7-28-2017

# What are the ultimate limits to computational techniques: Verifier theory and unverifiability

Roman V. Yampolskiy
*University of Louisville*, roman.yampolskiy@louisville.edu

**INVITED COMMENT**

# What are the ultimate limits to computational techniques: verifier theory and unverifiability

To cite this article: Roman V Yampolskiy 2017 *Phys. Scr.* **92** 093001

View the article online for updates and enhancements.

## Related content

- Responses to catastrophic AGI risk: a survey
  Kaj Sotala and Roman V Yampolskiy

- 'The concept of information in physics': an interdisciplinary topical lecture
  T Dittrich

- On the problem of the interpretation of quantum physics
  A A Grib

## Recent citations

- Designing AI for Explainability and Verifiability: A Value Sensitive Design Approach to Avoid Artificial Stupidity in Autonomous Vehicles
  Steven Umbrello and Roman V. Yampolskiy

- Unexplainability and Incomprehensibility of AI
  Roman V. Yampolskiy

- Unpredictability of AI: On the Impossibility of Accurately Predicting All Actions of a Smarter Agent
  Roman V. Yampolskiy

# Invited Comment

# What are the ultimate limits to computational techniques: verifier theory and unverifiability

## Roman V Yampolskiy⃝

Computer Engineering and Computer Science, University of Louisville, KY, United States of America

E-mail: roman.yampolskiy@louisville.edu

CrossMark

## Abstract

Despite significant developments in proof theory, surprisingly little attention has been devoted to the concept of proof verifiers. In particular, the mathematical community may be interested in studying different types of proof verifiers (people, programs, oracles, communities, superintelligences) as mathematical objects. Such an effort could reveal their properties, their powers and limitations (particularly in human mathematicians), minimum and maximum complexity, as well as self-verification and self-reference issues. We propose an initial classification system for verifiers and provide some rudimentary analysis of solved and open problems in this important domain. Our main contribution is a formal introduction of the notion of unverifiability, for which the paper could serve as a general citation in domains of theorem proving, as well as software and AI verification.

Keywords: verifier theory, proof theory, observer, verified verifier, verifiability

## 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 (AI), 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, super-intelligences, 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.

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. In this work we present an overview of different types of verifiers currently relied on by the mathematical community, as well as a few novel types of verifiers which we suggest be added to the repertoire of mathematicians at least as theoretical tools of *verifier theory*. Our general analysis should be equally applicable to different types of proofs (induction, contradiction, exhaustion, enumeration, refinement, nonconstructive, probabilistic, holographic, experiment, picture, etc) and to computer software.

## 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

1

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 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'[1], 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.

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. Realizing this dream of: computer access to a world repository of mathematical knowledge; visualizing and understanding this knowledge; reusing and combining it to discover new knowledge' [17].

## 3. 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

---

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

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–Harvard correspondence [18], proof verification and program verification are equivalent and software verification is a special case of theorem verification restricted to computational logic [19]. 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 [20, 21]. Yampolskiy presents verification of self-improving software [22, 23] 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' [24]. 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' [20].
- *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 [25–27].

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 [28, 29].
- *Mechanical verifier* (automated proof checker)—automated software and hardware verifiers such as computer programs have been developed to assist in verification of formal proofs [30]. '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' [31]. They are believed to be more accurate than human mathematicians and are capable of verifying much longer proofs, which may not be surveyable [32–35] or too complex (not comprehensible [36]) 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 [37].

- *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' [38].
- *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 does not 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 [39], permitting a gradual verification process to take place.
- *Meta-verifier*—a hypothetical verifier capable of checking correctness of other verifiers.
- *Self-verifier*—an agent which is capable of verifying its own accuracy [40]. 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 [41] and is also subject to limitations imposed by Gödel's Incompleteness theorem [42] and other similar self-referential constraints [21].

## 4. Unverifiability

Unverifiability, an idea frequently discussed in philosophy [43–45], 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 [46] and only alluded to in the literature, despite being a more general result than incompleteness [42, 47].

*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 [48], randomness [49, 50], incompleteness [42, 47], undecidability [51], undefinability [52], unprovability [53], incompressibility [14], noncomputability [54], and relativity [55]. Many paths can lead us to arrive at the concept of unverifiability, but in this paper 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!' [56]. Later, Calude and Muller emphasize: 'one cannot prove the correctness of the formal prover itself' [57]. 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?' [58]. 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 [59], hidden lemmas [60], exponential size proofs [61] (with recent publication of a 200 terabyte computer proof [62] being only a current record which is unlikely to stand for long), impenetrable proofs [63], hardware failures [64, 65], Rice's theorem [66], and Gödel's Incompleteness theorem [42].

After the advent of probabilistic proofs by Rabin [67], '[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' [68]. '… 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' [69], 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 [70].' To conclude, we reiterate Knuth's famous warning: 'Beware of bugs in the above code: I have only proved it correct, not tried it.'

# 5. 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 [18, 19]. Just like a large portion of published mathematical proofs, software is known to contain massive amounts of bugs [71], perhaps as many as fifty per thousand lines of code[2], but maybe as few as 2.3 [72]. 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 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' [73].

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' [74]. 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' [75]. Currently, most software is released without any attempt to formally verify it in the first place.

## 5.1. Unverifiability of AI

One particular type of software, known as 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 [24, 76–78].

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 Heisenberg in quantum mechanics and (consistency, completeness) discovered by 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' [56]. 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' [56]. 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 behavior, then by definition you cannot verify that it is safe, because you just don't know what it will do'[3]. 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, safety, and friendliness.'[4] Jilk, writing on limits to verification and validation in AI, points out that 'language of certainty' is unwarranted in reference to agentic behavior [79]. 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 AI, 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' [78].

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. AI, and even more so artificial superintelligence, is unverifiable and so potentially unsafe [80–85].

---

## 6. 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 [86–89]. Most famously, in 1960, Wigner published his seminal paper wondering about reasons for 'the unreasonable effectiveness of mathematics in natural sciences' [90]. Tegmark, in his mathematical universe hypothesis (MUH), suggested that the answer is that 'our external physical reality is a mathematical structure' [91–93]. Since the publication of the MUH a significant amount of evidence has been published linking fundamental theory of nature (Quantum Physics) with different mathematical structures, including many recent discoveries [94–96].

In this paper, 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 the MUH. In Copenhagen interpretation of quantum mechanics wavefunction description is probabilistic. The Born rule [97], 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 [98] and by extension so is all of physics. Using a different interpretation of quantum physics, for example many-worlds interpretation [99] which is more consistent with the MUH, leads to similar conclusions regarding verifiability [100].

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' [90]. 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' [101]. 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 [102] thought experiment, as well as a fundamental matter of reproducibility of scientific experiments [103], 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 [104, 105] in multiple fields of science [106–109] 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.

## 7. Conclusions and future work

Our preliminary work suggests that the 'verifier' be investigated as a new mathematical object of interest for future study and opens the door for an improved understanding of the topic. For example, an artificially intelligent verifier could be used to re-check all previously published mathematical proofs, greatly increasing correctness of all proofs. Problems such as infinite regress of verifiers may be unsolvable, but methods such as probabilistic verification should be capable of giving us as much assurance as we are willing to pay for. Any progress in the proposed 'verifier theory' will have additional benefits beyond its contribution to mathematics by making it possible to design safer advanced AI, a topic that is predicted to become one of the greatest problems in science in the upcoming decades [110, 111]. A verifier is a hidden component of any proof; we can improve our capacity to verify by explicitly describing the required verification agent.

It would be valuable to learn what types of physical or informational systems can act as verifiers and what their essential properties are. We should explore how selection of the type of the verifier influences mathematics as a field and specifically what categories of theorems we can prove and which we cannot prove with respect to different verifiers. Are there still undiscovered types of mathematical verifiers? Does a group of verifiers have greater power than the sum of its component modules? How can verifiers perform best while operating with limited computational resources? What is the formal relationship between the set of all verifiers and the set of all observers? Can a verifier be hacked and can the attack be contained in the proof it is examining? Can all these questions be reduced to a broader question on the nature of different possible types of intelligences [112]?

## ORCID

Roman V Yampolskiy ⓘ https://orcid.org/0000-0001-9637-1161

## References

[1] Tegmark M and Aguirre A 2015 Physics of the observer—an international request for proposals for research and outreach projects (http://braude.ac.il/files/research_development/2016-Request-for-Proposals_FQXi.pdf)

[2] Buss S R 1998 An introduction to proof theory *Handbook of Proof Theory* vol 137 (Amsterdam: Elsevier) pp 1–78

[3] Davis P J 1972 Fidelity in mathematical discourse: is one and one really two? *Am. Math. Mon.* **79** 252–63

[4] Calude C S, Calude E and Marcus S 2002 *Passages of Proof* CDMTCS-180 University of Auckland, Massey University, Romanian Academy, New Zealand Romania

[5] Detlefsen M and Luker M 1980 The four-color theorem and mathematical proof *J. Phil.* **77** 803–20

[6] Kornai A 2014 Bounding the impact of AGI *J. Exp. Theor. Artif. Intell.* **26** 417–38

[7] Hardy G H 1929 Mathematical proof *Mind* **38** 1–25

[8] Lamport L 1995 How to write a proof *Am. Math. Mon.* **102** 600–8

[9] Mackenzie D 1995 The automation of proof: a historical and sociological exploration *IEEE Ann. Hist. Comput.* **17** 7–29

[10] Krantz S G 2007 *The Proof is in the Pudding, A Look at the Changing Nature of Mathematical Proof* (Berlin: Springer)

[11] Calude C S and Marcus S 2004 Mathematical proofs at a crossroad? *Theory is Forever* (Berlin: Springer) pp 15–28

[12] Millo R A D, Lipton R J and Perlis A J 1979 Social processes and proofs of theorems and programs *Commun. ACM* **22** 271–80

[13] Omodeo E G and Schwartz J T 2002 A 'theory' mechanism for a proof-verifier based on first-order set theory *Computational Logic: Logic Programming and Beyond* (Berlin: Springer) pp 214–30

[14] Chaitin G J 1975 Randomness and mathematical proof *Sci. Am.* **232** 47–52

[15] Thurston W P 1994 On proof and progress in mathematics *Bull. Am. Math. Soc* **30** 161–77

[16] MacKenzie D 1999 Slaying the kraken: the sociohistory of a mathematical proof *Soc. Stud. Sci.* **29** 7–60

[17] Kohlhase M 2006 OMDoc: an open markup format for mathematical documents *Lecture Notes in Artificial Intelligence* vol 4180 (New York: Springer)

[18] Sørensen M H and Urzyczyn P 2006 *Lectures on the Curry-Howard Isomorphism* vol 149 (Amsterdam: Elsevier)

[19] Moore J S 2008 A mechanized program verifier *Verified Software: Theories, Tools, Experiments* (Berlin: Springer) pp 268–76

[20] Russell S *et al* 2015 Research priorities for robust and benecial artificial intelligence Future of Life Institute (http://futureoflife.org/static/data/documents/research_priorities.pdf)

[21] Yudkowsky E and Herreshoff M 2013 Tiling agents for self-modifying AI, and the Löbian obstacle *MIRI Technical Report* (http://intelligence.org/files/TilingAgentsDraft.pdf)

[22] Yampolskiy R V 2015 On the limits of recursively self-improving AGI *The 8th Conf. on Artificial General Intelligence (Berlin, Germany, 22–25 July)* (https://doi.org/10.1007/978-3-319-21365-1_40)

[23] Yampolskiy R V 2015 Analysis of types of self-improving software *The 8th Conf. on Artificial General Intelligence (Berlin, Germany, 22–25 July)* (https://doi.org/10.1007/978-3-319-21365-1_39)

[24] Yampolskiy R V 2013 Artificial intelligence safety engineering: Why machine ethics is a wrong approach *Philosophy and Theory of Artificial Intelligence* (Berlin: Springer) pp 389–96

[25] Yampolskiy R V 2013 Turing test as a defining feature of AI-completeness *Artificial Intelligence, Evolutionary Computation and Metaheuristics—In the Footsteps of Alan Turing* ed X-S Xang (Berlin: Springer) pp 3–17

[26] Yampolskiy R V 2012 AI-complete, AI-hard, or AI-easy—classification of problems in AI *The 23rd Midwest Artificial Intelligence and Cognitive Science Conf. (Cincinnati, OH, USA, 21–22 April)*

[27] Yampolskiy R V 2011 AI-complete CAPTCHAs as zero knowledge proofs of access to an artificially intelligent system *ISRN Artif. Intell.* **271878** 1–6

[28] Yampolskiy R V, Ashby L and Hassan L 2012 Wisdom of artificial crowds—a metaheuristic algorithm for optimization *J. Intell. Learn. Syst. Appl.* **4** 98–107

[29] Yampolskiy R V and Ahmed E L B Wisdom of artificial crowds algorithm for solving NP-hard problems *Int. J. Bio-Inspired Comput. (IJBIC)* **3** 358–69

[30] Wiedijk F 2006 *The Seventeen Provers of the World: Foreword by Dana S Scott* vol 3600 (Berlin: Springer)

[31] Necula G C and Lee P 1998 Safe, untrusted agents using proof-carrying code *Mobile Agents and Security* (Berlin: Springer) pp 61–91

[32] Mülhölzer F 2005 A mathematical proof must be surveyable. What Wittgenstein meant by this and what it implies *Grazer Phil. Stud.* **71** 57–86

[33] Coleman E 2009 The surveyability of long proofs *Found. Sci.* **14** 27–43

[34] Tymoczko T 1980 Computers, proofs and mathematicians: a philosophical investigation of the four-color proof *Math. Mag.* **53** 131–8

[35] Norwood F 1982 Long proofs *Am. Math. Mon.* **89** 110–2

[36] Yampolskiy R V 2013 Efficiency theory: a unifying theory for information, computation and intelligence *J. Discrete Math. Sci. Cryptogr.* **16** 259–77

[37] Jakobsson M, Sako K and Impagliazzo R 1996 Designated verifier proofs and their applications *Advances in Cryptology—EUROCRYPT'96* (Berlin: Springer)

[38] Peng K 2013 Efficient proof of bid validity with untrusted verifier in homomorphic e-auction *IET Inform. Secur.* **7** 11–21

[39] Arlt S *et al* 2014 The gradual verifier *NASA Formal Methods* (Berlin: Springer)

[40] Kondrat'ev D A and Promskii A V 2014 Towards the verified verifier, theory and practice *Model. Analiz Informatsionnykh Sistem (Model. Anal. Inform. Syst.)* **21** 71–82

[41] Appel A W 2001 Foundational proof-carrying code *16th Annual IEEE Symp. on Logic in Computer Science* pp 247–56

[42] Gödel K, Kleene S C and Rosser J B 1934 *On Undecidable Propositions of Formal Mathematical Systems* (Princeton, NJ: Institute for Advanced Study)

[43] Feuer L S 1951 The paradox of verifiability *Phil. Phenomenological Res.* **12** 24–41

[44] Black M 1934 The principle of verifiability *Analysis* **2** 1–6

[45] Hoy R C 1973 The unverifiability of unverifiability *Phil. Phenomenological Res.* **33** 393–8

[46] Wilder R L 1944 The nature of mathematical proof *Am. Math. Mon.* **51** 309–23

[47] Calude C S and Rudeanu S 2005 Proving as a computable procedure *Fundam. Inform.* **64** 43–52

[48] Heisenberg W 1927 Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik *Z. Phys.* **43** 172–98

[49] Calude C S and Stay M A 2007 From Heisenberg to Gödel via Chaitin *Int. J. Theor. Phys.* **46** 2013–25

[50] Calude C S 2002 Incompleteness, complexity, randomness and beyond *Minds Mach.* **12** 503–17

[51] Turing A M 1936 On computable numbers, with an application to the entscheidungsproblem *Proc. London Math. Soc.* **42** 230–65

[52] Murawski R 1998 Undefinability of truth, the problem of priority: Tarski vs Gödel *Hist. Phil. Logic* **19** 153–60

[53] Boolos G 2009 *The Unprovability of Consistency: an Essay in Modal Logic* (Cambridge: Cambridge University Press)

[54] Calude C S, Dinneen M J and Shu C-K 2002 Computing a glimpse of randomness *Exp. Math.* **11** 361–70

[55] Einstein A 2015 *Relativity: The Special and the General Theory* (Princeton, NJ: Princeton University Press)

[56] Calude C S, Calude E and Marcus S 2001 Passages of Proof Workshop: Truths and Proofs *Annual Conference of the Australasian Association of Philosophy (New Zealand Division) (Auckland, December 2001)*

[57] Calude C S and Müller C 2009 Formal proof: reconciling correctness and understanding *Intelligent Computer Mathematics* (Berlin: Springer) pp 217–32

[58] MacKenzie D 2004 *Mechanizing Proof: Computing, Risk, and Trust* (Cambridge, MA: MIT Press)

[59] Van Bendegem J P and Van Kerkhove B 2009 Mathematical arguments in context *Found. Sci.* **14** 45–57

[60] Lakatos I 1963 Proofs and refutations (III) *Br. J. Phil. Sci.* **14** 221–45

[61] Haken A 1985 The intractability of resolution *Theor. Comput. Sci.* **39** 297–308

[62] Heule M J, Marijn J H, Kullmann O and Marek V W 2016 Solving and verifying the boolean pythagorean triples problem via cube-and-conquer *International Conference on Theory and Applications of Satisfiability Testing* (Berlin: Springer)

[63] Castelvecchi D 2015 The biggest mystery in mathematics: Shinichi Mochizuki and the impenetrable proof *Nature* **526** 178–81

[64] Wolf M, Grodzinsky F and Miller K 2011 Is quantum computing inherently evil? *CEPE* (Crossing Boundaries) p 302

[65] Schmidhuber J 2009 Ultimate cognition à la Gödel *Cog. Comput.* **1** 177–93

[66] Rice H G 1953 Classes of recursively enumerable sets and their decision problems *Trans. Am. Math. Soc.* **74** 358–66

[67] Rabin M 1976 Probabilistic algorithms *Research Division* (Thomas J Watson IBM Research Center)

[68] Kleiner I 1991 Rigor and proof in mathematics: a historical perspective *Math. Mag.* **64** 291–314

[69] Manin Y I 1979 How convincing is a proof? *Math. Intell.* **2** 17–8

[70] Tarski A 1969 Truth and proof *Sci. Am.* **220** 63–77

[71] Holzmann G J 2001 Economics of software verification *Proc. 2001 ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering* (New York: ACM)

[72] Wallace D R and Fujii R U 1989 Software verification and validation: an overview *IEEE Softw.* **6** 10

[73] Fetzer J H 1988 Program verification: the very idea *Commun. ACM* **31** 1048–63

[74] Smith B C 1985 The limits of correctness *ACM SIGCAS Comput. Soc.* **14** 18–26

[75] Rodd M 1995 Safe AI—is this possible? *Eng. Appl. Artif. Intell.* **8** 243–50

[76] Russell S, Dewey D and Tegmark M 2015 Research priorities for robust and beneficial artificial intelligence *AI Mag.* **36** 3–4

[77] Menzies T and Pecheur C 2005 Verification and validation and artificial intelligence *Adv. Comput.* **65** 153–201

[78] Seshia S A and Sadigh D 2016 Towards verified artificial intelligence arXiv:1606.08514

[79] Jilk D J 2016 Limits to verification and validation of agentic behavior arXiv:1604.06963

[80] Pistono F and Yampolskiy R V 2016 Unethical research: how to create a malevolent artificial intelligence arXiv:1605.02817

[81] Yampolskiy R V 2016 Taxonomy of pathways to dangerous artificial intelligence *Workshops at the 30th AAAI Conf. on Artificial Intelligence*

[82] Yampolskiy R V 2014 Utility function security in artificially intelligent agents *J. Exp. Theor. Artif. Intell.* **26** 373–89

[83] Sotala K and Yampolskiy R V 2014 Responses to catastrophic AGI risk: a survey *Phys. Scr.* **90** 018001

[84] Yampolskiy R and Fox J 2013 Safety engineering for artificial general intelligence *Topoi* **32** 217–26

[85] Yampolskiy R V 2013 What to do with the singularity paradox? *Philosophy and Theory of Artificial Intelligence* (Berlin: Springer) pp 397–413

[86] Tegmark M 2014 It's all just mathematics *Phys. World* **27** 22

[87] Dirac P A M 1940 XI. The relation between mathematics and physics *Proc. R. Soc. Edinburgh.* **59** 122–9

[88] Wolfram S 2002 *A New Kind of Science* vol 5 (Champaign: Wolfram Media)

[89] Tipler F J 2005 The structure of the world from pure numbers *Rep. Prog. Phys.* **68** 897

[90] Wigner E P 1960 The unreasonable effectiveness of mathematics in the natural sciences, Richard courant lecture in mathematical sciences delivered at New York University, May 11, 1959 *Commun. Pure Appl. Math.* **13** 1–14

[91] Tegmark M 1998 Is 'the theory of everything' merely the ultimate ensemble theory? *Ann. Phys., NY* **270** 1–51

[92] Tegmark M 2008 The mathematical universe *Found. Phys.* **38** 101–50

[93] Tegmark M 2014 *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality* (New York: Vintage)

[94] Friedmann T and Hagen C 2015 Quantum mechanical derivation of the Wallis formula for $\pi$ *J. Math. Phys.* **56** 112101

[95] Swiecicki I, Gobron T and Ullmo D 2016 Schrödinger approach to mean field games *Phys. Rev. Lett.* **116** 128701

[96] Bender C M, Brody D C and Müller M P 2017 Hamiltonian for the zeros of the Riemann zeta function *Phys. Rev. Lett.* **118** 130201

[97] Born M 1926 Quantenmechanik der stoßvorgänge *Z. Phys.* A **38** 803–27

[98] Feynman R P 1951 The concept of probability in quantum mechanics *Proc. 2nd Berkeley Symp. on Mathematical Statistics and Probability* (The Regents of the University of California)

[99] Everett H III 1957 'Relative state' formulation of quantum mechanics *Rev. Mod. Phys.* **29** 454

[100] Albrecht A and Phillips D 2014 Origin of probabilities and their application to the multiverse *Phys. Rev.* D **90** 123514

[101] Ellis G F 1975 Cosmology and verifiability *Q. J. R. Astron. Soc.* **16** 245–64

[102] Wigner E P 1967 Remarks on the mind-body question *Symmetries and Reflections* (Bloomington, IN: Indiana University Press) pp 171–84

[103] Vaux D L, Fidler F and Cumming G 2012 Replicates and repeats—what is the difference and is it significant? *EMBO Rep.* **13** 291–6

[104] Ioannidis J P 2005 Why most published research findings are false *PLoS Med.* **2** e124

[105] Baker M 2016 1,500 scientists lift the lid on reproducibility *Nature* **533** 452–4

[106] Collaboration O S 2015 Estimating the reproducibility of psychological science *Science* **349** aac4716

[107] Abbott A 2013 Disputed results a fresh blow for social psychology: failure to replicate intelligence-priming effects ignites row in research community *Nature* **497** 16–7

[108] Ioannidis J P 2005 Contradicted and initially stronger effects in highly cited clinical research *J. Am. Med. Assoc.* **294** 218–28

[109] Baker M and Dolgin E 2017 Cancer reproducibility project releases first results *Nature* **541** 269–70

[110] Sotala K and Yampolskiy R V 2015 Responses to catastrophic AGI risk: a survey *Phys. Scr.* **90** 1–33

[111] Yampolskiy R V 2015 *Artificial Superintelligence: a Futuristic Approach* (London, Boca Raton, FL: Chapman and Hall, CRC)

[112] Yampolskiy R V 2015 The space of possible mind designs *Artificial General Intelligence* (Berlin: Springer) pp 218–27