precisely define the interface between the system and environment (i.e., to identify the variables/features of the environment that must be modeled), let alone to model all possible behaviors of the environment. Even if the interface is known, non-deterministic or over-approximate modeling is likely to produce too many spurious bug reports, rendering the verification process useless in practice. … [T]he complexity and heterogeneity of AI-based systems means that, in general, many decision problems underlying formal verification are likely to be undecidable. … To overcome this obstacle posed by computational complexity, one must … settle for incomplete or unsound formal verification methods" [57].

These results are not surprising. AI cannot be verified because AI itself can serve as a verifier which we already showed cannot be verified because of infinite regress problem and general unverifiability. By spending increasing computational resources, the best we can hope for is an increased statistical probability that our mathematical proofs, and software/AI are error free, but we should never forget that a 100% accurate verification is not possible, even in theory, and act accordingly. Artificial Intelligence, and even more so artificial Superintelligence, is unverifiable and so potentially unsafe [59-64]. In addition to Unverifiability, superintelligent systems will also be Unpredictable and Unexplainable, limiting our ability to control or understand them.

a) Unpredictability

"*Unpredictability* of AI, one of many impossibility results in AI Safety also known as Unknowability [172] or Cognitive Uncontainability [173], is defined as our inability to precisely and consistently predict what specific actions an intelligent system will take to achieve its objectives, even if we know terminal goals of the system. It is related but is not the same as unexplainability and incomprehensibility of AI. Unpredictability does not imply that better-than-random statistical analysis is impossible; it simply points out a general limitation on how well such efforts can perform, and is particularly pronounced with advanced generally intelligent systems (superintelligence) in novel domains. In fact we can present a proof of unpredictability for such, superintelligent, systems.

Proof. This is a proof by contradiction. Suppose not, suppose that unpredictability is wrong and it is possible for a person to accurately predict decisions of superintelligence. That means they can make the same decisions as the superintelligence, which makes them as smart as superintelligence but that is a contradiction as superintelligence is defined as a system smarter than any person is. That means that our initial assumption was false and unpredictability is not wrong." [174].

b) Unexplainability and Incomprehensibility

"Unexplainability as impossibility of providing an explanation for certain decisions made by an intelligent system which is both 100% accurate and comprehensible. … A complimentary concept to Unexplainability, *Incomprehensibility* of AI address capacity of people to completely

[5] https://intelligence.org/2013/10/03/proofs/

understand an explanation provided by an AI or superintelligence. We define Incomprehensibility as an impossibility of completely understanding any 100% - accurate explanation for certain decisions of intelligent system, by any human." [176].

"Incomprehensibility is supported by well-known impossibility results. Charlesworth proved his Comprehensibility theorem while attempting to formalize the answer to such questions as: "If [full human-level intelligence] software can exist, could humans understand it?" [177]. While describing implications of his theorem on AI, he writes [178]: "Comprehensibility Theorem is the first mathematical theorem implying the impossibility of any AI agent or natural agent—including a not-necessarily infallible human agent—satisfying a rigorous and deductive interpretation of the self-comprehensibility challenge. … Self-comprehensibility in some form might be essential for a kind of self-reflection useful for self-improvement that might enable some agents to increase their success." It is reasonable to conclude that a system which doesn't comprehend itself would not be able to explain itself.

Hernandez-Orallo et al. introduce the notion of K-incomprehensibility (a.k.a. K-hardness) [179]. "This will be the formal counterpart to our notion of hard-to-learn good explanations. In our sense, a k-*incomprehensible* string with a high *k* (difficult to comprehend) is different (harder) than a k-*compressible* string (difficult to learn) [180] and different from classical computational complexity (slow to compute). Calculating the value of *k* for a given string is not computable in general. Fortunately, the converse, i.e., given an arbitrary *k*, calculating whether a string is k-*comprehensible* is computable. … Kolmogorov Complexity measures the amount of information but not the complexity to understand them." [179].

Similarly, Yampolskiy writes: "Historically, the complexity of computational processes has been measured either in terms of required steps (time) or in terms of required memory (space). Some attempts have been made in correlating the compressed (Kolmogorov) length of the algorithm with its complexity [181], but such attempts didn't find much practical use. We suggest that there is a relationship between how complex a computational algorithm is and intelligence, in terms of how much intelligence is required to either design or comprehend a particular algorithm. Furthermore we believe that such an intelligence based complexity measure is independent from those used in the field of complexity theory. … Essentially the intelligence based complexity of an algorithm is related to the minimum intelligence level required to design an algorithm or to understand it. This is a very important property in the field of education where only a certain subset of students will understand the more advanced material. We can speculate that a student with an "IQ" below a certain level can be shown to be incapable of understanding a particular algorithm. Likewise we can show that in order to solve a particular problem (P VS. NP) someone with IQ of at least X will be required."

Yampolskiy also addresses limits of understanding other agents in his work on the space of possible minds [139]: "Each mind design corresponds to an integer and so is finite, but since the number of minds is infinite some have a much greater number of states compared to others. This property holds for all minds. Consequently, since a human mind has only a finite number of possible states, there are minds which can never be fully understood by a human mind as such mind designs have a much greater number of states, making their understanding impossible as can be demonstrated by the pigeonhole principle." Hibbard points out safety impact from

incomprehensibility of AI: "Given the incomprehensibility of their thoughts, we will not be able to sort out the effect of any conflicts they have between their own interests and ours."" [176].

5. Implications for Physics and Science in General

From ancient Greece to modern times the idea that a fundamental relationship exists between mathematics and physics has persisted, with multiple claims that our universe is "written" in the language of mathematics [65-68]. Most famously, in 1960, Wigner published his seminal paper wondering about reasons for "the unreasonable effectiveness of mathematics in natural sciences" [69]. Tegmark, in his Mathematical Universe Hypothesis (MUH), suggested that the answer is that "our external physical reality is a mathematical structure" [70-72]. Since publication of MUH a significant amount of evidence has been published linking fundamental theory of nature (Quantum Physics) with different mathematical structures, including many recent discoveries [73-75].

In this essay, we argue that all mathematical proofs are inherently probabilistic and by extension so is all of mathematics. Therefore, Unverifiability – one of the ultimate limits to computational techniques can be considered as another piece of evidence in favor of MUH. In Copenhagen interpretation of quantum mechanics wavefunction description is probabilistic. The Born rule [76], a fundamental component of Copenhagen interpretation, provides a link between mathematics and experimental observations. More specifically, the Born rule predicts the probability of observing a particle at a given location as proportionate to the square of the magnitude of the wavefunction at that coordinate. Observations in quantum physics, just like in mathematics, are probabilistic [77] and by extension so is all of physics. Using a different interpretation of quantum physics, for example Many-Worlds interpretation [78] which is more consistent with MUH, leads to similar conclusions regarding verifiability [79].

Interestingly, Wigner himself in the same paper hints at mathematical unverifiability: "Similarly, it is possible that the theories, which we consider to be "proved" by a number of numerical agreements which appears to be large enough for us, are false because they are in conflict with a possible more encompassing theory which is beyond our means of discovery" [69]. More generally, cosmology and unverifiability are intimately linked meaning "we are unable to obtain a model of the universe without some specifically cosmological assumptions which are completely unverifiable" [80]. In case of theories such as Many-Worlds, the multiverse by definition is unobservable by the same observer or multiple observers able to communicate and so is experimentally unverifiable.

We have constructed our argument for unverifiability based on an infinite regress of verifiers after reasoning that verifiers are a mathematical equivalent of observers in physics. Infinite regress of observers is a fundamental part of physics, as exemplified by Wigner's Friend [81] thought experiment, as well as a fundamental matter of reproducibility of scientific experiments [82], in which scientists are the observers. A particular experiment could be repeated many times always increasing our confidence in the results, but just like with proof verification, never giving us a 100% certainty in its results. With the recent reproducibility crisis [83, 84] in multiple fields of science [85-88] our results are very timely and should allow for a better understanding of limits to verifiability and resources required to get desired levels of confidence in the results.

References

1. Tegmark, M. and A. Aguirre, *Physics of the Observer - an international request for proposals for research and outreach projects*. 2015: http://www.braude.ac.il/files/research_development/2016-Request-for-Proposals_FQXi.pdf.
2. Buss, S.R., *An introduction to proof theory*. Handbook of proof theory, 1998. **137**: p. 1-78.
3. Davis, P.J., *Fidelity in mathematical discourse: Is one and one really two?* American Mathematical Monthly, 1972: p. 252-263.
4. Calude, C.S., E. Calude, and S. Marcus, *Passages of Proof*, in *University of Auckland. Massey University. Romanian Academy. CDMTCS-180*. February 2002: New Zealand. Romania.
5. Detlefsen, M. and M. Luker, *The four-color theorem and mathematical proof.* The Journal of Philosophy, 1980. **77**(12): p. 803-820.
6. Kornai, A., *Bounding the impact of AGI.* Journal of Experimental & Theoretical Artificial Intelligence, 2014. **26**(3): p. 417-438.
7. Hardy, G.H., *Mathematical proof.* Mind, 1929. **38**(149): p. 1-25.
8. Lamport, L., *How to write a proof.* The American mathematical monthly, 1995. **102**(7): p. 600-608.
9. Mackenzie, D., *The Automation of Proof: A Historical and Sociological Exploration.* IEEE Annals of the History of Computing, 1995. **17(3)**: p. 7-29.
10. Krantz, S.G., *The Proof is in the Pudding. A Look at the Changing Nature of Mathematical Proof*. 2007: Springer.
11. Calude, C.S. and S. Marcus, *Mathematical proofs at a crossroad?*, in *Theory Is Forever*. 2004, Springer. p. 15-28.
12. Millo, R.A.D., R.J. Lipton, and A.J. Perlis, *Social Processes and Proofs of Theorems and Programs.* Communications of the ACM, May 1979. **22(5)**.
13. Omodeo, E.G. and J.T. Schwartz, *A 'Theory'mechanism for a proof-verifier based on first-order set theory*, in *Computational Logic: Logic Programming and Beyond*. 2002, Springer. p. 214-230.
14. Chaitin, G.J., *Randomness and Mathematical Proof.* Scientific American, May 1975. **232(5)**: p. 47-52.
15. Thurston, W.P., *On Proof and Progress in Mathematics.* Bulletin of the American Mathematical Society, April 1994. **30(2)**: p. 161-177.
16. MacKenzie, D., *Slaying the Kraken: The Sociohistory of a Mathematical Proof.* Social Studies of Science, February 1999. **29(1)**: p. 7-60.
17. Kohlhase, M., *OMDoc: An Open Markup Format for Mathematical Documents.* Lecture Notes in Artificial Intelligence, 2006. **4180**.
18. Feuer, L.S., *The Paradox of Verifiability.* Philosophy and Phenomenological Research, 1951. **12**(1): p. 24-41.
19. Black, M., *The principle of verifiability.* Analysis, 1934. **2**(1-2): p. 1-6.
20. Hoy, R.C., *The Unverifiability of Unverifiability.* Philosophy and Phenomenological Research, 1973. **33**(3): p. 393-398.
21. Wilder, R.L., *The nature of mathematical proof.* The American Mathematical Monthly, 1944. **51**(6): p. 309-323.
22. Gödel, K., S.C. Kleene, and J.B. Rosser, *On undecidable propositions of formal mathematical systems*. 1934: Institute for Advanced Study Princeton, NJ.
23. Calude, C.S. and S. Rudeanu, *Proving as a computable procedure.* Fundamenta Informaticae, 2005. **64**(1-4): p. 43-52.
24. Heisenberg, W., *Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik.* Zeitschrift für Physik, 1927. **43**(3-4): p. 172-198.
25. Calude, C.S. and M.A. Stay, *From Heisenberg to Gödel via Chaitin.* International Journal of Theoretical Physics, 2007. **46**(8): p. 2013-2025.
26. Calude, C.S., *Incompleteness, complexity, randomness and beyond.* Minds and Machines, 2002. **12**(4): p. 503-517.
27. Turing, A.M., *On Computable Numbers, with an Application to the Entscheidungsproblem.* Proceedings of the London Mathematical Society, 1936. **42**: p. 230-265.
28. Murawski, R., *Undefinability of truth. The problem of priority: Tarski vs Gödel.* History and Philosophy of Logic, 1998. **19**(3): p. 153-160.
29. Boolos, G., *The unprovability of consistency: an essay in modal logic*. 2009: Cambridge University Press.
30. Calude, C.S., M.J. Dinneen, and C.-K. Shu, *Computing a glimpse of randomness.* Experimental Mathematics, 2002. **11**(3): p. 361-370.
31. Einstein, A., *Relativity: The special and the general theory*. 2015: Princeton University Press.
32. Calude, C.S., E. Calude, and S. Marcus, *Passages of proof.* arXiv preprint math/0305213, 2003.
33. Calude, C.S. and C. Müller, *Formal proof: reconciling correctness and understanding*, in *Intelligent Computer Mathematics*. 2009, Springer. p. 217-232.
34. MacKenzie, D., *Mechanizing proof: computing, risk, and trust*. 2004: MIT Press.
35. Van Bendegem, J.P. and B. Van Kerkhove, *Mathematical arguments in context.* Foundations of Science, 2009. **14**(1-2): p. 45-57.
36. Lakatos, I., *Proofs and refutations (III).* The British Journal for the Philosophy of Science, 1963. **14**(55): p. 221-245.
37. Haken, A., *The intractability of resolution.* Theoretical Computer Science, 1985. **39**: p. 297-308.
38. Heule, M.J., O. Kullmann, and V.W. Marek, *Solving and Verifying the boolean Pythagorean Triples problem via Cube-and-Conquer*. arXiv preprint arXiv:1605.00723, 2016.
39. Castelvecchi, D., *The biggest mystery in mathematics: Shinichi Mochizuki and the impenetrable proof.* Nature, 2015. **526**: p. 178-181.
40. Wolf, M., F. Grodzinsky, and K. Miller, *Is quantum computing inherently evil?* CEPE 2011: Crossing Boundaries: p. 302.
41. Schmidhuber, J., *Ultimate cognition à la Gödel*. Cognitive Computation, 2009. **1**(2): p. 177-193.
42. Rice, H.G., *Classes of recursively enumerable sets and their decision problems.* Transactions of the American Mathematical Society, 1953. **74**(2): p. 358-366.
43. Rabin, M., *Probabilistic algorithms*, in *Research Division, Thomas J. Watson IBM Research Center*. 1976.
44. Kleiner, I., *Rigor and proof in mathematics: A historical perspective*. Mathematics Magazine, 1991. **64**(5): p. 291-314.
45. Manin, Y.I., *How convincing is a proof.* Math. Intelligencer, 1979. **2**(1): p. 17-18.
46. Tarski, A., *Truth and Proof.* Scientific American, 1969. **220**: p. 63-77.
47. Sørensen, M.H. and P. Urzyczyn, *Lectures on the Curry-Howard isomorphism*. Vol. 149. 2006: Elsevier.
48. Moore, J.S., *A mechanized program verifier*, in *Verified Software: Theories, Tools, Experiments*. 2008, Springer. p. 268-276.
49. Holzmann, G.J. *Economics of software verification*. in *Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering*. 2001. ACM.
50. Wallace, D.R. and R.U. Fujii, *Software verification and validation: an overview.* IEEE Software, 1989. **6**(3): p. 10.
51. Fetzer, J.H., *Program verification: the very idea.* Communications of the ACM, 1988. **31**(9): p. 1048-1063.
52. Smith, B.C., *The limits of correctness.* ACM SIGCAS Computers and Society, 1985. **14**(1): p. 18-26.
53. Rodd, M., *Safe AI—is this possible?* Engineering Applications of Artificial Intelligence, 1995. **8**(3): p. 243-250.
54. Russell, S., D. Dewey, and M. Tegmark, *Research Priorities for Robust and Beneficial Artificial Intelligence.* AI Magazine, 2015. **36**(4).
55. Yampolskiy, R.V., *Artificial intelligence safety engineering: Why machine ethics is a wrong approach*, in *Philosophy and Theory of Artificial Intelligence*. 2013, Springer. p. 389-396.
56. Menzies, T. and C. Pecheur, *Verification and validation and artificial intelligence.* Advances in computers, 2005. **65**: p. 153-201.
57. Seshia, S.A. and D. Sadigh, *Towards Verified Artificial Intelligence.* arXiv preprint arXiv:1606.08514, 2016.
58. Jilk, D.J., *Limits to Verification and Validation of Agentic Behavior.* arXiv preprint arXiv:1604.06963, 2016.
59. Pistono, F. and R.V. Yampolskiy, *Unethical Research: How to Create a Malevolent Artificial Intelligence*. arXiv preprint arXiv:1605.02817, 2016.
60. Yampolskiy, R.V. *Taxonomy of Pathways to Dangerous Artificial Intelligence*. in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*. 2016.
61. Yampolskiy, R.V., *Utility function security in artificially intelligent agents.* Journal of Experimental & Theoretical Artificial Intelligence, 2014. **26**(3): p. 373-389.
62. Sotala, K. and R.V. Yampolskiy, *Responses to catastrophic AGI risk: a survey.* Physica Scripta, 2014. **90**(1): p. 018001.
63. Yampolskiy, R. and J. Fox, *Safety engineering for artificial general intelligence.* Topoi, 2013. **32**(2): p. 217-226.
64. Yampolskiy, R.V., *What to Do with the Singularity Paradox?*, in *Philosophy and Theory of Artificial Intelligence*. 2013, Springer. p. 397-413.
65. Tegmark, M., *It's all just mathematics.* Physics World, 2014. **27**(02): p. 22.
66. Dirac, P.A.M., *XI.—The Relation between Mathematics and Physics.* Proceedings of the Royal Society of Edinburgh, 1940. **59**: p. 122-129.
67. Wolfram, S., *A new kind of science*. Vol. 5. 2002: Wolfram media Champaign.
68. Tipler, F.J., *The structure of the world from pure numbers.* Reports on Progress in Physics, 2005. **68**(4): p. 897.
69. Wigner, E.P., *The unreasonable effectiveness of mathematics in the natural sciences. May 11, 1959.* Communications on pure and applied mathematics, 1960. **13**(1): p. 1-14.
70. Tegmark, M., *Is "the theory of everything" merely the ultimate ensemble theory?* Annals of Physics, 1998. **270**(1): p. 1-51.
71. Tegmark, M., *The mathematical universe.* Foundations of physics, 2008. **38**(2): p. 101-150.
72. Tegmark, M., *Our mathematical universe: My quest for the ultimate nature of reality*. 2014: Penguin UK.
73. Friedmann, T. and C. Hagen, *Quantum mechanical derivation of the Wallis formula for π.* Journal of Mathematical Physics, 2015. **56**(11): p. 112101.
74. Swiecicki, I., T. Gobron, and D. Ullmo, *Schrödinger approach to mean field games.* Physical review letters, 2016. **116**(12): p. 128701.
75. Bender, C.M., D.C. Brody, and M.P. Müller, *Hamiltonian for the zeros of the Riemann zeta function.* Physical Review Letters, 2017. **118**(13): p. 130201.
76. Born, M., *Quantenmechanik der stoßvorgänge.* Zeitschrift für Physik A Hadrons and Nuclei, 1926. **38**(11): p. 803-827.
77. Feynman, R.P. *The Concept of Probability in Quantum Mechanics*. in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. 1951. The Regents of the University of California.
78. Everett III, H., *" Relative state" formulation of quantum mechanics.* Reviews of modern physics, 1957. **29**(3): p. 454.
79. Albrecht, A. and D. Phillips, *Origin of probabilities and their application to the multiverse.* Physical Review D, 2014. **90**(12): p. 123514.
80. Ellis, G.F., *Cosmology and verifiability.* Quarterly Journal of the Royal Astronomical Society, 1975. **16**: p. 245-264.
81. Wigner, E.P., *Remarks on the Mind-Body Question*, in *Symmetries and reflections*. 1967, Indiana University Press. p. 171-184.
82. Vaux, D.L., F. Fidler, and G. Cumming, *Replicates and repeats—what is the difference and is it significant?* EMBO reports, 2012. **13**(4): p. 291-296.
83. Ioannidis, J.P., *Why most published research findings are false.* PLos med, 2005. **2**(8): p. e124.
84. Baker, M., *1,500 scientists lift the lid on reproducibility.* Nature, 2016. **533**(7604): p. 452-454.
85. Collaboration, O.S., *Estimating the reproducibility of psychological science.* Science, 2015. **349**(6251): p. aac4716.
86. Abbott, A., *Disputed results a fresh blow for social psychology: failure to replicate intelligence-priming effects ignites row in research community.* Nature, 2013. **497**(7447): p. 16-17.
87. Ioannidis, J.P., *Contradicted and initially stronger effects in highly cited clinical research*. Jama, 2005. **294**(2): p. 218-228.
88. Baker, M. and E. Dolgin, *Cancer reproducibility project releases first results.* Nature, 2017. **541**(7637): p. 269-270.
89. Sotala, K. and R.V. Yampolskiy, *Responses to Catastrophic AGI risk: A Survey.* Physica Scripta, January 2015. **90**.
90. Yampolskiy, R.V., *Artificial Superintelligence: a Futuristic Approach*. 2015: Chapman and Hall/CRC.
91. Yampolskiy, R.V., *The Space of Possible Mind Designs*, in *Artificial General Intelligence*. 2015, Springer. p. 218-227.
92. Russell, S., et al., *Research Priorities for Robust and Benecial Artificial Intelligence*, in *Future of Life Institute*. January 23, 2015: http://futureoflife.org/static/data/documents/research_priorities.pdf.
93. Yudkowsky, E. and M. Herreshoff. *Tiling agents for self-modifying AI, and the Löbian obstacle*. in *MIRI Technical Report*. 2013. Available at: http://intelligence.org/files/TilingAgentsDraft.pdf.
94. Yampolskiy, R.V., *On the Limits of Recursively Self-Improving AGI*, in *The Eighth Conference on Artificial General Intelligence*. July 22-25, 2015: Berlin, Germany.
95. Yampolskiy, R.V., *Analysis of Types of Self-Improving Software*, in *The Eighth Conference on Artificial General Intelligence*. July 22-25, 2015: Berlin, Germany.
96. Yampolskiy, R.V., *Turing Test as a Defining Feature of AI-Completeness*, in *Artificial Intelligence, Evolutionary Computation and Metaheuristics - In the footsteps of Alan Turing. Xin-She Yang (Ed.)*. 2013, Springer. p. 3-17.
97. Yampolskiy, R.V., *AI-Complete, AI-Hard, or AI-Easy – Classification of Problems in AI*, in *The 23rd Midwest Artificial Intelligence and Cognitive Science Conference*. April 21-22, 2012: Cincinnati, OH, USA.
98. Yampolskiy, R.V., *AI-Complete CAPTCHAs as Zero Knowledge Proofs of Access to an Artificially Intelligent System.* ISRN Artificial Intelligence, 2011. **271878**.
99. Yampolskiy, R.V., L. Ashby, and L. Hassan, *Wisdom of Artificial Crowds—A Metaheuristic Algorithm for Optimization.* Journal of Intelligent Learning Systems and Applications, 2012. **4**(2): p. 98-107.
100. Yampolskiy, R.V. and E.L.B. Ahmed, *Wisdom of artificial crowds algorithm for solving NP-hard problems.* International Journal of Bio-Inspired Computation (IJBIC). **3(6)**: p. 358-369.
101. Wiedijk, F., *The seventeen provers of the world: Foreword by Dana S. Scott*. Vol. 3600. 2006: Springer Science & Business Media.
102. Necula, G.C. and P. Lee, *Safe, untrusted agents using proof-carrying code*, in *Mobile Agents and Security*. 1998, Springer. p. 61-91.
103. Mülhölzer, F., *A mathematical proof must be surveyable.* What Wittgenstein meant by this and what it implies. Grazer Philosophische Studien, 2005. **71**: p. 57-86.
104. Coleman, E., *The surveyability of long proofs.* Foundations of Science, 2009. **14**(1-2): p. 27-43.
105. Tymoczko, T., *Computers, proofs and mathematicians: A philosophical investigation of the four-color proof.* Mathematics magazine, 1980. **53**(3): p. 131-138.
106. Norwood, F., *Long proofs.* The American Mathematical Monthly, 1982. **89**(2): p. 110-112.
107. Yampolskiy, R.V., *Efficiency Theory: a Unifying Theory for Information, Computation and Intelligence.* Journal of Discrete Mathematical Sciences & Cryptography, 2013. **16(4-5)**: p. 259-277.
108. Jakobsson, M., K. Sako, and R. Impagliazzo. *Designated verifier proofs and their applications*. in *Advances in Cryptology—EUROCRYPT'96*. 1996. Springer.
109. Peng, K., *Efficient proof of bid validity with untrusted verifier in homomorphic e-auction.* IET Information Security, 2013. **7**(1): p. 11-21.
110. Arlt, S., et al. *The Gradual Verifier*. in *NASA Formal Methods*. 2014. Springer.
111. Kondrat'ev, D.A. and A.V. Promskii, *Towards the verified verifier. Theory and practice.* Modelirovanie i Analiz Informatsionnykh Sistem [Modeling and Analysis of Information Systems], 2014. **21**(6): p. 71-82.
112. Appel, A.W., *Foundational proof-carrying code*, in *16th Annual IEEE Symposium on Logic in Computer Science*. 2001. p. 247-256.

APPENDIX: Classification of Verifiers

A certain connection exists between the concept of observer in physics and a verifier in mathematics/science. Both must be instantiated in the physical world as either hardware or software to perform its function, but other than that, we currently have a very limited understanding of types and properties associated with such agents. As the first step, we propose a simple classification system for verifiers, sorting them with respect to domain of application, type of implementation, and general properties. With respect to their domain, we see verifiers as necessary for checking mathematical proofs, scientific theories, software correctness, intelligent behavior safety, and consistency and properties of algorithms. Some examples:

- **Software Verifier** – evaluates correctness of a program. Via the Curry-Howard Correspondence [47], proof verification and program verification are equivalent and software verification is a special case of theorem verification restricted to computational logic [48]. A compiler or interpreter can be seen as a program syntax verifier among other things.
- **AI-Verifier** – is a particular type of Software Verifier capable of verifying the behavior of intelligent systems in novel environments unknown at the time of design [92, 93]. Yampolskiy presents verification of self-improving software [94, 95] as a particular challenge to the AI community: "Ideally every generation of self-improving system should be able to produce a verifiable proof of its safety for external examination" [55]. Consequently, research linking functional specification to physical states is of great interest. "This type of theory would allow use of formal tools to anticipate and control behaviors of systems that approximate rational agents, alternate designs such as satisficing agents, and systems that cannot be easily described in the standard agent formalism (powerful prediction systems, theorem-provers, limited-purpose science or engineering systems, etc.). It may also be that such a theory could allow rigorously demonstrating that systems are constrained from taking certain kinds of actions or performing certain kinds of reasoning" [92].
- **Scientific Theory Verifier** – examines the output of computer simulations of scientific theories. A scientific theory cannot be considered fully accepted until it can be expressed as an algorithm and simulated on a computer. It should produce observations consistent with measurements obtained in the real world, perhaps adjusting for the relativity of time scale between simulation and the real world. In other words, an unsimulatable hypothesis should be considered significantly weaker than a simulatable one. It is possible that the theory cannot be simulated due to limits in our current computational capacity, hardware design, or capability of programmers and that it will become simulatable in the future, but until such time, it should have a tentative status. A scientific theory verifier could be seen as a formalized equivalent of a peer-reviewer in science.
- **NP Solution Verifier –** is an algorithm which can quickly (in polynomial time) check a certificate (also called witness) representing a solution, which can then be used to determine if a computation produces a "yes" or "no" answer. In fact, one of the requirements of NP-Completeness states that a problem is in that class if there exists a verifier for the problem. An NP-Completeness Verifier would check a reduction of a novel problem to an already known problem in the NP class to determine if it is of equal or lesser complexity. Analogously, we can postulate an AI-Completeness Verifier capable of checking if a problem is reducible to an instance of the Turing Test [96-98].

With respect to type, verifiers could be people (or groups of people), software, hypothetical agents such as oracles, or artificial (super)intelligent entities. For example:

- **Human Mathematician** – historically the default verifier for most mathematical proofs. Individual mathematicians have been recruited to examine mathematical reasoning since the inception of the field. Recent developments in computer-generated proofs appear to be beyond the capacity of human verifiers due to the size of such proofs.
- **Mathematical Community** – a collective of mathematicians taken as a whole used to examine and evaluate claimed proofs, while at the same time removing any outlier opinions of individual mathematicians. It is well known that the wisdom of crowds can outperform individual experts [99, 100].
- **Mechanical Verifier** (Automated Proof Checker) – automated software and hardware verifiers such as computer programs have been developed to assist in verification of formal proofs [101]. "The proof checker verifies that each inference step in the proof is a valid instance of one of the axioms and inference rules specified as part of the safety policy" [102]. They are believed to be more accurate than human mathematicians and are capable of verifying much longer proofs, which may not be surveyable [103-106] or too complex (not comprehensible [107]) for human mathematicians.
- **Hybrid Verifier** – a combination of other types of verifiers, most typically a human mathematician assisted by a mechanical verifier.
- **Oracle Verifier** – a verifier with access to an Oracle Turing Machine. Particular types would include a Halting Verifier (a hypothetical verifier not subject to the halting problem), a Gödel Verifier (not subject to incompleteness limits), and an undecidable proof verifier. All such verification would be done in constant time.
- **(Super)Intelligent Verifier** – a verifier capable of checking all decidable proofs, particularly those constructed by superintelligent AI.

Some verifiers also have non-trivial mathematical properties, which include: ability to self-verify, probabilistic proof checking, relative correctness, designated nature, meta-verification capacity, honest or dishonest behavior, and axiomatic acceptance. For example:

- **Axiomatically Correct Verifier** – a type of authority based verifier, which decides the truth of a theorem without a need to disclose its process. This is a verifier whose correctness is accepted without justification, much like an axiom is accepted by the math community.
- **Designated Verifier** – for some proofs of knowledge it is important that only the verifier nominated by the confirmer can get any confirmation of the correctness of the proof [108].
- **Honest (Trusted) Verifier** – "does not try to extract any secret from the prover by deviating from the proof protocol. … **Untrusted-Verifier** does not need to assume that the verifier is honest" [109].
- **Probabilistic Verifier** – a verifier, which by examining an ever-greater number of parts of a proof, arrives at a probabilistic measure of the correctness of the theorem. Such verifiers are a part of Zero Knowledge based protocols.
- **Relative Verifier** – a verifier with respect to which a particular theorem has been shown to be correct, which doesn't guarantee that it would be confirmed by other verifiers.
- **Gradual Verifier** – a verifier which determines a percentage of statements that are already guaranteed to be safe [110], permitting a gradual verification process to take place.
- **Self-Verifier** – an agent which is capable of verifying its own accuracy [111]. A frequently suggested approach to avoid an infinite regress of verifiers, a self-verifying verifier could contain an error causing it to erroneously claim its own correctness [112] and is also subject to limitations imposed by Gödel's Incompleteness theorem [22] and other similar self-referential constraints [93].