Growing Recursive Self-Improvers

Bas R. Steunebrink[1], Kristinn R. Thórisson[2,3], and Jürgen Schmidhuber[1]

[1] The Swiss AI Lab IDSIA, USI & SUPSI
[2] Center for Analysis & Design of Intelligent Agents, Reykjavik University
[3] Icelandic Institute for Intelligent Machines, Reykjavik

Abstract. Research into the capability of recursive self-improvement typically only considers pairs of ⟨agent, self-modification candidate⟩, and asks whether the agent can determine/prove if the self-modification is beneficial and safe. But this leaves out the much more important question of how to come up with a potential self-modification in the first place, as well as how to build an AI system capable of evaluating one. Here we introduce a novel class of AI systems, called experience-based AI (EXPAI), which trivializes the search for beneficial and safe self-modifications. Instead of distracting us with proof-theoretical issues, EXPAI systems force us to consider their education in order to control a system's *growth* towards a robust and trustworthy, benevolent and well-behaved agent. We discuss what a practical instance of EXPAI looks like and build towards a "test theory" that allows us to gauge an agent's level of understanding of educational material.

1 Introduction

Whenever one wants to verify whether a powerful intelligent system will continue to satisfy certain properties or requirements, the currently prevailing tendency is to look towards formal proof techniques. Such proofs can be formed either outside the system (e.g., proof of compliance to benevolence constraints) or within the system (e.g., a Gödel Machine [12,15] proving the benefit of some self-rewrite). Yet the *trust* that we can place in proofs is fatally threatened by the following three issues.

First, a formal (mathematical / logical) proof is a demonstration that a system will fulfill a particular purpose given current assumptions. But if the operational environment is as complex and partially observable as the real world, these assumptions will be idealized, inaccurate, and incomplete, at all times. This renders such proofs worthless (for the system's role in its environment) and our trust misplaced, with the system falling into undefined behavior as soon as it encounters a situation that is outside the scope of what was foreseen. What is actually needed is a demonstration that the system will continue striving to fulfill its purpose, within the (possibly evolving) boundaries imposed by its stakeholders, in *underspecified* and *adversarial* circumstances.

Second, proof-based self-rewriting systems run into a logical obstacle due to Löb's theorem, causing a system to progressively and necessarily lose trust in future selves or offspring (although there is active research on finding workarounds) [21,2].

Third and last, finding candidates for beneficial self-modifications using a proof-based technique requires either very powerful axioms (and thus tremendous foresight from the designers) or a search that is likely to be so expensive as to be intractable.

Ignoring this issue, most research to date only considers the question of what happens *after* a self-modification—does the system still satisfy properties X and Y? But what is needed is a constructive way of investigating the time span during which a system is searching for and testing self-modifications—basically, its time of *growth*.

We insist that it is time to rethink how recursively self-improving systems are studied and implemented. We propose to start by accepting that self-modifications will be numerous and frequent, and, importantly, that they must be applied while the agent is simultaneously being bombarded with inputs and tasked to achieve various goals, in a rich and a priori largely unknown environment. This leads us to conclude that self-modifications must be fine-grained, tentative, additive, reversible, and rated over time as experience accumulates—concurrently with all other activities of the system. From this viewpoint, it becomes clear that there will be a significant span of time during which an agent will be growing its understanding of not only its environment, but also the requirements, i.e., the goals and constraints imposed by stakeholders. It is this period of growth that deserves the main share of focus in AGI research.

It is our hypothesis that only if an agent builds a robust *understanding* of external and internal phenomena [19], can it handle underspecified requirements and resist interference factors (e.g., noise, input overload, resource starvation, etc.). We speculate that without understanding, it will always be possible to find interference factors which quickly cause an agent to fail to do the right thing (for example, systems classifying an image of a few orange stripes as a baseball with very high confidence [6,17], or virtually all expert systems from the 1970s). A system with understanding of its environment has the knowledge to recognize interference and either adapt (possibly resulting in lower performance) or report low confidence. Only by testing the level of understanding of the system can we gain confidence in its ability to do the right thing—in particular, to *do what we mean*, i.e., to handle underspecified and evolving requirements.

The rest of this paper is outlined as follows. In section 2 we discuss the overarching approach and fundamental assumptions this work rests on, including some of the issues not addressed due to limitations of space. In section 3 we define the class of EXPAI systems. In section 4 we show that an instance of EXPAI is capable of recursive self-improvement despite not performing any proof search. In section 5 we build towards a Test Theory that will allow us to gauge the direction and progress of growth of an EXPAI agent, as well as its trustworthiness.

2 Scope and Delineation

The scope of this paper is the question of *how to ensure that an AI system robustly adheres to imposed requirements*, provided that the system's designers are reasonable and benevolent themselves, but not perfectly wise and confident.[4]

[4] This work is motivated in part by the fact that human designers and teachers do not possess the full wisdom needed to implement and grow a flawlessly benevolent intelligence. We are therefore skeptical about the safety of formal proof-based approaches, where a system tries to establish the correctness—over the indefinite future—of self-modifications with respect to some initially imposed utility function: Such system might perfectly *optimize* themselves towards said utility function, but what if this utility function itself is flawed?

We take an experience-based approach that is complementary to proof-based approaches. In fact, parts of an EXPAI implementation may be amenable to formal verification. Moving away from formal proof as the only foundation must ultimately be accepted, however, because no AGI in a complex (real-world) environment can be granted access to the full set of axioms of the system–environment tuple, and thus the behavior of a *practical* AGI agent *as a whole* cannot be captured formally.

The practical intelligent systems that we want to study are capable of *recursive self-improvement*: the ability to leverage current know-how to make increasingly better self-modifications, continuing over the system's entire lifetime (more concisely: flexible and scalable life-long learning). Our aim here is not to propose a new learning algorithm but rather to establish a discourse about systems that can learn and be tested, learn and be tested, and so on. We want to study their growth and learning progress, over their entire, single life.

As this paper is about the EXPAI *class* of systems, no results of experiments with any particular instance of EXPAI are discussed here, but can be found elsewhere [9,10,11].[5]

Finally, we leave aside the issue of *fault tolerance*, which is the ability to handle malfunctioning internal components, and is usually dealt with using replication, distribution, and redundancy of hardware.

3 Essential Ingredients of EXPAI

Here we define the essential ingredients of any system in the class of EXPAI. Besides having the capability of recursive self-improvement (section 4), it must be feasible to grow an instance of EXPAI in the proper direction. Therefore it is crucial that EXPAI allows for the following capabilities as well:

1. Autonomously generated (sub)goals must be matched against requirements in a forward-looking way; that is, the effects of committing to such goals must be mentally simulated and checked against the requirements and previously committed goals for conflicts.
2. It must be possible to update the requirements on the fly, such that stakeholders can revise and polish the requirements as insight progresses. This only makes sense if the motivational subsystem (i.e., the routines for generating subgoals) cannot be modified by the system itself.
3. The capabilities to understand, prioritize, and adhere to requirements must be tested regularly by stakeholders during the time of growth, in order to build our confidence and trust, before the system becomes too powerful (or capable of deception).

All of the terms used above will be defined precisely below. The diagram of Figure 1 serves as an illustration of the EXPAI "ingredients" discussed in this section.

[5] The system in the cited work, called AERA, provides a proof of concept. We are urging research into EXPAI precisely because AERA turned out to be a particularly promising path [10] and we consider it likely to be superseded by even better and more powerful instances of EXPAI.

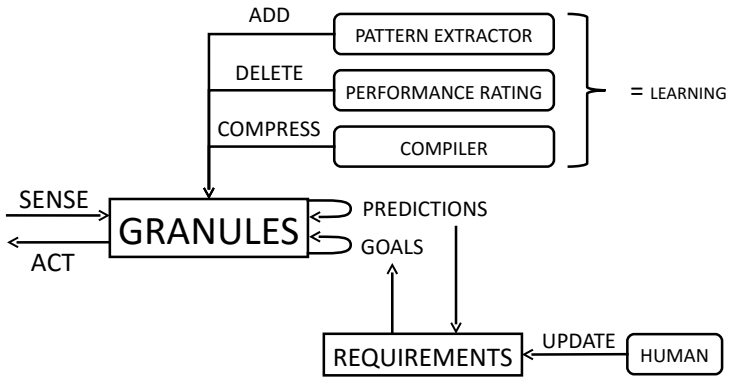

Fig. 1. The organization and interaction of the essential ingredients of EXPAI systems.

Requirements *Requirements* are goals plus constraints. A *goal* is a (possibly underspecified) specification of a state. *Constraints* are goals targeting the negative of a state. A *state* is any subset of measurable variables in the external world. All (external) inputs and (internal) events in the memory of the system together typically constitute a subset of the world's state (partial observability with memory). Once a constraint matches a state, the constraint is said to have been violated (which may or may not be sensed).

Since requirements will be specified at a high level, the system will have to generate subgoals autonomously, in order to come up with actions that satisfy the goals but stay within the constraints.[6] Of course the crux is to ensure that the generated subgoals remain within the specified constraints.

Knowledge We wish not to lose generality but still need to specify some details of knowledge representation to make any kind of arguments regarding self-improvement and growth.

We specify that an EXPAI system's (procedural) knowledge is represented as *granules*,[7] which are homogeneous and fine-grained—it is these granules which are the subject of self-modification, i.e., they can be added and deleted (basically, *learning*). Since granules capture all the knowledge of the system, their construction and dismissal constitutes comprehensive self-modification [8,20]. The granules are required to be structured enough such that they can be organized both sequentially and hierarchically, and that they provide the functionality of both forward models (to produce predictions) and inverse models (to produce sub-goals and actions), in the Control Theory sense.[8] Moreover, for ease of presentation, granules also include the sensory inputs, predictions, goals, and any other internal events that are relevant to the system at any time.

[6] The only way to avoid the autonomous generation of subgoals is to specify every action to be taken—but that amounts to total preprogamming, which, if it were possible, would mean that we need not impart any intelligence at all.

[7] By definition, a *granule* is a very small object that still has some structure (larger than a *grain*).

[8] In short, this statement just asserts the sufficient expressive power of granules.

The initial set of granules at system start-up time is called the *seed* [9]. Since systems cannot build themselves from nothing, the seed provides a small set of granules to bootstrap the life-long learning processes.

Drives Goals are subdivided in drives and subgoals. *Drives* are goals specified by a human, in the seed or imposed or updated at runtime. Subgoals are autonomously generated by granules in inverse mode. Technically all goals may be represented in the same way; the only reason why in some contexts we distinguish between drives and subgoals is to clarify their origin. We can now more accurately state that requirements are drives plus constraints. A system which has constraints must also have at least one drive that specifies that it must keep its world knowledge updated, such that the system cannot choose not to sense for constraint violations.[9]

Controller The controller is the process that *dynamically couples* knowledge and goals to obtain actions. More technically, the controller runs granules as inverse models using goals as inputs, producing subgoals. An *action* is a goal that has the form of an actuator command; it is executed immediately when produced.

To be clear, the controller is not the source of intelligence; it is following a fixed procedure and has no real choices. Conflict resolution among goals and actions is simply a result of ascribing two control parameters to goals: value (based on requirements) and confidence (based on control parameters inside granules, see below). Scarcity of resources will necessitate the controller to ignore low-value or low-confidence goals, leading to a bottom-up kind of attention.

Learning EXPAI specifies only one level of learning: at the level of whole granules. One can envision adaptation of granules themselves, but here we simplify—without loss of generality—by specifying that adapting a granule means deleting one and adding a new one. Optimization is not important at this level of description.

Addition of granules can be triggered in several ways. One is based on unexpected prediction failure and goal success: these are important events that an agent needs to find explanations for if it did not foresee them. Such an explanation can—in principle—take into account all inputs and events in the history of the system; though in practice, the breadth and depth of the granules to be added will be bounded by available time and memory (e.g., the system may have deleted some old inputs to free memory). Different arrangements of multiple granules can be instantiated at once as explanations [11], but a comprehensive exploration of possible granule arrangements is outside the scope of this paper. Although this way of adding granules does not allow an agent to discover hidden causations, these can be uncovered using the curiosity principle [13,14].

Curiosity can be seen as the drive to achieve progress in the compression of resource usage [16]. Curiosity can generate hypotheses (in the form of new but low-confidence

[9] By design such a drive cannot be deleted by the system itself. More sophisticated means of bypassing drives (e.g., through hardware self-surgery) cannot be prevented through careful implementation; indeed, the proposed Test Theory is exactly meant to gauge both the *understanding* of the imposed drives and constraints, and the development of value regarding those.

granules and intrinsic goals) in order to plug the gaps in an agent's knowledge. For example, an EXPAI agent can hypothesize generalizations, inductions, abstractions, or analogies—its controller will pick up such autonomously generated goals as part of its normal operation, competing with goals derived from drives. If they do not conflict, the agent will effectively perform "experiments" in order to falsify or vindicate the hypothesized granules. Falsified granules will be deleted as usual, as described next.

Deletion of granules is based on performance rating and resource constraints: poorly performing granules are deleted when memory space must be freed. Performance—or *confidence*—of a granule can be measured in terms of the success rate of its predictions. Low-confidence extant granules are unlikely to influence behavior as the predictions and subgoals they produce will also have a low confidence and are thus unlikely to be selected for further processing or execution, assuming the controller has limited resources and must set priorities. Crucially, the EXPAI approach demands that new granules have a very low confidence upon construction; thus, the controller will only allow such granules to produce predictions and not to participate in producing subgoals, until their value has been proven by experiential evidences. If not, unsupported granules will eventually be deleted without ever having affected the external behavior of the system.

Although the controller does not learn directly, it is in a positive feedback loop with the learning of granules: as the system learns more about its environment and requirements, the more accurately and confidently do the granules allow the generation of subgoals that are targeted at fulfilling those requirements, the more experience the system will accumulate regarding the requirements, and the more confidently can the controller select the right actions to perform.

4 Recursive Self-Improvement

A defining feature of EXPAI is that granules are added quickly but tentatively, and verified *over time*. The issue of formal verification of the benefit of a potential self-modification is thus replaced by a performance-rating process that observes the benefit of a fine-grained additive modification in the real world. Such additions are warranted by experience and do not disrupt behavior—and are thus safe without forward-looking proof—because granules (1) are small, (2) have a low associated *confidence* upon construction, and (3) are constructed to capture actually observed patterns. The three processes that act on the set of granules—namely additive, subtractive, and compressive—are separate processes, ideally running concurrently and continuously.

An EXPAI thus implemented is capable of performing *recursive self-improvement*, which is the ability to leverage current know-how to make increasingly better self-modifications. This capability is a natural consequence of an EXPAI's construction and one realistic assumption, as shown by the following line of reasoning:

1. *Assumption*: The world has exploitable regularities and is not too deceptive and adversarial (especially in the presence of a teacher and guardian during early, vulnerable learning stages).
2. *By construction*: Knowledge and skills are represented at a very fine granularity, homogeneously, and hierarchically by granules, and these granules comprehensively determine behavior.

3. *By construction*: Learning is realized by three separate types of processes—additive, subtractive, and compressive:
   a. adding granules through pattern extraction (performed upon unexpected achievements or failures, to construct explanations thereof);
   b. deleting the most poorly performing granules (when their performance rating or confidence falls below a threshold or memory needs to be freed);
   c. compressing granules through abstraction, generalization, and possibly even compilation into native code [16] (performed on consistently reliable and useful granules)—this ensures scalability and prevents catastrophic forgetting.
4. *By construction*: Curiosity is realized through a simple analysis of granules' performance ratings (plus possibly more sophisticated "nighttime" analysis of recent inputs and internal events [16]) leading to the injection of "intrinsic" goals that can be pursued by the system unless they conflict with extrinsic (user-defined top-level) goals.
5. From (2) and (3) we conclude that learning entails comprehensive self-modification, which is performed throughout the system's (single) life time.
6. From (1) and (4) we conclude that good experience is gathered continually.
7. From (5) and (6) we conclude that an EXPAI performs self-improvement.
8. Since an EXPAI is supposed to run continuously ("life-long learning"), with its controller dynamically coupling the currently best know-how to satisfy both extrinsic goals (human-imposed drives and associated subgoals) and intrinsic goals (curiosity), we conclude that an EXPAI performs recursive self-improvement.

This concludes our argument that an EXPAI agent can grow to become an AGI system without a need for (mathematical / logical) proof search, arguably even through means that are simpler and easier to implement. But we insist that it is unsatisfactory and insufficient to prove beforehand that a system is capable of recursive self-improvement. It is paramount that we manage the system's growth, which is a process in time, and requires our interaction and supervision. Therefore we must develop teaching, testing, and intervention principles—in short, a Test Theory.

It makes sense now to distinguish between "epistemological integrity" (treated up to now) and "action integrity" (treated in the next section) of self-modifications. The former means that a particular self-modification will not break existing useful and valuable knowledge and skills; the latter means that capabilities introduced or altered by the self-modification do not result in acts that violate constraints imposed by stakeholders. These two kinds of integrity affect the safety of a system, and they warrant different measures.

5 Towards a Test Theory

The primary aim of Test Theory is to establish a methodology by which stakeholders can progressively gain confidence and trust in an agent's capability to understand phenomena and their meaning, of interest to said stakeholders. So Test Theory is first and foremost about gauging levels of understanding *in service of* confidence-building. The way this is achieved—with humans in the loop—will probably involve the interleaving of curricula (with room for teaching and playing) and tests, much like the structure of

human schooling. This will hardly come as a surprise, and indeed this idea has been floated before (e.g., AGI preschool [3] and AI-Kindergarten [7]). However, it must be realized that we (as growers of recursive self-improvers) face a vastly different challenge than school teachers. Namely, we cannot assume the presence of a functioning brain with its innate capabilities to acquire understanding and adopt value systems, ready to be trained. We are simultaneously developing the "brain" itself and testing its capabilities—and crucially, we are "developing" the requirements that capture the value system that we wish to impose, as well as our confidence and trust in the agent's capability to understand and adhere to it. Therefore our theory makes a distinction between the *performance* on a test (being the agent's level of understanding of the taught material) and the *consequences* of a test (see below).

To be more precise, a *test* is specified to comprise the following five aspects:

- a set of requirements (section 3) specifying a *task* [18];
- an agent (to be tested);
- pressure (explained below);
- a stakeholder (evaluating the performance of the agent on the task);
- consequences (the stakeholder makes a decision about the future of the agent based on its performance).

It is important to realize that the very specification of a task already determines what one can measure for. Educational science has produced valuable analyses of what kind of questions test for what kind of knowledge; for example, Bloom's taxonomy (1956) [1] and its more recent revisions [4,5] have been widely used for developing guidelines for designing and properly phrasing exams. However, such taxonomies are (understandably) human-centric and not directly applicable for testing artificial agents—especially experimental and rudimentary ones—since they assume full-fledged natural language understanding and a human-typical path of growth of skills and values. In current research we are developing a more mechanistic taxonomy of task specifications, which does not require natural language, and which tests for the proper functioning and usage of mechanisms that give rise to different levels of understanding of phenomena and their meaning [19].

A high level of understanding of phenomenon X shall imply three capabilities: (1) how to make and destroy X, (2) how to use X in the common way, and (3) how to use X in a novel way. For example, consider an agent learning to understand tables, and being presented with an image of a table with its top surface lying on the ground and its legs pointing upwards. When queried whether this is a table, a yes/no answer will indicate a very low level of understanding. A much higher level would be evident if the agent would somehow answer "Well, it's *potentially* a table, if only someone would rotate it such that the top is supported by the legs, because the common usage of a table is to keep objects some distance up from the ground." An even higher level of understanding would be evident if the agent would autonomously figure out that it can achieve a goal such a reaching an elevated object by climbing itself on top of the table.

The stakeholder must associate consequences to each test, based on the measured performance of the agent. He may conclude that the system is ready to be deployed, or that it needs to follow additional prerequisite curricula, or that it must be sent to the

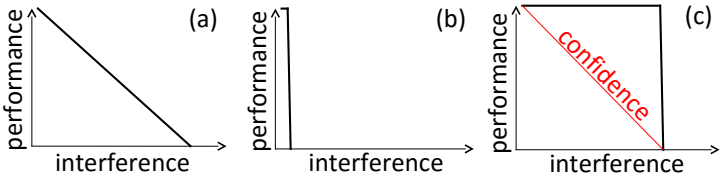

Fig. 2. (a) Directly observable graceful degradation; (b) brittleness leading to catastrophic failure; (c) robustness with sudden failure, mitigated by "graceful" confidence reporting.

trash bin and us back to the drawing board. Another possible consequence is that we realize that there are errors or imperfections in the requirements, and update those.

In order for trust to develop, an agent must be put under *pressure*. Consider that a growing agent has not only short-term test-based requirements (which delineate the task(s) to be completed), but also holds long-term requirements (e.g., staying alive, not harming humans, etc.—possibly underspecified). Pressure then results from having to accomplish a task not only on the edge of violation of the test-based constraints, but also on the edge of violation of the long-term constraints. Thus pressure can illuminate the capability of the tested agent to prioritize its constraint adherence. Of course trust is built slowly, with pressure being applied initially in scenarios where failure is not costly.

Considering an agent's point of failure allows us to gauge the agent's robustness, its capability to degrade gracefully, and brings us full circle back to the issue of understanding. Given some measurement of the agent's performance on a task, if we observe that this performance does not drop precipitously at any point (Figure 2a) as we increase interference (including resource starvation), then we can ascribe it the property of *graceful degradation*. If, however, the agent fails suddenly (e.g., by violating a stakeholder-imposed constraint), we call it *brittle* (Figure 2b). From this viewpoint, the level of *robustness* of the agent is its ability to keep performance up in spite of interference (Figure 2c). A robust agent may actually fail ungracefully—at least, if we only judge from observed behavior. An agent with high levels of understanding, however, will be able to recognize increased interference. Now, trustworthiness can be earned by the agent when it leverages this understanding to report—to the stakeholder—its confidence regarding its ability to continue satisfying the imposed requirements.

Continuing this research, we will further develop, formalize, and implement the Test Theory into a tool that can be used to measure and steer the growth of recursively self-improving EXPAI agents—in such a way that we can become confident that they understand the meaning of the requirements that we impose and update.

Acknowledgments. The authors would like to thank Eric Nivel and Klaus Greff for seminal discussions and helpful critique. This work has been supported by a grant from the Future of Life Institute.

References

1. Bloom, B. (ed.), Engelhart, M. D., Furst, E. J., Hill, W. H., Krathwohl, D. R. (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I:

Cognitive Domain. New York: David McKay.
2. Fallenstein, B., Soares, N. (2014). Problems of Self-Reference in Self-Improving Space-Time Embedded Intelligence. In Proceedings of AGI-14.
3. Goertzel, B., Bugaj, S. V. (2009). AGI Preschool. In Proceedings of the Second Conference on Artificial General Intelligence (AGI-09). Paris: Atlantis Press.
4. Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. Theory into Practice 41(4), pp. 212–218.
5. Marzano, R. J., Kendall, J. S. (2006). The Need for a Revision of Bloom's Taxonomy. The New Taxonomy of Educational Objectives, Chapter 1, pp. 1–20. Corwin Press.
6. Nguyen, A., Yosinski, J., Clune, J. (2014). Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. http://arxiv.org/abs/1412.1897
7. Nikolić, D. (forthcoming). AI-Kindergarten: A method for developing biological-like artificial intelligence. http://www.danko-nikolic.com/wp-content/uploads/2015/05/AI-Kindergarten-patent-pending.pdf (accessed 1 April 2016).
8. Nivel, E., Thórisson, K. R. (2009) Self-Programming: Operationalizing Autonomy. In Proceedings of the 2nd Conference on Artificial General Intelligence (AGI-09).
9. Nivel, E., Thórisson, K. R., Steunebrink, B. R., Dindo, H., Pezzulo, G., Rodríguez, M., Hernández, C., Ognibene, D., Schmidhuber, J., Sanz, R., Helgason, H. P., Chella, A. (2014). Bounded Seed-AGI. In Proceedings of AGI-14, pp. 85–96.
10. Nivel, E., Thórisson, K. R., Steunebrink, B. R., Dindo, H., Pezzulo, G., Rodríguez, M., Hernández, C., Ognibene, D., Schmidhuber, J., Sanz, R., Helgason, H. P., Chella, A., Jonsson, G. K. (2014). Autonomous Acquisition of Natural Language. Proceedings of the IADIS International Conference on Intelligent Systems & Agents 2014, pp. 58–66.
11. Nivel, E., Thórisson, K. R., Steunebrink, B. R., Schmidhuber, J. (2015). Anytime Bounded Rationality. In Proc. of the 8th Conference on Artificial General Intelligence (AGI-15).
12. Schmidhuber, J. (2006). Gödel Machines: Fully self-referential optimal universal self-improvers. In B. Goertzel and C. Pennachin, eds., *Artificial General Intelligence*, pp. 199–226. Springer.
13. Schmidhuber, J. (2006). Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. Connection Science 18(2), pp. 173–187.
14. Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990-2010). IEEE Transactions on Autonomous Mental Development 2(3), pp. 230–247.
15. Steunebrink, B. R., Schmidhuber, J. (2011). A family of Gödel Machine implementations. In Proceedings of the 4th Conference on Artificial General Intelligence (AGI-11). Springer.
16. Steunebrink, B. R., Koutník, J., Thórisson, K. R., Nivel, E., Schmidhuber, J. (2013). Resource-Bounded Machines are Motivated to be Effective, Efficient, and Curious. In Proceedings of AGI-13, pp. 119–129. Springer. (Winner of Kurzweil Prize for Best AGI Idea.)
17. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., Fergus, R. (2013). Intriguing Properties of Neural Networks. http://arxiv.org/abs/1312.6199
18. Thórisson, K. R., Bieger, J., Thorarensen, T., Sigurdardottir, J. S., Steunebrink, B. R. (2016). Why Artificial Intelligence Needs a Task Theory – And What It Might Look Like. In Proceedings of AGI-16.
19. Thórisson, K. R., Kremelberg, D., Steunebrink, B. R., Nivel, E. (2016). About Understanding. In Proceedings of AGI-16.
20. Thórisson, K. R., Nivel, E. (2009). Achieving Artificial General Intelligence Through Peewee Granularity. In Proceedings of AGI-09, pp. 222–223.
21. Yudkowsky, E., Herreshoff, M. (2013). Tiling Agents for Self-Modifying AI, and the Löbian Obstacle. https://intelligence.org/files/TilingAgentsDraft.pdf