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* [9] is a better fit for 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 [5] and Legg's formal measure of intelligence [6]) 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 [9] point out, this is reminiscent of Cartesian dualism, the idea that the human mind is a non-physical substance external to the body [10]. 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. Moreover, a self-improving AGI may eventually want to adopt an architecture very 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 [8]. A formal model that requires the agent to always stick with a particular architecture cannot capture this.

How can we reason about such an agent, and how can such an agent reason about future versions of itself? Orseau and Ring [9] 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 argue that to choose such an optimal π^* "precisely represents the goal of those attempting to build an Artificial General Intelligence in our world". By a similar argument, it also represents the goal *of a self-improving AGI* that is deciding what its next version should be. Unlike agents such as Hutter's AIXI, which takes as given that future versions of itself will choose actions that maximize expected utility, an agent using Orseau and Ring's framework would see future versions of itself simply as another part of the 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.

However, in such a framework our agent must be *able* to reason about its own behavior or about the behavior of an even more powerful variant, and this may introduce new difficulties. 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, we need to be very careful when specifying the mechanism by which our agent reasons about its potential successors, or we might end up with a system that is either too powerful (leading to inconsistencies, so that our agent may end up self-modifying in ways that are obviously bad), or not powerful enough (so that our agent isn't able to make even self-modifications that are obviously good).

With that in mind, we will 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 approve a self-modification if they can find a proof that this modification is, in a certain sense, safe, an architecture very similar to that of Schmidthuber's *Gödel machines* [11]. This 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*. Intuitively, we expect that an AGI will regularly want to modify itself, but that it will want to leave *most* of its architecture intact *most* of the time. We use our abstract model to show that under certain assumptions about the external world, our agent will be able to justify such minor modifications.

In the course of the proof, we run into the *Löbian obstacle to self-modifying AI* described by Yudkowsky and Herreshoff [13], and see both sides of the diagonalization coin: First, we will allow our agents to use an axiom saying that they can trust the reasoning of future versions of themselves if these future versions use the same formal proof system. We then show that while this seems reasonable, it

is too powerful, for reasons closely related to Gödel's incompleteness theorems. We also give a simple intuitive example showing how this assumption can be used to justify blatantly terrible decisions, a version of what we call the "procrastination paradox" [12]. On the other side of the coin, if we base our agents on a standard, trusted theory like ZFC, our proof simply does not go through at all, and our agents aren't able to justify obviously correct self-modifications.

Despite these setbacks, there is some cause for optimism: we consider partial solutions to this problem, which give some hope that a satisfactory solution can be found. Even so, these hurdles should make us wary of accepting intuitively plausible reasoning before seeing a formal version that provably works.

In this work, we consider agents that reason about their environment through formal logic (although we allow uncertainty in the form of a probability distribution over different environments). This is not a realistic assumption. However, there are two reasons why we think it is still a reasonable starting point: First, although formal logic is not a good tool to reason about the *physical environment*, it *is* a natural tool for reasoning about the source code of future versions of our agent, and we think it is likely that self-improving AGIs will use some form of formal logic whenever they want to achieve very high confidence in a formal property of a future version's source code; and second, it seems likely that many features of our analysis will have analogs even in frameworks not based on formal proofs. For example, a system due to Christiano et al. [1], which uses probabilities instead of proofs in an attempt to circumvent the Löbian obstacle, turns out to be subject to the "procrastination paradox" in almost the same form as proof-based systems [3].

Thus, we think it likely that diagonalization problems of the type discussed here will in *some* form be relevant to future AGIs, and find it plausible 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 we will use 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 $\pi \in \Pi := \mathcal{A}^{\mathcal{O}}$. The action a specifies both *external* activities (such as a command to move a robot's arm) and the *internal* state of the agent after making the choice; thus, a choice to self-modify is treated no differently from a choice take some external action, 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 ob-

servation o. This function contains essentially all of our agent's model of the environment. We do not suppose that it is ever *evaluated* for particular o and a; rather, we merely assume that our 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 $\pi_{\text{fb}} \in \Pi$, the *fallback policy*, which specifies a default action $\pi_{\text{fb}}(o)$ 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, $\ulcorner\cdot\urcorner$ (even on the right-hand side of a turnstile, writing, for example, $\text{ZFC} \vdash \ulcorner\hat{o} \notin \hat{o}\urcorner$), 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.[1] 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 $\ulcorner\text{EU}(\overline{o}, \overline{a}) \geq \text{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}))\urcorner$, in some formal theory T, such as $T = \text{ZFC}$. The agent checks this proof and, if it is valid, takes action a. Following Schmidthuber [11], we call a sentence of the form $\ulcorner\text{EU}(\overline{o}, \overline{a}) \geq \text{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}))\urcorner$ a *target theorem*.

If the suggester returns the error value, or if it returns an invalid proof, the agent executes action $\pi_{\text{fb}}(o)$. We write $\text{sv}_T(\text{EU}, \sigma, \pi_{\text{fb}}) \in \Pi$ for the policy defined by this *suggester-verifier architecture*, although again, 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.

Intuitively, we have $\text{EU}\big(o, \text{sv}_T(\text{EU}, \sigma, \pi_{\text{fb}})(o)\big) \geq \text{EU}(o, \pi_{\text{fb}}(o))$, since either $\text{sv}_T(\text{EU}, \sigma, \pi_{\text{fb}})(o) = \pi_{\text{fb}}(o)$ or T proves $\text{EU}(o, a) \geq \text{EU}(o, \pi_{\text{fb}}(o))$, where a is the literal denoting the value of $\text{sv}_T(\text{EU}, \sigma, \pi_{\text{fb}})(o)$; thus, as long as we believe that the theorems of T are actually true, we believe that the suggester-verifier architecture does at least as well as the fallback policy.

We can see this as the "almost certainly not destructive" part of Goertzel's phrasing of the AGI safety problem: We may not be able to write down an object-level fallback policy that solves the world's most pressing problems, but we can start with "do nothing", which is at least unlikely to *destroy* the world. Goertzel's "probably beneficial", on the other hand, is the suggester's job; we choose a suggester that we have reason to *hope* will come up with a better policy, but that we don't necessarily fully trust.

This architecture is very similar to that of a Gödel machine [11]; 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 will make it very difficult for the suggester to ever prove its target.

[1] 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.

3 Self-modification

In order to model self-modification, we assume that our agent has made a specific observation $o \in \mathcal{O}$ and is considering taking 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 $\pi' \in \Pi$ after making its next observation. 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. Here, $P'(o')$ is the probability that the agent will make observation o' next, and $P'(*)$ is the probability that the Cartesian boundary between agent and environment is about to break down, so that its behavior in the next timestep is not described by making an observation and executing the corresponding action. Moreover, we assume that there is a function $\mathrm{EU}' : \mathcal{O} \times \mathcal{A} \to [0,1]$ and a value $\mathrm{EU}'_* \in [0,1]$ such that

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

intuitively, this equation states that if our agent takes action a and the Cartesian boundary breaks down, then its utility is given by EU'_*, whereas if the agent makes observation o' next, then it will take action $\pi'(o')$, and its expected utility will be given by $\mathrm{EU}'(o', \pi'(o'))$.

We are particularly interested in the case where π' again uses the suggester-verifier architecture: $\pi' = \mathrm{sv}_{T'}(\mathrm{EU}', \sigma', \pi'_{\mathrm{fb}})$ for some theory T', suggester σ', and fallback policy π'_{fb}. We assume that

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

This condition says that, if we take action a, adopting policy π', but then counterfactually assume that we next take the action $\pi'_{\mathrm{fb}}(o')$ that our new fallback policy would choose (whether or not this is equal to the actual action $\pi'(o')$), then our expected utility is at least that obtained by our old fallback policy π_{fb}. If, in addition,

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

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 our initial suggester σ to justify choosing the action a, as desired.

But *can* we show in T that $\mathrm{EU}'(o', \pi'(o')) \geq \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o'))$? When justifying our initial choice of the suggester-verifier architecture, we proved in our informal metalanguage that $\mathrm{EU}\big(o, \mathrm{sv}_T(\mathrm{EU}, \sigma, \pi_{\mathrm{fb}})(o)\big) \geq \mathrm{EU}(o, \pi_{\mathrm{fb}}(o))$ 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 our AGI to be able to justify creating a successor 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" [13]. Our argument in the metalanguage required the fact that we believe that T is *sound*: if there is a proof in T that $\ulcorner \mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\mathrm{fb}}(\overline{o}))\urcorner$, then we believe that this inequality is in fact true. Write $\Box_{T'}\ulcorner\varphi\urcorner$ 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, we assume that for every predicate $\ulcorner\varphi(n)\urcorner$ in the language of set theory,

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

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

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

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

Now, we 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

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

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 $\ulcorner\varphi(\overline{o'}, \overline{\pi'(o')})\urcorner$ is provable, or else σ' fails, in which case $\pi'(o')$ equals the fallback action $\pi'_{\mathrm{fb}}(o')$.

Together, (5) and (6) imply

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

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

4 The Löbian obstacle and the procrastination paradox

The proof in the previous section shows that it's possible to prove the target theorem $\ulcorner \mathrm{EU}(\overline{o}, \overline{a}) \geq \mathrm{EU}(\overline{o}, \pi_{\mathrm{fb}}(\overline{o}))\urcorner$ in T if assumptions (1), (2) and (4) are satisfied. Assumptions (1) and (2) are about the environment, essentially stating that the action a 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 we 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 [7] 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 $\ulcorner \forall o' \in \mathcal{O}, a' \in \mathcal{A}\text{: } \mathrm{EU}'(o', a') \geq \mathrm{EU}'(o', \pi'_{\mathrm{fb}}(o')) \urcorner$, 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: Suppose that our agent's goal is that a certain button get pressed, but the agent does not care *when* this button is pressed. The agent must choose between two possible actions: Either press the button now, 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. Our 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. Therefore, it is fine not to press the button now." Thus, our 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. We call this the *procrastination paradox* 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 $\mathrm{EU}(\hat{o}, 0) := \mathrm{EU}(\hat{o}, \mathrm{sv}_T(\mathrm{EU}, \sigma, \pi_{\mathrm{fb}}))$, 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 [12] 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: 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; our agents cannot keep using the same theory. The simplest way to

push our proof through, then, is one discussed in [13]: Let $T_0 := \text{ZFC}$ and let T_{n+1} be the extension of T_n by the axiom schema $\ulcorner \forall n \in \mathbb{N}\colon \Box_{T_n} \ulcorner \varphi(n) \urcorner \to \varphi(n) \urcorner$, 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 $T' = T_6$, and 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 we reach 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 is given by [2], which 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

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

for all $\varphi(k)$ of the form $\forall \ell \in \mathbb{N}.\ \psi(k, \ell)$, where $\psi(k, \ell)$ is *decidable*, i.e., its truth value is a computable function of k and ℓ. This is the case, for example, if the environment is computable 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.

We can make use of this in a variant of our formalism if we replace our general expected utility function $\text{EU}(o, a)$ by a discounted sum over different timesteps, similar to Orseau and Ring's formalism: Let $\text{EU}(o, a, t) \in [0, 1]$ be the "utility" received in the t'th future timestep if in the current timestep our agent makes observation o and takes action a, and let γ_t be discount factors such that $\sum_{t=0}^{\infty} \gamma_t < \infty$; our goal is to achieve a high expected total discounted reward $\sum_{t=0}^{\infty} \gamma_t \cdot \text{EU}(o, a, t)$. Thus, we require our suggester to prove the modified target theorem

$$\ulcorner \sum_{t=0}^{\infty} \gamma_t \cdot \text{EU}(\overline{o}, \overline{a}, t) \;\;\geq\;\; \sum_{t=0}^{\infty} \gamma_t \cdot \text{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}), t) \urcorner, \tag{9}$$

which is provably equivalent to

$$\ulcorner \forall T, T' \in \mathbb{N}\colon\; \sum_{t=0}^{T} \gamma_t \cdot \text{EU}(\overline{o}, \overline{a}, t) \;+\; \sum_{t=T+1}^{\infty} \gamma_t \;\;\geq\;\; \sum_{t=0}^{T'} \gamma_t \cdot \text{EU}(\overline{o}, \pi_{\text{fb}}(\overline{o}), t) \urcorner. \tag{10}$$

(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 $\gamma_t \geq \gamma_t \cdot \text{EU}(o, a, t) \geq 0$ for all o, a, and t.) Moreover, if $\text{EU}(\cdot, \cdot, \cdot)$, γ_t and $c := \sum_{t=0}^{\infty} \gamma_t$ are all rational-valued and computable, (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. We lack the space to present all the details here, but this can be used to prove a result analogous to the one presented in Section 3.

6 Gödel machines

Our formalism is very similar to Schmidthuber's Gödel machine formalism [11], and it is natural to wonder whether our techniques for licensing self-modification can be adopted to Schmidthuber'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 some 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 a suggester-verifier agent (with a fallback policy that rewrites the agent into a suggester-verifier agent, and so on). Note that we consider the agent taking no action to be a special case wherein 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, for example, a Gödel machine attempting to self-rewrite into another Gödel machine using the same theory to verify proofs. Clearly, the existing verifier will be unable to prove the consistency of the suggested machine, as the existing theory cannot verify its own consistency: Gödel machines run afoul of the Löbian obstacle. We could overcome this obstacle by accepting that Gödel machines will only self-rewrite into systems using weaker theories, but this seems undesirable.

We run into related obstacles when Gödel machines fail to verify proofs. Remember that in order to self-rewrite, a Gödel machine must prove a target theorem stating that rewriting is better than not rewriting. But if the Gödel machine's fallback plan is to remain the same then we run into trouble: the Gödel machine cannot verify the consistency of its proof theory. If the fallback policy rewrites the agent into an inconsistent Gödel machine, then that inconsistent machine might verify any policy (if the suggester happens to spit out a proof that uses the inconsistency), and in particular, might verify an optimal policy. Because the verifier cannot rule out this possibility, the verifier will be unable to prove that any given action performs strictly better than the fallback action. Thus, our proof does not go through for Gödel machines.

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. We've discussed some partial

solutions, but finding a fully satisfying solution to these issues remains an open problem.

References

1. Paul Christiano, Eliezer Yudkowsky, Marcello Herreshoff, and Mihaly Barasz. Definability of truth in probabilistic logic. `http://intelligence.org/wp-content/uploads/2013/03/Christiano-et-al-Naturalistic-reflection-early-draft.pdf`, 2013.
2. Benja Fallenstein. An infinitely descending sequence of sound theories each proving the next consistent. Technical Report 2013-6, Machine Intelligence Research Institute, Berkeley, CA, 2013.
3. Benja Fallenstein. Procrastination in probabilistic logic, 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. Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005.
6. 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.
7. M. H. Lob. Solution of a problem of Leon Henkin. *J. Symb. Log.*, 20(2):115–118, 1955.
8. Luke Muehlhauser and Laurent Orseau. Laurent Orseau on Artificial General Intelligence (interview). `http://intelligence.org/2013/09/06/laurent-orseau-on-agi/`, 2013.
9. 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.
10. Howard Robinson. Dualism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2012 edition, 2012.
11. J. Schmidhuber. Ultimate cognition *à la* Gödel. *Cognitive Computation*, 1(2):177–193, 2009.
12. Eliezer Yudkowsky. The procrastination paradox. Technical report, Machine Intelligence Research Institute, Berkeley, CA, 2013.
13. Eliezer Yudkowsky and Marcello Herreshoff. Tiling agents for self-modifying AI, and the Löbian obstacle. 2013.