Problems of self-reference in self-improving space-time embedded intelligence

Benja Fallenstein and Nate Soares

Machine Intelligence Research Institute
2030 Addison St. #300, Berkeley, CA 94704, USA
{benja,nate}@intelligence.org

Abstract. By considering agents to be a part of their environment, Orseau and Ring's *space-time embedded intelligence* [11] is a better fit to the real world than the traditional agent framework. However, a self-modifying AGI that sees future versions of itself as an ordinary part of the environment may run into problems of self-reference. We show that in one particular model based on formal logic, naive approaches either lead to incorrect reasoning that allows an agent to put off an important task forever (the *procrastination paradox*), or fail to allow the agent to justify even obviously safe rewrites (the *Löbian obstacle*). We argue that these problems have relevance beyond our particular formalism, and discuss partial solutions.

1 Introduction

Most formal models of artificial general intelligence (such as Hutter's AIXI [6] and the related formal measure of intelligence [7]) are based on the traditional agent framework, in which the agent interacts with an environment, but is not *part* of this environment. As Orseau and Ring [11] point out, this is reminiscent of Cartesian dualism, the idea that the human mind is a non-physical substance external to the body [12]. A real-world AGI, on the other hand, will be part of the physical universe, and will need to deal with the possibility that external forces might observe or interfere with its internal operations.

The traditional separation of the agent from its environment seems even less attractive when one considers I.J. Good's idea that once AGI is sufficiently advanced, it may become better than any human at the task of *making itself even smarter*, leading to an "intelligence explosion" and leaving human intelligence far behind [5]. It seems plausible that an AGI undergoing an intelligence explosion may eventually want to adopt an architecture radically different from its initial one, such as one distributed over many different computers, where no *single* entity fulfills the agent's role from the traditional framework [9]. A formal model based on that framework cannot capture this.

How should one reason about such an agent? Orseau and Ring [11] have proposed a formal model of *space-time embedded intelligence* to deal with this complexity. Their model consists of a set Π of *policies*, describing the state of the agent at a given point in time; an environment $\rho(\pi_{t+1} \mid \pi_{1:t})$, giving the probability that the policy at time $(t+1)$ will be π_{t+1}, if the policies in the previous timesteps were given by $\pi_{1:t}$; a utility function $u(\pi_{1:t}) \in [0,1]$, giving the "reward" at time t; discount factors γ_t such that $\sum_{t=1}^{\infty} \gamma_t < \infty$; and a subset $\Pi^{\tilde{l}} \subseteq \Pi$ of policies of length $\leq l$, which describes the policies that can be run on the machine initially used to implement the AGI. They then define the optimal policy as the policy $\pi^* \in \Pi^{\tilde{l}}$ which maximizes the expectation of the total discounted reward $\sum_{t=1}^{\infty} \gamma_t \, u(\pi_{1:t})$, subject to $\pi_1 = \pi^*$ and the transition probabilities $\rho(\cdot \mid \cdot)$.

Orseau and Ring propose their formalism as a tool for *humans* to reason about AGIs they might create; they argue that to choose an optimal π^* "precisely represents the goal of those attempting to build an Artificial General Intelligence in our world" [11]. By the same token, their formalism also represents the goal of *a self-improving AGI* capable undergoing an intelligence explosion, and could be used by such an AGI to reason about potential self-modifications.

Unlike agents such as Hutter's AIXI, which takes as given that future versions of itself will exist and will choose actions that maximize expected utility, an agent using this framework would see future versions of itself simply as one possible part of the future environment, and would have to convince itself that these future versions behave in desirable ways. This would allow the agent to consider radical changes to its architecture on equal footing with actions that leave its code completely unchanged, and to use the same tools to reason about both.

Such an agent would have to be *able* to reason about its own behavior or about the behavior of an even more powerful variant, and this may prove difficult. From the halting problem to Russell's paradox to Gödel's incompleteness theorems to Tarski's undefinability of truth (a formal version of the liar paradox), logic and computer science are replete with examples showing that the ability of a formal system reason about itself is often limited by diagonalization arguments, with too much power quickly leading to inconsistency. Thus, one must be very careful when specifying the mechanism by which a space-time embedded agent reasons about its potential successors, or one may end up with a system that is either too powerful (leading to inconsistencies, allowing self-modifications that are obviously bad), or not powerful enough (leading to an agent unable self-modify in ways that are obviously good).

With that in mind, in this paper we investigate in detail how a self-improving AGI can use a model similar to Orseau and Ring's to reason about its own future behavior. In particular, we consider agents that will only choose to perform an action (such as a self-modification) if they can find a proof that this action is, in a certain sense, safe. This architecture is very similar to that of Schmidhuber's *Gödel machines* [13], and is one way to approach the problem of creating an AGI that is, as Goertzel [4] puts it, *probably beneficial* and *almost certainly not destructive.*

Can such an AGI prove that it is safe for it to self-modify into an even more powerful version? We show that diagonalization problems arise even if it tries to prove the safety of a safe action that *leaves it completely unchanged.*

Intuitively, one would expect that since a human mathematician can prove that an AGI using this architecture will only take safe actions, the AGI would be able to use the same argument to prove that leaving itself unchanged is safe (by showing that it will only take safe actions in the future). However, the human's proof uses the fact that if the AGI *proves* an action to be safe, this action is *actually* safe. Löb's theorem [8], a variant of Gödel's second incompleteness theorem, implies that no consistent theory T as strong as Peano Arithmetic can show that everything provable in T is actually true:[1] The human mathematician will only be able to carry out their proof in a strictly *stronger* proof system than that used by the AGI, implying that the AGI will not be able to use this proof. Yudkowsky and Herreshoff [15] have called this the *Löbian obstacle to self-modifying AI.*

We consider two partial solutions to this problem. First, given a sound theory T_0, it is possible to construct a sound extension T_1 which can show that anything provable in T_0 is true. We show that an AGI using T_1 can prove the safety of safe actions which modify the AGI to use T_0. One can iterate this process (T_{n+1} proves that anything provable in T_n is true), but with this method, one can only build an AGI that is able to justify a fixed number of self-modifications (an AGI using T_7 can only self-modify seven times). Second, it is possible to find an infinite sequence of theories T_n in which every theory T_n proves that the *next* theory, T_{n+1}, is *consistent*; we show that under certain assumptions, this is enough to allow an AGI using T_n to prove that it is safe to self-modify into an AGI using T_{n+1}. However, neither of these solutions seems fully satisfactory.

In this work, we focus on agents that reason about their environment through formal logic (allowing for uncertainty only in the form of a probability distribution over different environments). This is not a realistic assumption. There are two reasons why we think it is still a reasonable starting point: First, although formal logic is not a good tool for reasoning about the *physical environment*, it *is* a natural tool for reasoning about the source code of future versions of an agent, and it seems likely that self-improving AGIs will need to use some form of formal logic if they want to to achieve very high confidence in a formal property of a future version's source. Second, it seems likely that many features of the following analysis will have analogs in frameworks not based on formal proofs. We give an intuitive example, based on what Yudkowsky [14] calls the "procrastination paradox", of how an agent that trusts future versions of itself too much can reason itself into taking actions that are clearly unsafe. More formally, a system due to Christiano et al. [1], which uses probabilities instead of proofs in an attempt to circumvent the Löbian obstacle, attains "too much self-trust" and succumbs to the procrastination paradox in almost the same form as proof-based systems [3].

The authors think it likely that diagonalization problems of the type discussed here will in *some* form be relevant to future AGIs, and find it plausible

[1] More precisely: T cannot prove $\Box_T\ulcorner\varphi\urcorner \to \varphi$ for every sentence φ, where $\Box_T\ulcorner\varphi\urcorner$ is the proposition in the language of T stating that the sentence φ is provable in T.

that examining partial solutions in proof-based systems can lead to insights that will help address these problems, whatever exact form they end up taking.

2 A myopic view of space-time embedded intelligence

In this section, we introduce the formal model of space-time embedded intelligence used in this paper. As in the traditional agent framework, we assume that there are finite sets $\mathcal{A}$ and $\mathcal{O}$ of actions and observations. However, instead of considering sequences of actions and observations, we take a "myopic" view that focuses even more on the initial choice of the AGI or of its programmers than Orseau and Ring's framework does, and assume that the agent makes only a single observation $o \in \mathcal{O}$ and chooses a single action $a \in \mathcal{A}$. A *policy* is thus a function $\pi \in \mathit{\Pi} := \mathcal{A}^{\mathcal{O}}$. An action a includes both *external* activities (such as a command to move a robot's arm) and the *internal* state of the agent after making the choice, making it unnecessary to draw a sharp distinction between the two. Thus, a choice to self-modify is treated no differently from a choice to perform some external activity, and even a choice *not* to self-modify is conceptually no different from a choice to self-modify.

We further assume that there is a function $\mathrm{EU} : \mathcal{O} \times \mathcal{A} \to [0,1]$ such that $\mathrm{EU}(o,a)$ specifies the expected utility when choosing action a after making observation o. This function contains essentially all of our agent's model of the environment: intuitively, it combines a probabilistic model of what happens in the world if different actions are taken with a von Neumann-Morgenstern [10] utility function which maps world histories to real numbers. (It is no restriction to require that the utility function take values in $[0,1]$, since von Neumann-Morgenstern utilities can always be represented in this way.) We do not suppose that EU is ever *evaluated* on particular o and a; rather, we merely assume that the agent has access to a *definition* of this function in the language of set theory.

The suggester-verifier architecture. We consider agents with a policy $\pi_{\mathrm{fb}} \in \mathit{\Pi}$, the *fallback policy*, which specifies a default action $\pi_{\mathrm{fb}}(o)$ to take when making observation o, and an intelligent but not fully trusted component, the *suggester* σ, which tries to improve on this default action.

In this paper, for consistency, we will always enclose formulas of first-order logic in quotes, $\ulcorner\cdot\urcorner$ (except on the right-hand side of a turnstile, as in $T \vdash 0 = 0$), and we will use overbars to denote an expression in the language of set theory denoting the *value* of a metalanguage variable: For example, if $m = 2$ and $n = 1$, then $\ulcorner\overline{m} > \overline{n}\urcorner$ denotes the formula $\ulcorner S(S(0)) > S(0)\urcorner$, where $S(\cdot)$ is the successor function.[2]

With this convention, a suggester is a deterministic algorithm taking an observation $o \in \mathcal{O}$ and returning either an error value or an action $a \in \mathcal{A}$ together with a formal proof of the sentence $\ulcorner\mathrm{EU}(\overline{o},\overline{a}) \geq \mathrm{EU}(\overline{o},\pi_{\mathrm{fb}}(\overline{o}))\urcorner$ in some formal theory T, such as $T = \mathrm{ZFC}$. Note that the exact syntactical form of this sentence

[2] Technically, the language of set theory does of course not contain a symbol for the successor function, but this can be eliminated in the usual manner.

depends on the agent's observation o and the suggester's proposed action a, since $\overline{o}$ and $\overline{a}$ denote the numerals representing the value of these metavariables.

When an agent using the suggester-verifier architecture receives an action / proof pair (a, p) from its suggester component, it checks the proof p, and executes the action a if p is a valid proof of $\ulcorner \mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\mathrm{fb}}(\overline{o}))\urcorner$. Following Schmidhuber [13], we call a sentence of this form a *target theorem*. If the suggester returns the error value, or if p is not a valid proof of the target theorem, the agent executes the fallback action $\pi_{\mathrm{fb}}(o)$.

We write $\mathrm{sv}_T(\ulcorner \mathrm{EU} \urcorner, \sigma, \pi_{\mathrm{fb}}) \in \mathit{\Pi}$ for the policy defined by this *suggester-verifier architecture*, where we use the Gödel quotes around EU to indicate that EU only needs to be given in the form of a *definition* in the language of set theory, whereas σ and π_{fb} need to be executable programs.[3] Here, T is the theory in which the suggester must prove the target theorem.

Intuitively, we have $\mathrm{EU}\big(o, \mathrm{sv}_T(\ulcorner \mathrm{EU} \urcorner, \sigma, \pi_{\mathrm{fb}})(o)\big) \geq \mathrm{EU}(o, \pi_{\mathrm{fb}}(o))$, since either $\mathrm{sv}_T(\ulcorner \mathrm{EU} \urcorner, \sigma, \pi_{\mathrm{fb}})(o) = \pi_{\mathrm{fb}}(o)$ or T proves $\ulcorner \mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\mathrm{fb}}(\overline{o}))\urcorner$, where a is the value of $\mathrm{sv}_T(\ulcorner \mathrm{EU} \urcorner, \sigma, \pi_{\mathrm{fb}})(o)$; thus, as long as T is sound (i.e., if everything T proves is actually true), the suggester-verifier architecture does at least as well as the fallback policy.

This can be understood as the "almost certainly not destructive" part of Goertzel's phrasing of the AGI safety problem: One may not be able to write down a concrete step-by-step fallback policy that solves the world's most pressing problems, but a simpler fallback policy such as "do nothing" is at least unlikely to *destroy* the world. Goertzel's "probably beneficial", on the other hand, is the suggester's job. The suggester-verifier architecture would allow an AGI's programmers to choose a suggester that they *hope* will come up with a better policy, even if they do not entirely trust it.

This architecture is very similar to that of a Gödel machine [13]; it becomes essentially the same if the fallback policy π_{fb} uses the suggester-verifier architecture to choose what actions to take in future timesteps. However, we will see below that this makes it very difficult for the suggester to prove its target theorem.

3 Self-modification

In order to model self-modification, consider an agent that has made a specific observation $o \in \mathcal{O}$ and is considering a specific action $a \in \mathcal{A}$, which will (among other things) have the effect of leaving the agent in a state in which it will use policy $\pi' \in \mathit{\Pi}$ after making its next observation. In other words, the agent expects that after it executes action a, there will be an agent in the environment which receives another observation $o' \in \mathcal{O}$ and takes another action $\pi'(o')$.

To formalize this, we assume that there is a probability distribution P' over $\mathcal{O} \cup \{*\}$, where $* \notin \mathcal{O}$ is a special error value, such that $P'(o')$ is the

[3] Our use of the symbol π_{fb} for the argument of sv_T is an abuse of notation, since sv_T depends on the *source code* of π_{fb} rather than only on the function $\pi_{\mathrm{fb}} : \mathcal{O} \to \mathcal{A}$, as π_{fb} is referenced inside the target theorem.

agent's probability that, in the next "timestep", the environment will contain an agent which makes observation o' and executes action $\pi'(o')$. $P'(*)$ denotes the probability that executing a does not in fact cause there to be an agent implementing π', for example because the agent is about to be destroyed. One can think of this as the "Cartesian boundary" between the agent and the environment breaking down.

We further assume that there is a value $\mathrm{EU}'_* \in [0,1]$ giving the expected utility the agent expects to receive if the Cartesian boundary is broken, and a function $\mathrm{EU}' : \mathcal{O} \times \mathcal{A} \to [0,1]$ such that $\mathrm{EU}'(o', a')$ is the expected utility it expects to receive if the "offspring" agent makes observation o' and is *counterfactually* assumed to take action a': that is, we assume that the agent models the world in a way that allows it to evaluate what *would* happen if its offspring took an action different from its actual one. (For example, the agent could model its world as a Bayes net.)

We assume that EU'_* and EU' are definable in the language of set theory, and that the agent is able to prove

$$T \;\vdash\; \mathrm{EU}\,(\overline{o}, \overline{a}) \;=\; P'(*) \cdot \mathrm{EU}'_* \,+\, \sum_{o' \in \mathcal{O}} P'(o') \cdot \mathrm{EU}'(o', \pi'(o')), \tag{1}$$

since this equation simply rewrites the expected utility of action a as a sum over all cases in $\mathcal{O} \cup \{*\}$, using the fact that $\mathrm{EU}'(o', \pi'(o'))$ returns the expected utility when counterfactually assuming that the offspring performs action $\pi'(o')$, which happens to be the action the offspring will *actually* perform if it observes o'.

We are particularly interested in the case where executing a implements a policy π' that again uses the suggester-verifier architecture: $\pi' = \mathrm{sv}_{T'}(\ulcorner \mathrm{EU}' \urcorner, \sigma', \pi'_{\mathrm{fb}})$ for some theory T', suggester σ', and fallback policy π'_{fb} used by the offspring agent. We further restrict consideration to policies where

$$T \;\vdash\; P'(*) \cdot \mathrm{EU}'_* \,+\, \sum_{o' \in \mathcal{O}} P'(o') \cdot \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o')) \;\geq\; \mathrm{EU}(\overline{o}, \pi_{\mathrm{fb}}(\overline{o})). \tag{2}$$

This condition says that if the agent takes action a, adopting policy π', then assuming counterfactually that the "offspring" agent takes the action $\pi'_{\mathrm{fb}}(o')$ that the new fallback policy would choose (whether or not this is equal to the offspring agent's actual action $\pi'(o')$), then the expected utility is at least that obtained by the "parent" agent's fallback policy π_{fb}. If, in addition,

$$T \;\vdash\; \forall o' \in \mathcal{O}\colon\; \mathrm{EU}'(o', \pi'(o')) \;\geq\; \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o')), \tag{3}$$

then (1) and (2) together imply that T proves the target theorem $\ulcorner \mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\mathrm{fb}}(\overline{o}))\urcorner$, allowing the parent's suggester σ to justify choosing the action a, as desired.

But *is* it provable in T that $\mathrm{EU}'(o', \pi'(o')) \geq \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o'))$? When justifying the initial choice of the suggester-verifier architecture, we proved in our informal metalanguage that $\mathrm{EU}\big(o, \mathrm{sv}_T(\ulcorner \mathrm{EU} \urcorner, \sigma, \pi_{\mathrm{fb}})(o)\big) \geq \mathrm{EU}(o, \pi_{\mathrm{fb}}(o))$ by arguing that the suggester-verifier either returns the same action as π_{fb}, or the suggester has found a proof in T that the action chosen is at least as good as that chosen by π_{fb}. We would like the parent agent to be able to justify creating an offspring of the same architecture by the same argument that we used to justify this architecture in the first place.

Here we run into the "Löbian obstacle" [15]. Our argument in the metalanguage required the fact that we believe that T is *sound*: if there is a proof in T that $\ulcorner \mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}))\urcorner$, then we believe that this inequality does in fact hold. Write $\square_{T'}\ulcorner\varphi\urcorner$ for the proposition in the language of set theory that states that φ is provable in T'. In order to carry out the intended proof in T, T needs to be able to prove that T' is sound. We assume that for every predicate $\ulcorner\varphi(n)\urcorner$ in the language of set theory,

$$T \;\vdash\; \forall n \in \mathbb{N}\colon\ \square_{T'}\ulcorner\varphi(\overline{n})\urcorner \;\to\; \varphi(n) \tag{4}$$

Since $\mathcal{O}$ and $\mathcal{A}$ are finite sets, we can identify them with subsets of $\mathbb{N}$, and conclude that for every predicate $\ulcorner\varphi(o', a')\urcorner$,

$$T \;\vdash\; \forall o' \in \mathcal{O},\, a' \in \mathcal{A}\colon\ \square_{T'}\ulcorner\varphi(\overline{o'}, \overline{a'})\urcorner \;\to\; \varphi(o', a'). \tag{5}$$

Specifically, we apply this to $\ulcorner\varphi(o', a')\urcorner = \ulcorner\mathrm{EU}'(o', a') \geq \mathrm{EU}'(o', \pi'_{\text{fb}}(o'))\urcorner$; in other words, if the inequality is provable, it is true.

Now, the parent agent's suggester can formally carry out the argument that we earlier gave in the metalanguage: by inspecting the source code of the new suggester-verifier π', it can be shown that

$$T \;\vdash\; \forall o' \in \mathcal{O}\colon\ \square_{T'}\ulcorner\varphi(\overline{o'}, \pi'(\overline{o'}))\urcorner \;\vee\; (\pi'(o') = \pi'_{\text{fb}}(o')), \tag{6}$$

since either the new suggester σ' returns a proof of a target theorem $\ulcorner\varphi(\overline{o'}, \overline{a'})\urcorner$, in which case $\pi'(o') = a'$ and thus $\ulcorner\varphi(\overline{o'}, \pi'(o'))\urcorner$ is provable, or else σ' fails, in which case $\pi'(o')$ equals the fallback action $\pi'_{\text{fb}}(o')$.

Together, (5) and (6) imply

$$T \;\vdash\; \forall o' \in \mathcal{O}\colon\ \mathrm{EU}'(o', \pi'(o')) \;\geq\; \mathrm{EU}'(o', \pi'_{\text{fb}}(o')) \tag{7}$$

(since $\pi'(o') = \pi'_{\text{fb}}(o')$ clearly implies $\mathrm{EU}'(o', \pi'(o')) = \mathrm{EU}'(o', \pi'_{\text{fb}}(o'))$, and hence also the inequality). Finally, (1), (2) and (7) together imply that T proves the target theorem $\ulcorner\mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}))\urcorner$.

4 The Löbian obstacle and the procrastination paradox

The proof in the previous section shows that it's possible to prove the target theorem $\ulcorner\mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}))\urcorner$ in T if assumptions (1), (2) and (4) are satisfied. Assumptions (1) and (2) are about the environment, essentially stating that the action a successfully executes a self-modification and that the new fallback policy does at least as well as the old fallback policy, respectively. Assumption (4), on the other hand, is an assumption about the first-order theories T and T'.

Naively, we might expect that one should be able to have $T = T'$: that our agent would be able to keep using the same theory when rewriting itself. By the diagonal lemma, it is possible to construct a theory $T = T'$ that extends ZFC and satisfies (4). However, if we substitute $\varphi(n) \equiv (0 = 1)$ in (4), we see that this axiom is saying "if a contradiction is provable in T', then a contradiction is true", which is equivalent to "no contradiction is provable in T'", or simply "T' is consistent". Hence, if we set $T = T'$, then (4) says that T shows its own consistency, which by Gödel's second incompleteness theorem implies that T is inconsistent. Löb's theorem [8] sharpens this by showing that whenever (4) holds for $T = T'$ and any $\varphi(n)$, then we must already have $T \vdash \ulcorner \forall n : \varphi(n) \urcorner$. We certainly don't want T to prove $\ulcorner \forall o' \in \mathcal{O}, a' \in \mathcal{A}$: $\mathrm{EU}'(o', a') \geq \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o'))\urcorner$, which would say that the new fallback policy is the worst of all possible policies!

Gödel's and Löb's theorems make use of much of the structure of proofs in first-order logic, and one might wonder whether these problems are mere artifacts of our use of formal proofs in the suggester-verifier architecture. The following version of the argument, however, suggests that the problem extends beyond this context.

Imagine an agent which trusts its own reasoning and has a goal that a certain button be pressed. This agent does not care *when* this button is pressed. The agent must choose between two possible actions: Either press the button immediately, or do nothing, in which case its source code remains unchanged and it will find itself in exactly the same situation in the next timestep. The agent reasons as follows: "Suppose that I don't press the button now. Then either I press the button in the next timestep, or I must have found an argument showing that even if I do not press the button in the next round, it will still get pressed eventually. Either way, the button will get pressed eventually. Therefore, it is fine not to press the button now." Thus, the agent does not press the button in the current timestep; and all subsequent timesteps are exactly the same, implying that the button is never pressed. This is called the *procrastination paradox* [14] by analogy to a human who is perenially planning to get around to doing a certain piece of work "later".

Given (4) and $T = T'$, it is not difficult to formalize this reasoning using the proof from the previous section (we set $\mathcal{O} := \{\hat{o}\}$; $\mathcal{A} := \{1, 0\}$ for pressing and not pressing the button; $\mathrm{EU}(\hat{o}, 1) := 1$; and $\mathrm{EU}(\hat{o}, 0) := \mathrm{EU}(\hat{o}, \mathrm{sv}_T(\ulcorner \mathrm{EU} \urcorner, \sigma, \pi_{\mathrm{fb}}))$, with $\pi_{\mathrm{fb}}(\hat{o}) = 1$). But at the same time, the informal reasoning does not depend on details of how our AGI works; it merely requires a sufficient amount of trust in the reasoning of the agent's successor. In fact, different formal versions of the same intuitive argument have been used in [14] and [3] to show problems with two distinct attempts at avoiding the Löbian obstacle.

What happens if we set $T = T' = \mathrm{ZFC}$, which is generally assumed to be consistent? In this case, (4) fails to hold, our proof does not go through, and we have run into the Löbian obstacle headfirst: our agent is not able to rewrite itself. But if we consider the situation of the procrastination paradox, it becomes apparent that there is an upside to this shortcoming, in that failure to choose the rewriting action 0 in this case means choosing the action 1 and actually pressing the button. Thus, we can see the Löbian obstacle as the way that ZFC solves the procrastination paradox. A good solution to the Löbian obstacle must take care not to reintroduce this paradox.

5 Partial solutions

Because of Löb's theorem, we cannot hope to have $T = T'$ if our proof is to go through: suggester-verifier agents cannot keep using the same theory to verify proofs as they self-modify. The simplest way to push our proof through, then, is one discussed in [15]: Let $T_0 :=$ ZFC and let T_{n+1} be the extension of T_n by the axiom schema $\ulcorner \forall n \in \mathbb{N}\colon \square_{T_n} \ulcorner \varphi(\overline{n}) \urcorner \to \varphi(n) \urcorner$, for every formula $\varphi(n)$. One can easily show that T_n is consistent for every n if one assumes ZFC + the existence of a strongly inaccessible cardinal, the same assumption usually made to show the consistency of ZFC. Then we can set, for example, $T = T_7$ and $T' = T_6$, which allows the parent agent's suggester to use the axiom schema to conclude the target theorem. This makes our proof go through, but it implies that our agent can only rewrite itself seven times before it reaches T_0 and cannot use the same argument to justify further rewrites. This "solution" does not seem particularly satisfactory, but it may point the way to better insights.

A more promising result [2] shows that it is possible to have an infinite sequence of sound theories T_n (i.e., theories proving only true things) such that we have

$$T_n \;\;\vdash\;\; \forall k \in \mathbb{N}\colon \square_{T_{n+1}} \ulcorner \varphi(\overline{k}) \urcorner \;\to\; \varphi(k) \tag{8}$$

for all $\varphi(k)$ of the form $\forall \ell \in \mathbb{N}.\ \psi(k, \ell)$, where $\psi(k, \ell)$ is *primitive recursive*, i.e., its truth value is a primitive recursive function of k and ℓ. This is the case, for example, if the environment is computable with a bounded amount of computational resources per timestep, and $\psi(k, \ell)$ is some property of what happens in the first ℓ timesteps, such as "up to time ℓ, the AI does not destroy the world". Since in this case, the indices n go *up* rather than down, an agent using this system is able to justify an indefinite sequence of rewrites.

An agent using the suggester-verifier architecture can make use of this result if its utility function has the form of a discounted sum over different timesteps, similar to Orseau and Ring's formalism: Suppose that EU' can be written as

$$\mathrm{EU}'(o', a') \;\;=\;\; \sum_{t=0}^{\infty} \gamma_t \cdot \mathrm{EU}'(o', a', t),$$

where $\mathrm{EU}'(o', a', t) \in [0, 1]$ is the expected "utility" received t timesteps into the future if in the next timestep, the offspring agent makes observation o' and takes action a', and γ_t is a discount factor such that $\sum_{t=0}^{\infty} \gamma_t < \infty$. Then, the predicate $\ulcorner \varphi(o', a') \urcorner = \ulcorner \mathrm{EU}'(o', a') \geq \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o')) \urcorner$ for which we need (5) to hold is provably equivalent to

$$\ulcorner \sum_{t=0}^{\infty} \gamma_t \cdot \mathrm{EU}'(o', a', t) \;\;\geq\;\; \sum_{t=0}^{\infty} \gamma_t \cdot \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o'), t) \urcorner, \tag{9}$$

which in turn is provably equivalent to

$$\ulcorner \forall T \in \mathbb{N}\colon \ \sum_{t=0}^{T} \gamma_t \cdot \mathrm{EU}'(o', a', t) \ + \sum_{t=T+1}^{\infty} \gamma_t \ \geq \ \sum_{t=0}^{T} \gamma_t \cdot \mathrm{EU}(o', \pi'_{\mathrm{fb}}(o'), t) \urcorner. \quad (10)$$

(To see this, first note that (9) is the limit of the inequality in (10) for $T \to \infty$; thus, (10) implies (9). In the other direction, note that $\gamma_t \geq \gamma_t \cdot \mathrm{EU}'(o', a', t) \geq 0$ for all o', a', and t.) Moreover, if $\mathrm{EU}'(\cdot,\cdot,\cdot)$, γ_t and $c := \sum_{t=0}^{\infty} \gamma_t$ are all rational-valued and primitive recursive, (10) has the form required in (8), because the infinite sum $\sum_{t=T+1}^{\infty} \gamma_t$ can be expressed as $c - \sum_{t=0}^{T} \gamma_t$. Thus, an agent using T_n is able to show that if a later version of the agent proves (9) in T_{n+1}, then (9) is in fact true, allowing the proof presented in Section 3 to go through.

6 Gödel machines

Our suggester-verifier formalism is very similar to Schmidhuber's Gödel machine architecture [13], and it is natural to wonder whether our techniques for justifying self-modification can be adopted to Schmidhuber's framework. Unfortunately, we run into difficulty when attempting to do so.

While the Gödel machine is somewhat more realistic than our formalism (it avoids our big discrete timesteps and has other nice properties), it is simple enough to tweak our formalism to produce something very similar to a Gödel machine. In our terminology, a Gödel machine is a suggester-verifier agent with a fallback policy that rewrites the agent into another suggester-verifier agent (with a fallback policy that rewrites the agent into yet another suggester-verifier agent, and so on). Note that we consider the agent taking no action to be a special case of this, where the agent rewrites itself into an identical agent.

Under this interpretation, it is clear that the obstacles we have encountered apply to Gödel machines as well. Consider a Gödel machine G_1 whose fallback policy would "rewrite" it into another Gödel machine G_2 with the same suggester (proof searcher, in Schmidhuber's terminology). G_1's suggester now wants to prove that it is acceptable to instead rewrite itself into G'_2, a Gödel machine with a very slightly modified proof searcher. It must prove that G'_2 will obtain at least as much utility as G_2. In order to do so, naively we would expect that G'_2 will again only execute rewrites if its proof searcher has shown them to be useful; but clearly, this runs into the Löbian obstacle, unless G_1 can show that theorems proven by G'_2 are in fact true.

We could attempt to use the techniques in this paper to make the proof go through, but this would require us to accept that G'_2 uses a strictly weaker proof system than G_1, which seems like a departure from the framework envisioned by Schmidhuber. Worse, even if we do so, it would not at all be clear how to prove (2): In the Gödel machine setting, π_{fb} and π'_{fb} would themselves be Gödel machines and hence suggester-verifiers. Thus, proving (2) would again require reasoning about agents that search for proofs: We have reduced our initial problem to a different one that is just as difficult. (In our own formalism, we solve this problem by choosing π_{fb} and π'_{fb} to be "object-level" policies involving no proof search—in the simplest case, mere lookup tables that specify which action should be taken given the observation.)

7 Conclusions

In this paper, we have introduced a concrete formalism for space-time embedded intelligence that a proof-based AGI can use to reason about its own future behavior. We have shown how, under certain assumptions, an agent using this formalism is able to justify minor self-modifications that leave its overall architecture intact.

However, in doing so, we have seen that naive approaches run into one of two major problems of self-reference: the *procrastination paradox*, which allows an agent to put off an important task forever, or the *Löbian obstacle*, which prevents an agent from justifying even clearly safe rewrites. Hurdles such as these should make the reader wary of accepting intuitively plausible formalisms allowing for self-modification before seeing a formal version that provably avoids these obstacles. We discussed partial solutions, but finding a fully satisfactory solution remains an open problem.

References

1. Paul Christiano, Eliezer Yudkowsky, Marcello Herreshoff, and Mihaly Barasz. Definability of truth in probabilistic logic. `http://intelligence.org/files/DefinabilityOfTruthInProbabilisticLogic-EarlyDraft.pdf`, 2013.
2. Benja Fallenstein. An infinitely descending sequence of sound theories each proving the next consistent. `https://intelligence.org/files/ConsistencyWaterfall.pdf`, 2013.
3. Benja Fallenstein. Procrastination in probabilistic logic. `https://intelligence.org/files/ProbabilisticLogicProcrastinates.pdf`, 2014.
4. Ben Goertzel. Golem: Toward an agi meta-architecture enabling both goal preservation and radical self-improvement. `http://goertzel.org/GOLEM.pdf`, 2010.
5. Irving John Good. Speculations concerning the first ultraintelligent machine. *Advances in computers*, 6(31):88, 1965.
6. Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability.* Springer, Berlin, 2005.
7. Shane Legg and Marcus Hutter. A formal measure of machine intelligence. In *Proc. 15th Annual Machine Learning Conference of Belgium and The Netherlands (Benelearn'06)*, pages 73–80, Ghent, Belgium, 2006.
8. M. H. Lob. Solution of a problem of Leon Henkin. *J. Symb. Log.*, 20(2):115–118, 1955.
9. Luke Muehlhauser and Laurent Orseau. Laurent Orseau on Artificial General Intelligence (interview). `http://intelligence.org/2013/09/06/laurent-orseau-on-agi/`, 2013.
10. Ludwig Johann Neumann and Oskar Morgenstern. *Theory of games and economic behavior*, volume 60. Princeton university press Princeton, NJ, 1947.

11. Laurent Orseau and Mark B. Ring. Space-time embedded intelligence. In Joscha Bach, Ben Goertzel, and Matthew Iklé, editors, *AGI*, volume 7716 of *Lecture Notes in Computer Science*, pages 209–218. Springer, 2012.
12. Howard Robinson. Dualism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2012 edition, 2012.
13. J. Schmidhuber. Ultimate cognition *à la* Gödel. *Cognitive Computation*, 1(2):177–193, 2009.
14. Eliezer Yudkowsky. The procrastination paradox. `https://intelligence.org/files/ProcrastinationParadox.pdf`, 2013.
15. Eliezer Yudkowsky and Marcello Herreshoff. Tiling agents for self-modifying AI, and the Löbian obstacle. 2013.