On the Limits of Recursively Self-Improving AGI

Roman V. Yampolskiy(✉)

Computer Engineering and Computer Science, Speed School of Engineering,
University of Louisville, Louisville, USA
roman.yampolskiy@louisville.edu

Abstract. Self-improving software has been a goal of computer scientists since the founding of the field of Artificial Intelligence. In this work we analyze limits on computation which might restrict recursive self-improvement. We also introduce Convergence Theory which aims to predict general behavior of RSI systems.

Keywords: Recursive self-improvement · Convergence theory · Bootstrapping

1 Introduction

Intuitively most of us have some understanding of what it means for a software system to be self-improving, however we believe it is important to precisely define such notions and to systematically investigate different types of self-improving software[1]. First we need to define the notion of improvement. We can talk about improved efficiency – solving same problems faster or with less need for computational resources (such as memory). We can also measure improvement in error rates or finding closer approximations to optimal solutions, as long as our algorithm is functionally equivalent from generation to generation. Efficiency improvements can be classified as either producing a trivial improvement as between different algorithms in the same complexity class (ex. NP), or as producing a fundamental improvement as between different complexity classes (ex. P vs NP) [1]. It is also very important to remember that complexity class notation (Big-O) may hide significant constant factors which while ignorable theoretically may change relative order of efficiency in practical applications of algorithms.

This type of analysis works well for algorithms designed to accomplish a particular task, but doesn't work well for general purpose intelligent software as an improvement in one area may go together with decreased performance in another domain. This makes it hard to claim that the updated version of the software is indeed an improvement. Mainly, the major improvement we want from self-improving intelligent software is higher degree of intelligence which can be approximated via machine friendly IQ tests [2] with a significant G-factor correlation.

[1] This paper is based on material excerpted, with permission, from the book - Artificial Superintelligence: a Futuristic Approach © 2015 CRC Press.

J. Bieger (Ed.): AGI 2015, LNAI 9205, pp. 394–403, 2015.
DOI: 10.1007/978-3-319-21365-1_40

A particular type of self-improvement known as Recursive Self-Improvement (RSI) is fundamentally different as it requires that the system not only get better with time, but that it gets better at getting better. A truly RSI system is theorized not to be subject to diminishing returns, but would instead continue making significant improvements and such improvements would become more substantial with time. Consequently, an RSI system would be capable of open ended self-improvement. As a result, it is possible that unlike with standard self-improvement, in RSI systems from generation-to-generation most source code comprising the system will be replaced by different code. This brings up the question of what "self" refers to in this context. If it is not the source code comprising the agent then what is it? Perhaps we can redefine RSI as Recursive Source-code Improvement (RSI) to avoid dealing with this philosophical problem. Instead of trying to improve itself such a system is trying to create a different system which is better at achieving same goals as the original system. In the most general case it is trying to create an even smarter artificial intelligence.

2 On the Limits of Recursively Self-Improving AGI

The mere possibility of recursively self-improving software remains unproven. In this section we present a number of arguments against such phenomenon. First of all, any implemented software system relies on hardware for memory, communication and information processing needs even if we assume that it will take a non-Von Neumann (quantum) architecture to run such software. This creates strict theoretical limits to computation, which despite hardware advances predicted by Moore's law will not be overcome by any future hardware paradigm. Bremermann [3], Bekenstein [4], Lloyd [5], Anders [6], Aaronson [7], Shannon [8], Krauss [9], and many others have investigated ultimate limits to computation in terms of speed, communication and energy consumption with respect to such factors as speed of light, quantum noise, and gravitational constant. Some research has also been done on establishing ultimate limits for enhancing human brain's intelligence [10]. While their specific numerical findings are outside of the scope of this work, one thing is indisputable: there are ultimate physical limits to computation. Since more complex systems have greater number of components and require more matter, even if individual parts are designed at nanoscale, we can conclude that just like matter and energy are directly related [11] and matter and information ("it from bit") [12] so is matter and intelligence. While we are obviously far away from hitting any limits imposed by availability of matter in the universe for construction of our supercomputers it is a definite theoretical upper limit on achievable intelligence.

In addition to limitations endemic to hardware, software-related limitations may present even bigger obstacles for RSI systems. Intelligence is not measured as a standalone value but with respect to the problems it allows to solve. For many problems such as playing checkers [13] it is possible to completely solve the problem (provide an optimal solution after considering all possible options) after which no additional performance improvement would be possible [14]. Other problems are known to be unsolvable regardless of level of intelligence applied to them [15]. Assuming separation of complexity classes (such as P vs NP) holds [1], it becomes obvious that certain

classes of problems will always remain only approximately solvable and any improvements in solutions will come from additional hardware resources not higher intelligence.

Wiedermann argues that cognitive systems form an infinite hierarchy and from a computational point of view human-level intelligence is upper-bounded by the $\sum_2$ class of the Arithmetic Hierarchy [16]. Because many real world problems are computationally infeasible for any non-trivial inputs even an AI which achieves human level performance is unlikely to progress towards higher levels of the cognitive hierarchy. So while theoretically machines with super-Turing computational power are possible, in practice they are not implementable as the non-computable information needed for their function is just that – not computable. Consequently Wiedermann states that while machines of the future will be able to solve problems, solvable by humans, much faster and more reliably they will still be limited by computational limits found in upper levels of the Arithmetic Hierarchy [16, 17].

Mahoney attempts to formalize what it means for a program to have a goal G and to self-improve with respect to being able to reach said goal under constraint of time, t [18]. Mahoney defines a goal as a function $G: N \rightarrow R$ mapping natural numbers N to real numbers R. Given a universal Turing machine L, Mahoney defines $P(t)$ to mean the positive natural number encoded by output of the program P with input t running on L after t time steps, or 0 if P has not halted after t steps. Mahoney's representation says that P has goal G at time t if and only if there exists $t' > t$ such that $G(P(t')) > G(P(t))$ and for all $t' > t$, $G(P(t') \geq G(P(t))$. If P has a goal G, then $G(P(t))$ is a monotonically increasing function of t with no maximum for $t > C$. Q improves on P with respect to goal G if and only if all of the following condition are true: P and Q have goal Q. $\exists t$, $G(Q(t)) > G(P(t))$ and $\sim\exists t$, $t' > t$, $G(Q(t)) > G(P(t))$ [18]. Mahoney then defines an improving sequence with respect to G as an infinite sequence of program $P_1, P_2, P_3, \ldots$ such that for $\forall i$, $i > 0$, P_{i+1} improves P_i with respect to G. Without the loss of generality Mahoney extends the definition to include the value -1 to be an acceptable input, so $P(-1)$ outputs appropriately encoded software. He finally defines P_1 as an RSI program with respect to G *iff* $P_i(-1) = P_{i+1}$ for all $i > 0$ and the sequence P_i, $i = 1, 2, 3 \ldots$ is an improving sequence with respect to goal G [18]. Mahoney also analyzes complexity of RSI software and presents a proof demonstrating that the algorithmic complexity of P_n (the nth iteration of an RSI program) is not greater than $O(\log n)$ implying a very limited amount of knowledge gain would be possible in practice despite theoretical possibility of RSI systems [18]. Yudkowsky also considers possibility of receiving only logarithmic returns on cognitive reinvestment: $\log(n) + \log(\log(n)) + \ldots$ in each recursive cycle [19].

Other limitations may be unique to the proposed self-improvement approach. For example Levin type search through the program space will face problems related to Rice's theorem [20] which states that for any arbitrarily chosen program it is impossible to test if it has any non-trivial property such as being very intelligent. This testing is of course necessary to evaluate redesigned code. Also, universal search over the space of mind designs which will not be computationally possible due to the No Free Lunch theorems [21] as we have no information to reduce the size of the search space [22]. Other difficulties related to testing remain even if we are not taking about

arbitrarily chosen programs but about those we have designed with a specific goal in mind and which consequently avoid problems with Rice's theorem. One such difficulty is determining if something is an improvement. We can call this obstacle – "multi-dimensionality of optimization".

No change is strictly an improvement; it is always a tradeoff between gain in some areas and loss in others. For example, how do we evaluate and compare two software systems one of which is better at chess and the other at poker? Assuming the goal is increased intelligence over the distribution of all potential environments the system would have to figure out how to test intelligence at levels above its own a problem which remains unsolved. In general the science of testing for intelligence above level achievable by naturally occurring humans (IQ < 200) is in its infancy. De Garis raises a problem of evaluating quality of changes made to the top level structures responsible for determining the RSI's functioning, structures which are not judged by any higher level modules and so present a fundamental difficulty in accessing their performance [23].

Other obstacles to RSI have also been suggested in the literature. Löb's theorem states that a mathematical system can't assert its own soundness without becoming inconsistent [24], meaning a sufficiently expressive formal system can't know that everything it proves to be true is actually so [24]. Such ability is necessary to verify that modified versions of the program are still consistent with its original goal of getting smarter. Another obstacle, called *procrastination paradox* will also prevent the system from making modifications to its code since the system will find itself in a state in which a change made immediately is as desirable and likely as the same change made later [25, 26]. Since postponing making the change carries no negative implications and may actually be safe this may result in an infinite delay of actual implementation of provably desirable changes.

Similarly, Bolander raises some problems inherent in logical reasoning with self-reference, namely, self-contradictory reasoning, exemplified by the Knower Paradox of the form - "This sentence is false" [27]. Orseau and Ring introduce what they call "Simpleton Gambit" a situation in which an agent will chose to modify itself towards its own detriment if presented with a high enough reward to do so [28]. Yampolskiy reviews a number of related problems in rational self-improving optimizers, above a certain capacity, and concludes, that despite opinion of many, such machines will choose to "wirehead" [29]. Chalmers [30] suggests a number of previously unanalyzed potential obstacles on the path to RSI software with *Correlation obstacle* being one of them. He describes it as a possibility that no interesting properties we would like to amplify will correspond to ability to design better software.

Yampolskiy is also concerned with accumulation of errors in software undergoing an RSI process, which is conceptually similar to accumulation of mutations in the evolutionary process experienced by biological agents. Errors (bugs) which are not detrimental to system's performance are very hard to detect and may accumulate from generation to generation building on each other until a critical mass of such errors leads to erroneous functioning of the system, mistakes in evaluating quality of the future generations of the software or a complete breakdown [31].

The self-reference aspect in self-improvement system itself also presents some serious challenges. It may be the case that the minimum complexity necessary to become RSI is higher than what the system itself is able to understand. We see such situations frequently at lower levels of intelligence, for example a squirrel doesn't have mental capacity to understand how a squirrel's brain operates. Paradoxically, as the system becomes more complex it may take exponentially more intelligence to understand itself and so a system which starts capable of complete self-analysis may lose that ability as it self-improves. Informally we can call it the Munchausen obstacle, inability of a system to lift itself by its own bootstraps. An additional problem may be that the system in question is computationally irreducible [32] and so can't simulate running its own source code. An agent cannot predict what it will think without thinking it first. A system needs 100% of its memory to model itself, which leaves no memory to record the output of the simulation. Any external memory to which the system may write becomes part of the system and so also has to be modeled. Essentially the system will face an infinite regress of self-models from which it can't escape. Alternatively, if we take a physics perspective on the issue, we can see intelligence as a computational resource (along with time and space) and so producing more of it will not be possible for the same reason why we can't make a perpetual motion device as it would violate fundamental laws of nature related to preservation of energy. Similarly it has been argued that a Turing Machine cannot output a machine of greater algorithmic complexity [14].

We can even attempt to formally prove impossibility of intentional RSI process via proof by contradiction: Let's define RSI R_I as a program not capable of algorithmically solving a problem of difficulty X, say X_i. If R_I modifies its source code after which it is capable of solving X_i it violates our original assumption that R_I is not capable of solving X_i since any introduced modification could be a part of the solution process, so we have a contradiction of our original assumption, and R_I can't produce any modification which would allow it to solve X_i, which was to be shown. Informally, if an agent can produce a more intelligent agent it would already be as capable as that new agent. Even some of our intuitive assumptions about RSI are incorrect. It seems that it should be easier to solve a problem if we already have a solution to a smaller instance of such problem [33] but in a formalized world of problems belonging to the same complexity class, re-optimization problem is proven to be as difficult as optimization itself [34-37].

3 Analysis

A number of fundamental problems remain open in the area of RSI. We still don't know the minimum intelligence necessary for commencing the RSI process, but we can speculate that it would be on par with human intelligence which we associate with universal or general intelligence [38], though in principal a sub-human level system capable of self-improvement can't be excluded [30]. One may argue that even human level capability is not enough because we already have programmers (people or their intellectual equivalence formalized as functions [39] or Human Oracles [40, 41]) who have access to their own source code (DNA), but who fail to understand how DNA

(nature) works to create their intelligence. This doesn't even include additional complexity in trying to improve on existing DNA code or complicating factors presented by the impact of learning environment (nurture) on development of human intelligence. Worse yet, it is not obvious how much above human ability an AI needs to be to begin overcoming the "complexity barrier" associated with self-understanding. Today's AIs can do many things people are incapable of doing, but are not yet capable of RSI behavior.

We also don't know the minimum size of program (called Seed AI [42]) necessary to get the ball rolling. Perhaps if it turns out that such "minimal genome" is very small a brute force [43] approach might succeed in discovering it. We can assume that our Seed AI is the smartest Artificial General Intelligence known to exist [44] in the world as otherwise we can simply delegate the other AI as the seed. It is also not obvious how the source code size of RSI will change as it goes through the improvement process, in other words what is the relationship between intelligence and minimum source code size necessary to support it. In order to answer such questions it may be useful to further formalize the notion of RSI perhaps by representing such software as a Turing Machine [45] with particular inputs and outputs. If that could be successfully accomplished a new area of computational complexity analysis may become possible in which we study algorithms with dynamically changing complexity (Big-O) and address questions about how many code modification are necessary to achieve certain level of performance from the algorithm.

This of course raises the question of speed of RSI process, are we expecting it to take seconds, minutes, days, weeks, years or more (hard takeoff VS soft takeoff) for the RSI system to begin hitting limits of what is possible with respect to physical limits of computation [46]? Even in suitably constructed hardware (human baby) it takes decades of data input (education) to get to human-level performance (adult). It is also not obvious if the rate of change in intelligence would be higher for a more advanced RSI, because it is more capable, or for a "newbie" RSI because it has more low hanging fruit to collect. We would have to figure out if we are looking at improvement in absolute terms or as a percentage of system's current intelligence score.

Yudkowsky attempts to analyze most promising returns on cognitive reinvestment as he considers increasing size, speed or ability of RSI systems. He also looks at different possible rates of return and arrives at three progressively steeper trajectories for RSI improvement which he terms: "fizzle", "combust" and "explode" aka "AI go FOOM" [19]. Hall [47] similarly analyzes rates of return on cognitive investment and derives a curve equivalent to double the Moore's Law rate. Hall also suggest that an AI would be better of trading money it earns performing useful work for improved hardware or software rather than attempt to directly improve itself since it would not be competitive against more powerful optimization agents such as Intel corporation.

Fascinatingly, by analyzing properties which correlate with intelligence, Chalmers [30] is able to generalize self-improvement optimization to properties other than intelligence. We can agree that RSI software as we describe it in this work is getting better at designing software not just at being generally intelligent. Similarly other properties associated with design capacity can be increased along with capacity to design software for example capacity to design systems with sense of humor and so in addition to intelligence explosion we may face an explosion of funniness.

4 RSI Convergence Theorem

A simple thought experiment regarding RSI can allow us to arrive at a fascinating hypothesis. Regardless of the specifics behind the design of the Seed AI used to start an RSI process all such systems, attempting to achieve superintelligence, will converge to the same software architecture. We will call this intuition - RSI Convergence Theory. There is a number of ways in which it can happen, depending on the assumptions we make, but in all cases the outcome is the same, a practically computable agent similar to AIXI (which is an incomputable but superintelligent agent [48]).

If an upper limit to intelligence exists, multiple systems will eventually reach that level, probably by taking different trajectories, and in order to increase their speed will attempt to minimize the size of their source code eventually discovering smallest program with such level of ability. It may even be the case that sufficiently smart RSIs will be able to immediately deduce such architecture from basic knowledge of physics and Kolmogorov Complexity [49]. If, however, intelligence turns out to be an unbounded property RSIs may not converge. They will also not converge if many programs with maximum intellectual ability exist and all have the same Kolmogorov complexity or if they are not general intelligences and are optimized for different environments. It is also likely that in the space of minds [50] stable attractors include sub-human and super-human intelligences with precisely human level of intelligence being a rare point [51].

In addition to architecture convergence we also postulate goal convergence because of basic economic drives, such as resource accumulation and self-preservation. If correct, predictions of RSI convergence imply creation of what Bostrom calls a Singleton [52], a single decision making agent in control of everything. Further speculation can lead us to conclude that converged RSI systems separated by space and time even at cosmological scales can engage in acausal cooperation [53, 54] since they will realize that they are the same agent with the same architecture and so are capable of running perfect simulations of each other's future behavior. Such realization may allow converged superintelligence with completely different origins to implicitly cooperate particularly on meta-tasks. One may also argue that humanity itself is on the path which converges to the same point in the space of all possible intelligences (but is undergoing a much slower RSI process). Consequently, by observing a converged RSI architecture and properties humanity can determine its ultimate destiny, its purpose in life, its Coherent Extrapolated Volition (CEV) [55].

5 Conclusions

Intelligence is a computational resource and as with other physical resources (mass, speed) its behavior is probably not going to be just a typical linear extrapolation of what we are used to, if observed at high extremes (IQ > 200+). It may also be subject to fundamental limits such as the speed limit on travel of light or fundamental limits we do not yet understand or know about (unknown unknowns). In this work we reviewed a number of computational upper limits to which any successful RSI system will asymptotically strive to grow, we can note that despite existence of such upper

bounds we are currently probably very far from reaching them and so still have plenty of room for improvement at the top. Consequently, any RSI achieving such significant level of enhancement, despite not creating an infinite process, will still seem like it is producing superintelligence with respect to our current state [56].

The debate regarding possibility of RSI will continue. Some will argue that while it is possible to increase processor speed, amount of available memory or sensor resolution the fundamental ability to solve problems can't be intentionally and continuously improved by the system itself. Additionally, critics may suggest that intelligence is upper bounded and only differs by speed and available info to process [57]. In fact they can point out to such maximum intelligence, be it a theoretical one, known as AIXI, an agent which given infinite computational resources will make purely rational decisions in any situation.

Others will say that since intelligence is the ability to find patterns in data, intelligence has no upper bounds as the number of variables comprising a pattern can always be greater and so present a more complex problem against which intelligence can be measured. It is easy to see that even if in our daily life the problems we encounter do have some maximum difficulty it is certainly not the case with theoretical examples we can derive from pure mathematics. It seems likely that the debate will not be settled until a fundamental unsurmountable obstacle to RSI process is found or a proof by existence is demonstrated. Of course the question of permitting machines to undergo RSI transformation is a separate and equally challenging problem.

This paper is a part of a two paper set presented at AGI2015 with the complementary paper being: "Analysis of Types of Self-Improving Software" [58].

References

1. Yampolskiy, R.V., Construction of an NP Problem with an Exponential Lower Bound (2011). Arxiv preprint arXiv:1111.0305
2. Yonck, R.: Toward a Standard Metric of Machine Intelligence. World Future Review **4**(2), 61–70 (2012)
3. Bremermann, H.J.: Quantum noise and information. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (1967)
4. Bekenstein, J.D.: Information in the holographic universe. Scientific American **289**(2), 58–65 (2003)
5. Lloyd, S.: Ultimate Physical Limits to Computation. Nature **406**, 1047–1054 (2000)
6. Sandberg, A.: The physics of information processing superobjects: daily life among the Jupiter brains. Journal of Evolution and Technology **5**(1), 1–34 (1999)
7. Aaronson, S.: Guest column: NP-complete problems and physical reality. ACM Sigact News **36**(1), 30–52 (2005)
8. Shannon, C.E.: A Mathematical Theory of Communication. Bell Systems Technical Journal **27**(3), 379–423 (1948)
9. Krauss, L.M., Starkman, G.D.: Universal limits on computation (2004). arXiv preprint astro-ph/0404510
10. Fox, D.: The limits of intelligence. Scientific American **305**(1), 36–43 (2011)
11. Einstein, A.: Does the inertia of a body depend upon its energy-content? Annalen der Physik **18**, 639–641 (1905)

12. Wheeler, J.A.: Information, Physics, Quantum: The Search for Links. Univ. of Texas (1990)
13. Schaeffer, J., et al.: Checkers is Solved. Science **317**(5844), 1518–1522 (2007)
14. Mahoney, M.: Is there a model for RSI?. In: SL4, June 20, 2008. http://www.sl4.org/archive/0806/19028.html
15. Turing, A.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society **2**(42), 230–265 (1936)
16. Wiedermann, J.: A Computability Argument Against Superintelligence. Cognitive Computation **4**(3), 236–245 (2012)
17. Wiedermann, J.: Is There Something Beyond AI? Frequently Emerging, but Seldom Answered Questions about Artificial Super-Intelligence, p. 76. Artificial Dreams, Beyond AI
18. Mahoney, M.: A Model for Recursively Self Improving Programs (2010). http://mattmahoney.net/rsi.pdf
19. Yudkowsky, E., Intelligence Explosion Microeconomics. In: MIRI Technical Report. www.intelligence.org/files/IEM.pdf
20. Rice, H.G.: Classes of recursively enumerable sets and their decision problems. Transactions of the American Mathematical Society **74**(2), 358–366 (1953)
21. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1**(1), 67–82 (1997)
22. Melkikh, A.V.: The No Free Lunch Theorem and hypothesis of instinctive animal behavior. Artificial Intelligence Research **3**(4), p43 (2014)
23. de Garis, H.: The 21st. Century Artilect: Moral Dilemmas Concerning the Ultra Intelligent Machine. Revue Internationale de Philosophie **44**(172), 131–138 (1990)
24. Yudkowsky, E., Herreshoff, M.: Tiling agents for self-modifying AI, and the Löbian obstacle. In: MIRI Technical Report (2013)
25. Fallenstein, B., Soares, N.: Problems of self-reference in self-improving space-time embedded intelligence. In: MIRI Technical Report (2014)
26. Yudkowsky, E.: The Procrastination Paradox (Brief technical note). In: MIRI Technical Report (2014). https://intelligence.org/files/ProcrastinationParadox.pdf
27. Bolander, T.: Logical theories for agent introspection. Comp. Science **70**(5), 2002 (2003)
28. Orseau, L.: Ring, M.: Self-modification and mortality in artificial agents. In: 4th international conference on Artificial general intelligence, pp. 1–10. Mount. View, CA. (2011)
29. Yampolskiy, R.V.: Utility Function Security in Artificially Intelligent Agents. Journal of Experimental and Theoretical Artificial Intelligence (JETAI), 1–17 (2014)
30. Chalmers, D.: The Singularity: A Philosophical Analysis. Journal of Consciousness Studies **17**, 7–65 (2010)
31. Yampolskiy, R.V.: Artificial intelligence safety engineering: Why machine ethics is a wrong approach. In: Philosophy and Theory of Artificial Intelligence, pp. 389–396, Springer (2013)
32. Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., May 14, 2002
33. Yampolskiy, R.V.: Computing Partial Solutions to Difficult AI Problems. In: Midwest Artificial Intelligence and Cognitive Science Conference, p. 90 (2012)
34. Böckenhauer, H.-J., Hromkovič, J., Mömke, T., Widmayer, P.: On the hardness of reoptimization. In: Geffert, V., Karhumäki, J., Bertoni, A., Preneel, B., Návrat, P., Bieliková, M. (eds.) SOFSEM 2008. LNCS, vol. 4910, pp. 50–65. Springer, Heidelberg (2008)
35. Ausiello, G., Escoffier, B., Monnot, J., Paschos, V.T.: Reoptimization of minimum and maximum traveling salesman's tours. In: Arge, L., Freivalds, R. (eds.) SWAT 2006. LNCS, vol. 4059, pp. 196–207. Springer, Heidelberg (2006)

36. Archetti, C., Bertazzi, L., Speranza, M.G.: Reoptimizing the traveling salesman problem. Networks **42**(3), 154–159 (2003)
37. Ausiello, G., Bonifaci, V., Escoffier, B.: Complexity and approximation in reoptimization. Imperial College Press/World Scientific (2011)
38. Loosemore, R., Goertzel, B.: Why an intelligence explosion is probable. In: Singularity Hypotheses, pp. 83–98. Springer (2012)
39. Shahaf, D., Amir, E.: Towards a theory of AI completeness. In: 8th International Symposium on Logical Formalizations of Commonsense Reasoning. California, March 26–28, 2007
40. Yampolskiy, R.V.: Turing test as a defining feature of AI-completeness. In: Yang, X.-S. (ed.) Artificial Intelligence, Evolutionary Computing and Metaheuristics. SCI, vol. 427, pp. 3–17. Springer, Heidelberg (2013)
41. 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, OH, USA (2012)
42. Yudkowsky, E.S.: General Intelligence and Seed AI (2001). http://singinst.org/ourresearch/publications/GISAI/
43. Yampolskiy, R.V.: Efficiency Theory: a Unifying Theory for Information, Computation and Intelligence. J. of Discrete Math. Sciences & Cryptography **16**(4–5), 259–277 (2013)
44. Yampolskiy, R.V.: AI-Complete CAPTCHAs as Zero Knowledge Proofs of Access to an Artificially Intelligent System. ISRN Artificial Intelligence **271878** (2011)
45. Turing, A.M.: On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society **42**, 230–265 (1936)
46. Bostrom, N.: Superintelligence: Paths, dangers, strategies. Oxford University Press (2014)
47. Hall, J.S.: Engineering utopia. Frontiers in AI and Applications **171**, 460 (2008)
48. Hutter, M.: Universal algorithmic intelligence: A mathematical top→ down approach. In: Artificial general intelligence, pp. 227–290. Springer (2007)
49. Kolmogorov, A.N.: Three Approaches to the Quantitative Definition of Information. Problems Inform. Transmission **1**(1), 1–7 (1965)
50. Yampolskiy, R.V.: The Universe of Minds (2014). arXiv:1410.0369
51. Yudkowsky, E.: Levels of organization in general intelligence. In: Artificial general intelligence, pp. 389–501. Springer (2007)
52. Bostrom, N.: What is a Singleton? Linguistic and Philosophical Invest. **5**(2), 48–54 (2006)
53. Yudkowsky, E.: Timeless decision theory. The Singularity Institute, San Francisco (2010)
54. LessWrong: Acausal Trade, September 29, 2014. http://wiki.lesswrong.com/wiki/Acausal_trade
55. Yudkowsky, E.S.: Coherent Extrapolated Volition. Singularity Institute for Artificial Intelligence, May 2004. http://singinst.org/upload/CEV.html
56. Yudkowsky, E.: Recursive Self-Improvement. In: Less Wrong, December 1, 2008. http://lesswrong.com/lw/we/recursive_selfimprovement/, September 29, 2014
57. Hutter, M.: Can Intelligence Explode? J. of Consciousness Studies **19**(1–2), 1–2 (2012)
58. Yampolskiy, R.V.: Analysis of types of self-improving software. In: The Eighth Conference on Artificial General Intelligence, Berlin, Germany, July 22–25, 2015