Understanding Trust*

Abram Demski, Norman Hsia, and Paul Rapoport

March 11, 2025

Abstract

We examine conditions under which agents maintain their properties over time (the properties "tile"). Historically, tiling has been studied in the case of logical agents which must prove that their actions achieve some criteria [YH13]. However, this faced serious obstacles due to Löb's Theorem. Stuart Armstrong noted that this obstacle does not apply to agents who reason probabilistically [Arm13], but he did not provide a positive tiling result on that basis. The present work sketches basic tiling results for agents who make decisions using expected utility maximization, studying several versions of updateless decision theory.

1 Tiling & Trust

For safety purposes, we want to establish rigorously that an AI system ensures some desirable property or properties, such as avoiding catastrophic harm to humans. This is already difficult. However, for the highly intelligent and agentic AI systems of the foreseeable future, it is not enough to design an AI system that initially has such properties – we must also ensure that it will not be motivated to modify or remove safeguards, either through direct self-modification or through manipulative influence over humans.

This leads us to study what properties of agents persist over time. We will say that a set of properties of an agent ***tiles*** if the agent conforming to those properties is not motivated to change them over time.[1] The term "tiling" reflects the idea that the property should repeat over time, as a tiling of the plane repeats over space.

To make this more precise, we need to distinguish between agent-moment (an agent at a specific moment in time, when a single decision occurs) and agent-over-time (a collection of agent-moments which we group together into a single identity). Consider two agent-moments which I'll call "trustor" and "trustee". I'll say that the trustor ***trusts*** the trustee when the trustor does not prefer to modify the trustee, given the opportunity. (Such a claim needs to be made relative to a specific space of possible modifications.)

Our more-precise notion of tiling is now: a set of properties tiles across an agent-over-time if earlier agent-moments trust later agent moments (with respect to a set of modifications which can violate the desired properties).

It is not generally plausible that a desirable property tiles in isolation. To argue that a desirable property tiles, we need to know enough about an agent to put together a tiling argument. Any properties involved in the tiling argument need to tile over time as well, or the argument will fail. Therefore, we typically want to investigate whether a whole abstract description of an agent-architecture will tile[2] – either because we expect powerful AI systems to be able to gain access to their own source code or neural weights and modify themselves, or out of an abundance of caution even if we think we have ruled out the possibility of self-modification.

*We are grateful to Hanna Gábor, and Roman Malov for their valuable feedback on an earlier draft of this paper.

[1] Note, however, that our formalism will avoid an explicit treatment of time or causality. The notion that any modification an agent makes to itself will only impact its *future* is relegated to the setup of a specific decision problem, rather than forced by the formalism.

[2] An example of the sort of property *not* included in such an 'abstract description' might be the specifics of the hardware; so long as the hardware provides adequate guarantees with respect to running the software faithfully, it isn't important.

We can also consider trust between two agent-moments which are not part of one agent-over-time, which creates more applications for tiling theorems. We can think of four major areas of application which motivate tiling theorems for AI safety work:

1. **AI → AI** Preservation of desirable properties in AI systems.
2. **Agent → Agent** Generic tiling can be treated as a rationality condition.
3. **Human → AI** The AI alignment problem; justified trust in AI systems.
4. **AI → Human** Corrigibility [SFAY15] and non-manipulation of humans; the AI is content to let humans pursue their own actions.

The AI-to-AI motivation is what has been discussed so far: tiling theorems allow us to examine the stability of safety properties over time.

The more general agent-to-agent case can teach us about rationality. If a decision theory *itself recommends* switching to a different decision theory, then it is in some sense inconsistent. This turns out to be a rather strict rationality condition; for example, Causal Decision Theory (CDT) fails to satisfy this condition in the case of Newcomb's Problem. Conclusions drawn from approaching rationality in this way have some impact on strategic considerations for AI safety, such as reasoning about coordination between powerful AI agents.

Taking humans as trustor and AI systems as trustee, we get a version of the AI alignment problem: when can humans justifiably trust decisions made by AI systems? This is, perhaps, even more important than the AI-to-AI tiling problem. AI-to-AI tiling helps us preserve desirable safety properties; human-to-AI tiling helps us to think about what safety properties are desirable. We ask the question: if we build AIs according to a specific proposal, would humans have cause to wish it were otherwise?

Considering AI systems as trustor and humans as trustee gets at a different aspect of AI safety: corrigibility [SFAY15]. A central aspect of corrigibility is this: when a human approaches an AI system with the intent to modify it, the AI system should not want to interfere (encouraging, preventing, or otherwise manipulating the human). For example, would the AI system have any preference to interfere with a human trying to shut down the AI system? [Tho24] Tiling frames this in terms of the AI trusting the humans.

Tiling theorems have a long way to go before they can provide significant practical advice relating to these four application areas. Our hope is that tiling theorems can be refined to that point by starting in the easiest place, the study of abstract rationality, and working to make these theorems more realistic and applicable.

So far, tiling has mainly been studied in the context of agents who reason logically [YH13]. In that context, Löb's Theorem creates a significant obstacle (see [YH13] for details). However, Stuart Armstrong observed that Löb's Theorem does not hold for probabilistic reasoning [Arm13]. This suggests a possibility of much simpler tiling results for such settings. The present work provides basic results in this direction.

The results to follow are *mathematical sketches*. We present them here in preliminary form, with hope that they can be further developed.

2 Formalism

Let $\mathcal{O}$ denote the space of observations and $\mathcal{A}$ denote the space of actions, with $\mathcal{A}_o \subseteq \mathcal{A}$ being the specific actions available when facing observation o. If $o_1 \neq o_2$, then $\mathcal{A}_{o_1} \cap \mathcal{A}_{o_2}$ is empty. Let Π denote the set of all policies $\pi : \mathcal{O} \to \mathcal{A}$ which map observations $o \in \mathcal{O}$ to actions $a \in \mathcal{A}_o$. Note that any persistent internal state (ie, memory) is treated as part of the observation. The actual policy chosen by the agent will be denoted by π^*.

A policy-point is a pair $(o, a) \in \mathcal{O} \times \mathcal{A}_o$. We will treat a policy as a set of policy-points, so that $(o, a) \in \pi$ is synonymous with $\pi(o) = a$.

We distinguish a subset $\mathcal{A}^m \subseteq \mathcal{A}$ of self-modifying actions, and denote the complementary subset of non-self-modifying actions by $\mathcal{A}^{-m} = \mathcal{A} - \mathcal{A}^m$. Similarly, Π^m represents the set of policies which involve any self-modifying actions, and Π^{-m} its compliment.

For any $a \in \mathcal{A}^m$, let $\hat{a} \in \mathcal{A}^{-m}$ denote a non-self-modifying version of a, which appears identical from the outside perspective. For example, if $a \in \mathcal{A}^m$ involves a robot modifying its circuitry with a soldering iron, then $\hat{a}$ involves the robot pretending to use the soldering iron in a way that is indistinguishable to outside observers.

For a self-modifying action $a \in \mathcal{A}^m$, let $mod(a) \subseteq \mathcal{O} \times \mathcal{A}$ denote the set of policy-points modified by a.[3][4] These actions simply force the policy to include specific policy-points; the current formalism doesn't allow for complex self-modifying actions which check what the current policy is and then modify the policy in a way that depends on that.

Given a policy π and a particular self-modifying policy-point $\pi(o) = a \in \mathcal{A}^m$, we can eliminate a by defining a policy $elim(\pi, o, a)$:

$$elim(\pi, o, a)(o') = \begin{cases} \hat{a} & \text{if } o' = o \\ a' & \text{if } (o', a') \in mod(a) \\ \pi(o') & \text{otherwise.} \end{cases} \tag{1}$$

For any policy π, we define the effective policy $eff(\pi)$ as the limit of repeated applications of $elim$. This is a policy which achieves the same outward behavior as π without any self-modification. We need to assume that $eff(\pi)$ always exists and is uniquely defined. This is a constraint on the action-sets $\mathcal{A}_o$ available at particular observations.[5] $eff(\pi) \in \Pi - \Pi^m$. Importantly, this implies $eff(\pi) = eff(eff(\pi))$, since $eff(\pi)$ contains no self-modifications.

We will use $\wedge$ and $\vee$ for conjunction and disjunction.

$p(.)$ will always be a probability, and $E_p(.)$ an expected value taken according to probability distribution p. These will represent an agent's subjective beliefs rather than objective facts of the situation.

We will use the same notation for the meta-language in which we describe agents from the outside, and the internal language in which agents may think about themselves. For example, the expected utility of the agent taking action a in situation o (taken with respect to distribution p) will be written $E_p(u|\pi^*(o) = a)$. Read incorrectly, this could look like conditioning on a contradiction (in the case that $\pi^*(o) \neq a$) or a tautology (in the case that $\pi^*(o) = a$). Instead, when it occurs inside an expectation or a probability, the reader should understand π^* as a symbol in the agent's own internal language, which the agent uses to describe its own policy. Hence, in such a context, $\pi^*(o) = a$ means something like "I do a". If the agent does not already know the details of its own policy, this is neither a contradiction nor a tautology.

When we *do* want the reader to understand something inside an expectation or probability expression as a meta-language description rather than the agent's own internal language, we will use quasi-quotes $\ulcorner . \urcorner$. For example, the expected utility of the actual policy taken by the agent would be $E_p(u|\pi^* = \ulcorner \pi^* \urcorner)$. Here the agent is evaluating the expected utility of a sentence "I do X" where X is an explicit table[6] representation of the policy the agent will actually choose. This is like f-strings in python, where `f"Hello, {username}"` would evaluate to `"Hello, Alice"` if `username="Alice"`.

This convention would occasionally create ambiguity if the agent itself uses quasi-quotes in its thinking. e will use $\llcorner . \lrcorner$ for such a case. For example, $\arg\max_{a \in \mathcal{A}_o} E_p(u|\pi^*(o) = \ulcorner a \urcorner)$ evaluates the $\ulcorner a \urcorner$ to whatever value the arg max outside the expectation sets a to, but $E_p\big(u|\pi^*(o) = \arg\max_a E_p(u|\pi^*(o) = \llcorner a \lrcorner)\big)$ has the agent *thinking about* evaluating $\llcorner a \lrcorner$ as set by the arg max which sits inside the first expectation and outside of the second.

[3] Please note that we are not trying to embed the space of all possible policy modifications into the action-space, which would result in a recursive explosion of possible actions. Instead, think of the initial set of actions as fixed; then, we distinguish some special actions $\mathcal{A}^m$ as self-modifying, and we annotate these with a set $mod(a)$ of policy-points which they modify.

[4] Really, we imagine agents as having source code defining their behavior, and self-modifying actions would edit the source code rather than directly modifying the policy. However, it is simpler to deal with the policy-point modifications directly. A more realistic treatment should recognize that the list of policy-points modified is derived from a more fundamental understanding of the consequences of a self-modifying action.

[5] In this formalism, a specific decision problem needs to specify the available actions in a way which doesn't allow race conditions creating ambiguity in the effective policy (depending on the order in which $elim$ applies specific self-modifications), nor infinite loops of self-modifying actions which fail to converge. These restrictions impose a notion of time, or causality: for example, a self-modifying action cannot effectively change some action that was taken in the past. However, we are not convinced that the way we handle this here is the best way. For example, it forces any self-modifications to be deterministic (absolutely sure to work). We have avoided an explicit notion of time or causality in an attempt to be faithful to the spirit of Updateless Decision Theory, which avoids causal notions (in keeping with a more evidentialist conception of decision theory).

[6] The idea of writing out an explicit table will only make sense when $\mathcal{O}$ is finite. In other cases, we have to imagine that the internal language of the AI has ways to finitely represent infinite policies. As will become clear later on, we want to avoid conditioning on infinite policies (preferring UDT 1.0 to UDT 1.1).

3 Updateless Decision Theory

Updateless Decision Theory (UDT) was introduced by Wei Dai [Dai09a] in response to Eliezer Yudkowsky's early writing on Timeless Decision Theory (TDT). TDT is motivated by concepts of dynamic consistency and reflective consistency [Yud10], precursors to Yudkowsky's later work on tiling [YH13]. UDT offers an extremely simple solution to the dynamic consistency problems which plague other decision theories, preventing those decision theories from tiling. Nevertheless, to our knowledge, no tiling theorem for UDT has yet been published – a gap which the present work aims to fill.

Informally, UDT makes decisions by asking what action you would have wanted to commit to, had you considered that you might end up in this situation. Thus, at least intuitively, UDT should have no need for commitments. This is unlike the two most popular decision theories, Causal Decision Theory (CDT) and Evidential Decision Theory (EDT). Agents who follow CDT or EDT might wish they could credibly threaten other agents to enforce contracts, for example; but upon finding themselves in the situation where the contract has been broken, CDT and EDT will advise against pursuing costly follow-through on such threats.[7]

3.1 Counterfactual Mugging

One of the basic motivating problems for UDT is *counterfactual mugging* [Nes09]. The agent knows the situation is as follows: Omega, a powerful being who can predict you perfectly, flips a fair coin. If the coin lands on heads, Omega asks you for $10. If the coin lands on tails, Omega thinks about what you would have done if the coin had landed on heads. In the case that you would have given up the $10, Omega will now give you $100. If you would not have given up the $10, however, Omega gives you nothing.

Now imagine that you are in the world where Omega's coin landed heads. Omega approaches you and asks for $10. Updateful decision theories, such as CDT and EDT, will advise you to refuse to give up the $10, since this is a pure loss from an updateful perspective. UDT, on the other hand, gives up the $10; this is what you would have preferred to commit to ahead of time, before knowing whether the coin would land on heads or tails.[8]

3.2 Two Versions of UDT

Dai's first version of UDT, now called UDT 1.0, optimized only one policy-point at a time, selecting an action with maximal expected utility when considered as the response to the current observation:[9]

$$\pi^*(o) := \text{argmax}_{a \in \mathcal{A}_o} E_p(u | \pi^*(o) = \ulcorner a \urcorner)$$

The expectation E here is taken with respect to the agent's prior, p, rather than an updated probability distribution, as would be more common; hence the "updateless" in UDT. u is the agent's utility (understood as a random variable).

Dai noticed that this version of UDT fails to correctly solve several decision problems [Dai10]. The common element between these problems was *self-coordination*. For example, suppose that you offer a UDT agent a choice between pressing a red button or a green button. You then immediately erase its memory, tell it that it is now the second time it is being asked to make the choice, and ask it to choose again. You will give the agent $10 if it chooses green both times, $5 if it chooses red both times, and otherwise, nothing. (You explain this payout before the experiment, and do not erase the memory of the explanation.)

The optimal choice is to push the green button, but UDT 1.0 can choose the red button, depending on its prior; if it believes that its other instance will probably choose red, then it is a choice between (probably) $5 or (probably) $0, so choosing red is better.[10]

[7] For example, a Mutually Assured Destruction strategy prevents aggression by promising that it will always be met by retaliation. CDT or EDT cannot credibly commit to such a strategy, so long as the retaliation has some cost.

[8] One way to think about UDT is that it applies the idea of utilitarian welfare maximization to your possible selves; it happily gives up $10 for the heads-world, since this maximizes welfare overall.

[9] Technical note: notice that this definition of π^* is, in a sense, circular. This circularity can be resolved through the use of quines. This remark also applies to the definition of UDT 1.1.

[10] Notice that this conclusion depended on the fact that we told the agent that it was the second try, so the agent is choosing actions for *different* observations. If we made the two choices observationally identical, UDT 1.0 would select green every time.

In [Dai10], Dai corrected this issue by optimizing the whole policy together, rather than one choice at a time:

$$\pi^* := \text{argmax}_{\pi \in \Pi} E_p(u|\pi^* = \ulcorner \pi \urcorner)$$

Dai called this version UDT 1.1.

4 UDT 1.1 Tiling & The Vingean Principle

UDT 1.1 has a very simple tiling theorem. First, we assume that the world is "fair". In the context of tiling, a decision problem is "fair" if success depends only on the rationality of the agent, not some other features ([Yud10], Section 3). For example, it would be fair to punish agents for walking east (the agent can choose to walk a different direction); however, it would be unfair to punish agents for running on RISC processors rather than GPUs, or for being written in C++ rather than Python. One way to further articulate this idea is that it is unfair to punish an agent for the details of how they think (what Yudkowsky calls their "ritual of cognition" in [Yud10], Sections 6-7).

Therefore, we will refer to any assumption which codifies this intuition as a "fairness assumption". It seems apparent that we need to make a fairness assumption in order to prove any tiling theorem, since otherwise, the environment could simply punish tiling. We will start with the following fairness assumption:

Assumption 1 (Policy Fairness)**.** *The utility of a policy depends only on its effective behavior – that is, for any policies π_1 and π_2, if $\mathit{eff}(\pi_1) = \mathit{eff}(\pi_2)$, then $E_p(u|\pi^* = \ulcorner \pi_1 \urcorner) = E_p(u|\pi^* = \ulcorner \pi_2 \urcorner)$.*

We can now prove a simple tiling theorem for UDT 1.1:

Theorem 1. *Under the Policy Fairness assumption, UDT 1.1 does not strictly prefer any self-modifying policy.*

Proof. Consider some self-modifying policy $\pi_m \in \Pi^m$. Applying Policy Fairness, since $\text{eff}(\pi_m) = \text{eff}(\text{eff}(\pi_m))$, π_m must be just as good as $\text{eff}(\pi_m)$:

$$E_p(u|\pi^* = \ulcorner \pi_m \urcorner) = E(u|\pi^* = \ulcorner \text{eff}(\pi_m) \urcorner)$$

Thus, no self-modifying policy can be strictly preferred to all non-self-modifying policies.

□

4.1 The Vingean Principle

The above result is not very satisfying. Because the whole policy is chosen by a single maximization, there are not really two distinct decision points such that one is trusting another. Even if the computation is re-run at each decision point rather than computed once and saved, the agent is not doing any significant reasoning about other versions of itself. This reflects something unrealistic about UDT 1.1: it is not feasible to construct powerful agents by selecting each policy-point at once in one big computation. Agents realistically need to do some new work to decide what to do, when faced with a situation.

Yudkowsky & Herreshoff therefore articulate the *Vingean Principle* [YH13]: tiling results need to allow for the possibility that the trustor *cannot predict in advance what the trustee will do.* Trust needs to be established by more abstract reasoning about the properties of the trustee.

UDT 1.0 seems more promising in this regard: policy-points are considered individually, under uncertainty about how other policy-points are chosen.

5 UDT 1.0 Tiling

In order to prove a tiling result for UDT 1.0, we need to assume that the situation faced by the agent lacks any of the coordination problems which plague UDT 1.0; otherwise, there would be an incentive for the agent to modify itself towards UDT 1.1.

Assumption 2 (Policy Coordination)**.** *For any policy $\pi \in \Pi$, if $E_p(u|\pi^* = \ulcorner\pi\urcorner) \geq E_p(u|\pi^* = \ulcorner\pi^*\urcorner)$, then for all observations $o \in \mathcal{O}$:*

$$E_p(u|\pi^*(\ulcorner o\urcorner) = \ulcorner\pi(o)\urcorner) \geq E_p(u|\pi^*(\ulcorner o\urcorner) = \ulcorner\pi^*(o)\urcorner)$$

This assumption states that if a policy is at least as good as π^* overall, then individual actions from that policy should be judged at least as good as taking the actions specified by π^*. This implies the agent cannot be stuck in a locally-optimal but globally-suboptimal policy.

We can now prove a simple tiling theorem for UDT 1.0:

Theorem 2. *Under Policy Fairness and Policy Coordination, UDT 1.0 does not strictly prefer any self-modifying action.*

Proof. Suppose for the sake of contradiction that at some observation o, UDT 1.0 strictly prefers a self-modifying action $a_m \in \mathcal{A}^m$:

$$E_p(u|\pi^*(\ulcorner o\urcorner) = \ulcorner a_m\urcorner) > E_p(u|\pi^*(\ulcorner o\urcorner) = \ulcorner a'\urcorner) \text{ for all } a' \in \mathcal{A}_o - \mathcal{A}^m$$

By Policy Fairness:

$$E_p(u|\pi^* = \ulcorner\text{eff}(\pi^*)\urcorner) = E_p(u|\pi^* = \ulcorner\pi^*\urcorner)$$

Therefore, by Policy Coordination:

$$E_p(u|\pi^*(\ulcorner o\urcorner) = \ulcorner\text{eff}(\pi^*)(o)\urcorner) \geq E_p(u|\pi^*(\ulcorner o\urcorner) = \ulcorner\pi^*(o)\urcorner)$$

Since $\text{eff}(\pi^*) \notin \Pi^m$, this contradicts the initial assumption. □

This is better than the UDT 1.1 tiling theorem, but it is still not very satisfying. The coordination assumption plays too large a role. Trust is proven *directly* from the assumption that the agent doesn't fall into any self-coordination traps. This means the agent doesn't trust itself due to features of its decision procedure; instead, it trusts itself due to a bold and unjustifiable assumption that it believes its own decisions are good. This fails to shed the desired light on *how* an agent can trust itself. We *need* to make some assumption about the absence of coordination problems; but, can we do it in a way that requires the agent to reason about its own decisions more realistically?

Here are some modified assumptions:[11][12]

Assumption 3 (Limited Self-Modification)**.** *Any self-modifying action $a_m \in \mathcal{A}^m$ modifies exactly one policy-point $(ob(a_m), ac(a_m))$, where $ac(a_m) \notin \mathcal{A}_m$ (self-modifications can't force further self-modifications) and $a_m \notin \mathcal{A}_{ob(a_m)}$ (an action can't modify its own policy-point).*

This assumption limits self-modifying actions to only change one policy-point, and more importantly, not force other self-modifying actions. This avoids chains of self-modifications which propagate other self-modifications.

Assumption 4 (Fine-Grained Fairness)**.** *For any self-modifying action $a_m \in \mathcal{A}^m$:*

$$E_p(u|\pi^*(o) = \ulcorner a_m\urcorner) = E_p(u|\pi^*(o) = \ulcorner \hat{a_m}\urcorner \wedge \pi^*(\ulcorner ob(a_m)\urcorner) = \ulcorner ac(a_m)\urcorner)$$

Here, $ob()$ and $ac()$ are the same as defined by the previous assumption. This assumption says that the expected utility of taking a self-modifying action a_m is equal to the expected utility of the corresponding non-self-modifying action $\hat{a}_m$ additionally conditioned on the proposition that the policy already takes the action which the self-modification would have forced.

Assumption 5 (Faith in Joint Argmax)**.** *For any subset of actions $\mathcal{A}' \subseteq \mathcal{A}$ and any pair of observations $o \neq o'$,*

$$\max_{a \in \mathcal{A}'} E_p\Big(u|\pi^*(o) = \ulcorner a\urcorner \wedge \pi^*(o') = \arg\max_{a' \in \mathcal{A}} \max_{a'' \in \mathcal{A}'} E_p\big(u|\pi^*(o) = \llcorner a''\lrcorner \wedge \pi^*(o') = \llcorner a'\lrcorner\big)\Big)$$
$$\geq \max_{a' \in \mathcal{A}} \max_{a \in \mathcal{A}'} E_p\big(u|\pi^*(o) = \ulcorner a\urcorner \wedge \pi^*(o') = \ulcorner a'\urcorner\big)$$

[11] The approach was inspired by work done by Linda Linsefors and Alex Mennen during an internship at the Machine Intelligence Research Institute.

[12] An earlier variation on this theorem and proof is presented in Appendix A. We judged the version in the main text to be an improvement, but the other version may still be of interest to some readers.

This assumption says that when the agent takes, in response to the second observation, the argmax action dictated by its self-determined joint maxima over the space $\mathcal{A}' \times \mathcal{A}$ of action pairs, the expected value attained by the best response to the first observation is not lower than the value attained by the actual joint maxima. In essence, this assumption validates the agent's internal process for determining one of the components of the joint maxima.

Assumption 6 (Action Coordination)**.** *For any pair of observations $o \neq o'$,*

$$p\Big(\arg\max_{a' \in \mathcal{A}} E_p\big(u|\pi^*(o') = \llcorner a' \lrcorner\big) = \arg\max_{a' \in \mathcal{A}} \max_{a \in \mathcal{A}^{-m}} E_p\big(u|\pi^*(o) = \llcorner a \lrcorner \wedge\ \pi^*(o') = \llcorner a' \lrcorner\big)\Big) = 1.$$

This assumption says that the agent expects the best action it chooses in response to the primed observation to still maximize expected utility under the additional condition that, given its response to the primed observation, the agent is choosing the best non-self-modifying response to the second observation.

Assumption 7 (Knowledge of Decision Procedure)**.**

$$p\big(\pi^*(o) = \arg\max_{a \in \mathcal{A}} E_p(u|\pi^*(o) = \llcorner a \lrcorner)\big) = 1$$

Note, critically, that there are not quasi-quotes around the whole argmax expression; this isn't saying that the agent knows what specific actions it takes, only that it knows in the abstract that whatever it does is the output of the UDT 1.0 decision rule.

Theorem 3. *Assuming Fine-Grained Fairness, Limited Self-Modification, Faith in Joint Argmax, Action Coordination, and Knowledge of Decision Procedure, UDT 1.0 does not strictly prefer any self-modifying action.*

Proof.
Let $a_m \in \mathcal{A}^m$ be an arbitrary self-modifying action, and let $(\bar{o}, \bar{a}) = \big(\mathrm{ob}(a_m), \mathrm{ac}(a_m)\big)$. By Fine-Grained Fairness, the expectation of the self-modifying action is equal to the expectation of the corresponding non-self-modifying action $\hat{a_m}$, when conditioned on knowledge that the forced action would be taken anyway:

$$E_p(u|\pi^*(o) = a_m) = E_p\big(u|\pi^*(o) = \hat{a_m} \wedge\ \pi^*(\ulcorner \bar{o} \urcorner) = \ulcorner \bar{a} \urcorner\big)$$

By definition of max and the fact that $\hat{a_m}$ is non-self-modifying, this expectation is not greater than the expectation of the best non-self-modifying action which could be substituted for $\bar{a}$:

$$\leq \max_{a \in \mathcal{A}^{-m}} E_p\big(u|\pi^*(o) = \ulcorner a \urcorner \wedge\ \pi^*(\ulcorner \bar{o} \urcorner) = \ulcorner \bar{a} \urcorner\big)$$

By definition of max, this expectation is not greater than the expectation of the best action which could be substituted for $\hat{a_m}$:

$$\leq \max_{a' \in \mathcal{A}} \max_{a \in \mathcal{A}^{-m}} E_p\big(u|\pi^*(o) = \ulcorner a \urcorner \wedge\ \pi^*(\ulcorner \bar{o} \urcorner) = \ulcorner a' \urcorner\big)$$

By Generalized Faith in Argmax, this expectation is not greater than the expected utility conditioning on the abstract statement that the second action chosen will be a component of the joint maxima (with the first action still being the concrete maximum), rather than conditioning on the actual joint maxima:

$$\leq \max_{a \in \mathcal{A}^{-m}} E_p\Big(u|\pi^*(o) = \ulcorner a \urcorner \wedge\ \pi^*(\ulcorner \bar{o} \urcorner) = \arg\max_{a' \in \mathcal{A}} \max_{a'' \in \mathcal{A}^{-m}} E_p\big(u|\pi^*(o) = \llcorner a'' \lrcorner \wedge\ \pi^*(\ulcorner \bar{o} \urcorner) = \llcorner a' \lrcorner\big)\Big)$$

By Weak Action Coordination, in the second conditional, we may replace the iterated maximization over both conditionals by an argmax over the second conditional:

$$= \max_{a \in \mathcal{A}^{-m}} E_p\Big(u|\pi^*(o) = \ulcorner a \urcorner \wedge\ \pi^*(\bar{o}) = \arg\max_{a' \in \mathcal{A}} E_p\big(u|\pi^*(\ulcorner \bar{o} \urcorner) = \llcorner a' \lrcorner\big)\Big)$$

By Knowledge of Decision Procedure, the argmax condition could be dropped entirely:

$$= \max_{a \in \mathcal{A}^{-m}} E_p\big(u|\pi^*(o) = \ulcorner a \urcorner\big)$$

Putting it all together, the self-modifying action a_m is at most just as good as the best non-self-modifying action:

$$E_p(u|\pi^*(o) = a_m) \leq \max_{a \in \mathcal{A}^{-m}} E_p\big(u|\pi^*(o) = \ulcorner a \urcorner\big)$$

Since a_m is arbitrary, this implies that no self-modifying action is strictly preferred. □

This puts us in a better position. The tiling result is Vingean: the agent is not assumed to know precisely which actions it eventually takes, only that those actions are consistent with the definition of UDT 1.0. The result successfully provides a picture of an agent reasoning about its own decision procedure in order to decide that its actions are trustworthy, and therefore, do not require self-modification.

The restrictions to self-modification are obviously severe. Our intuition is that it should be possible to weaken Limited Self-Modification significantly, but that some limits will still be reflected in the most realistic tiling theorems, particularly those that deal with learning agents: an agent cannot learn about the quality of self-modifications which permanently change the agent (either directly changing infinitely many policy-points, or doing so indirectly through an infinite chain of self-modifications), because it will not observe the consequences at finite time.

One weakness of this proof is that it relies a lot on the agent's ability to reason about itself. In particular, expressions which nest one expectation inside of another expectation apparently require a probability distribution to reason about itself. Paul Christiano et al investigate the limitations of such reasoning and propose a framework for self-referential probabilistic reasoning to handle these limitations without contradiction [CYHB13]; working within such a framework to explicitly handle these issues would improve the result. However, since that system is not itself computationally feasible, such an improved result would remain somewhat unrealistic.

Another weakness is that the Faith in Joint Argmax assumption is rather strong. It does not merely say that the agent abstractly knows "the action which comes out of an argmax achieves the max value" – it says that in particular, the expectation conditioned on argmaxing is at least the maximal expectation. In other words, it might not know the specific action chosen by the argmax, but it *is* required to know the *exact value* of that action (or, perhaps, overestimate that value). This is like knowing *how good* your next move in chess will be *before* you know specifically what that move will be. As such, this is *almost* a violation of the Vingean Principle: we don't require the agent to know its future actions, but we require its estimates of the quality of an argmax to be as good as if it did.

Both of these weaknesses have to do with questions of embedded agency [DG19] and bounded rationality. People familiar with the discourse around the Machine Intelligence Research Institute (MIRI) will say that these are problems of "logical uncertainty" – although we have come to prefer the term "computational uncertainty" ourselves. A realistic tiling theorem needs to tackle these questions head-on, providing a believable framework for reasoning about computationally uncertain agents. The remainder of this text briefly summarizes our current speculations about what this framework should look like.

6 Computational Uncertainty

We start with the framework of Logical Induction proposed by Garrabrant et al [GBTC+16]. To briefly summarize:

Many Bayesians are used to thinking of the Bayesian framework for uncertain reasoning as being underpinned and justified by Dutch Book arguments [Vin22]: if you violate the laws of probability theory (as spelled out by the Kolmogorov axioms), then it is possible for a bookie to offer you some set of bets which you will accept, but which are guaranteed to extract money from you. (This set of bets is the so-called "Dutch Book".)

However, Dutch Book arguments tacitly assume that the bookie knows "exactly as much as you"; after all, it would not be a fair test of your rationality otherwise. *Of course* a bookie can extract money from you if they know something you don't know. (Indeed, your assumed willingness to take bets proposed by the bookie is questionable in this case.)

Unfortunately, this assumption is not so clear in the case of computational uncertainty. The bookie is ordinarily assumed to know all logical consequences of beliefs, and use this knowledge in arranging Dutch Books. When computational uncertainty is involved, the boundary around what is known is fuzzier; inferences can be easier or harder, rather than known vs unknown.

Logical Induction therefore discards this idea. Instead, we want to construct epistemic states which are not *easily* Dutch Booked. This notion of "easily" needs to account for both description complexity [MV90] and computational complexity [BBJ02]. Roughly speaking, the description complexity accounts for empirical uncertainty, while the computational complexity accounts for computational uncertainty. The resulting framework has a host of desirable properties, including properties addressing the sorts of self-referential reasoning needed in the previous section. See [GBTC+16] for further details.

However, Logical Induction does not give us everything we need. In particular, the framework does not support a suitable version of UDT. In joint work with Martín Soto [MSD23], we spelled out some additional rationality conditions to allow a version of UDT to be specified. Unfortunately, the version of UDT achieved there is not a very good one, and a tiling theorem did not result from that work.

We will now sketch, very briefly, some conditions related to that work and also incorporating more recent ideas, which we conjecture will be enough to get a nice tiling theorem. Due to space limitations, and also because these ideas are still quite preliminary, the following will be a very inadequate explanation.

6.1 What Computational Updatelessness Should Be

The first problem we encounter when trying to specify a satisfactory computationally-uncertain UDT is that the "prior" of a computationally uncertain agent is terrible. In the context of Logical Induction, the early beliefs of the Logical Inductor can be almost anything, since the agent has not yet had time to think and make its beliefs more coherent. Therefore, the basic idea of UDT – taking actions which the agent would have committed to ahead of time – is questionable in this context. The judgements rendered by the initial epistemic state of such an agent will be basically arbitrary.

Intuitively, we want the agent to treat its computational uncertainty similarly to how empirically-uncertain UDT treats its empirical uncertainty. A really bad prior should not interfere too much if we get better information later. A modification of Counterfactual Mugging illustrates this principle:

6.1.1 Counterfactual Mugging With Evidence

Consider a variation of the Counterfactual Mugging problem 3.1, in which Omega uses a *biased* coin, with the coin bias selected uniformly. Before Omega makes the official coinflip which will determine the outcome, Omega lets you see several demonstration flips, so that you can estimate the bias of the coin. For simplicity, we will consider UDT 1.1.

If the coin lands on tails, what counterfactual worlds does Omega consider when deciding whether to give you $100?

If Omega considers only what you would do given the evidence you saw (I'll call this Version A), then the optimal solution (according to UDT 1.1) is to estimate the bias of the coin using the observed coinflips, and then give up the $10 or not based on that estimated probability of the final flip. If the probability of heads is less than $\frac{10}{11}$, one should accept; if it is greater, one should refuse.

If Omega gives you the $100 with probability equal to the *total* probability that you'll give up the $10, summing over all the possible coin-flip evidence you might see before deciding (I'll call this Version B), then UDT advises you to give up the $10 just like in the original Counterfactual Mugging. The evidence you see is irrelevant; the final coin-flip is treated just as if the coin were fair, since the *average* coin bias is fair.

The point here is that UDT 1.1 behaves updatefully with respect to information *exactly when Omega's counterfactual reasoning is updateful with respect to that information.* In Version A, Omega's counterfactual reasoning updates on the preliminary coin-flips, so our reasoning should as well. In Version B, Omega doesn't update, so we shouldn't either.

Therefore, if computational uncertainty is analogous to empirical uncertainty, then computationally uncertain UDT should behave similarly – acting *as if* it had updated, whenever that is advantageous. This could help mitigate its terrible prior.

The problem, then, is to find extra rationality conditions to impose on a system similar to Logical Induction, such that it reasons in this way. In other words: what properties does a computationally uncertain "prior" need to have, in order for it to reason as above?

We would suggest generalizing the reasoning as follows: given some mutually exclusive and jointly exhaustive set of observations $\mathcal{O}'$, we ordinarily have:

$$E_p(u|\phi) = \sum_{o \in \mathcal{O}'} p(o|\phi) E_p(u|\phi, o)$$

This allows us to decompose UDT calculations into "branches" of possibility. A specific branch o_1 will "behave updatefully" if the condition ϕ has negligible impact on the expected utility of other branches ($E_p(u|\phi, o_2)$ for $o_2 \neq o_1$) and negligible impact on branch probabilities ($p(o|\phi)$ for any o). We could summarize this as: UDT behaves updatefully unless it thinks it has a good reason not to. (Wei Dai similarly observed conditions under which UDT behaves updatefully [Dai09b].)

Unfortunately, it is not yet clear how to achieve a result like this for computationally uncertain UDT, in part because it is not obvious how to spell out a notion of observation for that case.

6.1.2 What Are Computational Observations?

The second problem we encounter is a problem of *size*. In the case of empirical uncertainty, we are given a well-defined observation set $\mathcal{O}$, from which we define a notion of policy. With computational uncertainty, the "observations" are much less clear. As we think longer, we change our beliefs somehow. We can't account for these changes based on only our empirical observations. Our knowledge is no longer summarized by a set of propositions which we know, since knowledge also consists of the details of bounded inferences we make from those propositions. Therefore, we replace the notion of "observation" with the full list of our beliefs. A policy is now a function from full belief states to actions.

Unfortunately, in the Logical Induction formalism, this creates a problem for UDT. UDT needs to maximize expected value across all the possible observations, based on their probability. However, a Logical Induction epistemic state only assigns probabilities to finitely many propositions at any finite time. The list of propositions which we assign probabilities grows over time, but at any finite time, a proposition which *lists probabilities assigned to all those propositions* will be far larger than any proposition which we currently assign probabilities to.

Sam Eisenstat (in unpublished work) proposed a modification of Logical Induction which he called Bayesian Logical Induction, in which we enforce extra rationality principles for the "large" beliefs (beliefs about sentences which are not assigned probabilities by the Logical Inductor). These extra rationality constraints allow us to interpret the non-Bayesian updates of the Logical Inductor as Bayesian updates again, providing us with a notion of "observation" for UDT. Details of this construction are provided in Appendix B.

This takes care of the "size" problem. However, several more problems remain for a computationally uncertain UDT.

6.1.3 Other Rationality Constraints

Bayesian Logical Induction gives us a notion of reasoning coherently about the probability of "observations" for computationally uncertain UDT. However, UDT needs more than that. We also needs to be able to condition on policy-points, to implement the UDT decision criterion. This requires sensibly extrapolating "small" beliefs about consequences of behavior into "large" beliefs. The most direct adaptation of UDT 1.0 to this setting needs to reason sanely about conditions of the form $\pi^*(\mathcal{Q}) = a$.

We conjecture that enforcing some form of *real-value coherence* helps us here: beliefs about random variables should cohere appropriately, so that for example, the computationally uncertain expectation of random variables A and B sums to the expectation of $A + B$.

Martín Soto also proposed the constraint of *universal instantiation*: to the extent that we believe $\forall x : \phi(x)$, we should also believe $\phi(c)$, even for "large" sentences. This helps propagate knowledge about consequences from small to large beliefs.

We conjecture that some version of these assumptions allows UDT 1.0 to be adapted to a computationally uncertain setting, enabling more realistic (if still woefully unrealistic) tiling results.

References

[Arm13] Stuart Armstrong. Probabilistic Löb theorem. https://www.lesswrong.com/posts/oBDTMnEzptBidvmw7/probabilistic-loeb-theorem, 2013. Published on LessWrong, April 26, 2013.

[BBJ02] George S Boolos, John P Burgess, and Richard C Jeffrey. *Computability and logic*. Cambridge university press, 2002.

[CYHB13] Paul Christiano, Eliezer Yudkowsky, Marcello Herreshoff, and Mihaly Barasz. Definability of truth in probabilistic logic. Technical report, Machine Intelligence Research Institute, June 2013. Early draft.

[Dai09a] Wei Dai. Towards a new decision theory. https://www.lesswrong.com/posts/de3xjFaACCAk6imzv/towards-a-new-decision-theory, 2009. Published on LessWrong, August 13, 2009.

[Dai09b] Wei Dai. Why (and why not) Bayesian updating? https://www.lesswrong.com/posts/W6nXfmKTrgaiaLSRg/why-and-why-not-bayesian-updating, 2009. Published on LessWrong, November 16, 2009.

[Dai10] Wei Dai. Explicit optimization of global strategy (fixing a bug in UDT1). https://www.lesswrong.com/posts/g8xh9R7RaNitKtkaa/explicit-optimization-of-global-strategy-fixing-a-bug-in, 2010. Published on LessWrong, February 18, 2010.

[DG19] Abram Demski and Scott Garrabrant. Embedded agency. *arXiv preprint arXiv:1902.09469*, 2019.

[GBTC+16] Scott Garrabrant, Tsvi Benson-Tilsen, Andrew Critch, Nate Soares, and Jessica Taylor. Logical induction. *arXiv preprint arXiv:1609.03543*, 2016.

[MSD23] A. A. Martín Soto and Abram Demski. Logically updateless decision-making. In *PIBBSS Symposium*, 2023.

[MV90] LI Ming and Paul MB Vitányi. Kolmogorov complexity and its applications. In *Algorithms and complexity*, pages 187–254. Elsevier, 1990.

[Nes09] Vladimir Nesov. Counterfactual mugging. https://www.lesswrong.com/posts/mg6jDEuQEjBGtibX7/counterfactual-mugging, 2009. Published on LessWrong, March 19, 2009.

[SFAY15] Nate Soares, Benja Fallenstein, Stuart Armstrong, and Eliezer Yudkowsky. Corrigibility. In *AI and Ethics*, 2015.

[Tho24] Elliott Thornley. The shutdown problem: an ai engineering puzzle for decision theorists. *Philosophical Studies*, pages 1–28, 2024.

[Vin22] Susan Vineberg. Dutch Book Arguments. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2022 edition, 2022.

[YH13] Eliezer Yudkowsky and Marcello Herreshoff. Tiling agents for self-modifying ai, and the löbian obstacle. *Early Draft MIRI*, 2013.

[Yud10] Eliezer Yudkowsky. Timeless decision theory. *The Singularity Institute, San Francisco*, 2010.

Appendix A: Variation on Theorem 3

Assumption 8 (Faith in Argmax)**.** *For all observations o:*

$$E_p\big(u|\phi \wedge \pi^*(\ulcorner o \urcorner) = \arg\max_{a\in\mathcal{A}} E(u|\phi \wedge \pi^*(\ulcorner o \urcorner) = \llcorner a \lrcorner)\big) \geq \max_{a\in\mathcal{A}} E_p(u|\phi \wedge \pi^*(\ulcorner o \urcorner) = \ulcorner a \urcorner)$$

Here, $\phi = \pi^*(o') = a'$ for some $o' \neq o$. This assumption says that the agent expects argmax to achieve (at least) the maximal value (not only for the basic case of selecting a single action, but also in the presence of further information ϕ).

Assumption 9 (Naive Action Coordination)**.** *For any observations $o_1 \neq o_2$ and action $a_1 \notin \mathcal{A}^m$:*

$$p\big(\arg\max_{a\in\mathcal{A}} E_p(u|\pi^*(o_1) = \llcorner a \lrcorner \wedge \pi^*(o_2) = a_1) = \arg\max_{a\in\mathcal{A}} E_p(u|\pi^*(o_1) = \llcorner a \lrcorner)\big) = 1$$

What this assumption says is that the agent doesn't expect knowledge of a different policy-point (o_2, a_1) to change the optimal decision with respect to o_1.

Theorem 4. *Assuming Fine-Grained Fairness, Limited Self-Modification, Faith in Argmax, Naive Action Coordination, and Knowledge of Decision Procedure, UDT 1.0 does not strictly prefer any self-modifying action.*

Proof. Suppose for contradiction that some self-modification $a_m \in \mathcal{A}^m$ is strictly preferred:

$$E_p(u|\pi^*(o) = a_m) > E_p(u|\pi^*(o) = a) \text{ for all } a \in \mathcal{A}^{-m}$$

By Fine-Grained Fairness, the expectation of the self-modifying action is equal to the expectation of the corresponding non-self-modifying action $\hat{a_m}$, when conditioned on knowledge that the forced action would be taken anyway:

$$E_p(u|\pi^*(o) = a_m) = E_p\big(u|\pi^*(o) = \ulcorner \hat{a_m} \urcorner \wedge \pi^*(\ulcorner ob(a_m) \urcorner) = \ulcorner ac(a_m) \urcorner\big)$$

By definition of max, this expectation is at least as great as the expectation of the best action which could be substituted for $ac(a_m)$:

$$\leq \max_{a\in\mathcal{A}} E_p(u|\pi^*(o) = \ulcorner \hat{a_m} \urcorner \wedge \pi^*(\ulcorner ob(a_m) \urcorner) = \ulcorner a \urcorner)$$

By Faith in Argmax, this expectation is at least as great as the expected utility conditioning on the abstract statement that the best action will be chosen, rather than conditioning on the concrete best action:

$$\leq E_p\Big(u|\pi^*(o) = \ulcorner \hat{a_m} \urcorner \wedge \pi^*\big(\ulcorner ob(a_m) \urcorner\big) = \arg\max_{a\in\mathcal{A}} E_p\big(u|\pi^*(o) = \hat{a_m} \wedge \pi^*\big(\ulcorner ob(a_m) \urcorner\big) = \llcorner a \lrcorner\big)\Big)$$

By Naive Action Coordination, we can drop one of the conditions inside the argmax:

$$= E_p\Big(u|\pi^*(o) = \ulcorner \hat{a_m} \urcorner \wedge \pi^*\big(\ulcorner ob(a_m) \urcorner\big) = \arg\max_{a\in\mathcal{A}} E_p\big(u|\pi^*(\ulcorner ob(a_m) \urcorner) = \llcorner a \lrcorner\big)\Big)$$

By Knowledge of Decision Procedure, we can drop the argmax condition entirely:

$$= E_p(u|\pi^*(o) = \ulcorner \hat{a_m} \urcorner)$$

Putting it all together, the self-modifying action is just as good as its non-self-modifying version:

$$E_p(u|\pi^*(o) = a_m) \leq E_p(u|\pi^*(o) = \ulcorner \hat{a_m} \urcorner)$$

This contradicts the initial assumption. □

Appendix B: Bayesian Logical Induction

Eisenstat's approach starts with a propositionally coherent logical inductor $\mathbb{Q} = \mathbb{Q}_1, \mathbb{Q}_2, ...$ and transforms it into a Bayesian Logical Inductor (BLI) which we will call $\mathbb{P} = \mathbb{P}_1, \mathbb{P}_2, ...$

To be clear: a propositionally coherent logical inductor is one where each price assignment $\mathbb{Q}_n$ obeys the following two constraints (which amount to the Kolmogorov axioms with only finite additivity rather than countable additivity):

- $\mathbb{Q}_n(\phi) = 1$ if ϕ is a propositional tautology (eg, $\phi = \psi \vee \neg\psi$).
- $\mathbb{Q}_n(\phi \vee \psi) = \mathbb{Q}_n(\phi) + \mathbb{Q}_n(\psi)$ if $\phi \wedge \psi$ is a propositional contradiction (eg, $\phi = \neg\psi$).

Note that this implies other intuitively expected probabilistic relationships, such as:

- $\mathbb{Q}_n(\phi) = 0$ if ϕ is a propositional contradiction (eg, $\phi = \psi \wedge \neg\psi$).
- $\mathbb{Q}_n(\phi) + \mathbb{Q}_n(\neg\phi) = 1$

Eisenstat requires propositional coherence so that the logical inductor already "behaves like a normal probability distribution". This means we don't have to change it as much to get it fully into the normal Bayesian framework.

For a given $\mathbb{Q}_n$, we can distinguish "small" propositions which the market assigns real prices to (if the logical inductor was constructed by the Logical Induction Algorithm, this is the set of sentences bought or sold by any trader). We can call the rest of the sentences "large". The only real constraint on the way we define small and large is that if we assign new prices to large sentences (across all the $\mathbb{Q}_n$) in any way we like, it does not jeopardize the Logical Induction Criterion.[13]

Eisenstat's approach is to modify the prices of large sentences to represent changes in belief as a Bayesian update. Specifically, choose some notation (in the language $\mathbb{Q}$ has beliefs over) for writing out a full belief state. We will abbreviate this notation as Q, so that $\mathbb{Q}_n = Q$ will be the assertion that on day n, the logical inductor's price list is Q. Any sentence containing Q must be "large" with respect to a time which Q could describe. We will write $Q[\phi]$ to indicate the price Q assigns to ϕ. We will use subscripts like Q_1, Q_2 to distinguish different written-out belief states; notice that this use of subscripts differs from the interpretation of $\mathbb{Q}_1$ vs $\mathbb{Q}_2$, which is about time.

To keep the space of possibilities finite at every time, we can use some discretization function D_n which rounds a finite number of possible price-lists for day n (maintaining propositional consistency). It is necessary for this approximation to improve rapidly as a function of n in order to avoid violating the Logical Inductor Criterion (but the present write-up will not go into such details). I will use $\mathcal{D}_n$ to represent the set of possible price-lists to round to on day n.

Eisenstat defines $\mathbb{P}$ from $\mathbb{Q}$ via the following constraints:

1. $\mathbb{P}_n(\phi) := D_n\big(\mathbb{Q}_n(\phi)\big)$ for ϕ which are small on day n.
2. $\mathbb{P}_n(\phi|\mathbb{Q}_m = Q) := Q[\phi]$ when $m > n$
3. $\mathbb{P}_n(\phi|\mathbb{Q}_m = Q_1 \wedge \mathbb{Q}_o = Q_2) = Q_1[\phi]$ in the case that $m > o$; in the case that $o > m$ we instead have $= Q_2[\phi]$. Similarly for longer conjunctions.
4. Finally, the beliefs $\mathbb{P}_{n-1}(\mathbb{Q}_n = Q)$ are required to balance the equation:

$$\mathbb{P}_{n-1}(\phi) = \sum_{Q \in \mathcal{D}_n} \mathbb{P}_{n-1}(\mathbb{Q}_n = Q) Q[\phi]$$

A final detail of the construction: since $\mathbb{P}_n$ learns $\mathbb{Q}_n = Q$, we also need to constrain the deductive process of $\mathbb{Q}_n$ to learn this sentence – it's too big to make a difference at time n, but it will eventually matter. Like propositional consistency, this is a constraint on which Logical Inductors can be converted into Bayesian Logical Inductors.

$\mathbb{P}$ is now a logical inductor, since it agrees with $\mathbb{Q}$ on small prices; furthermore, we can see $\mathbb{P}$'s updates from one state $\mathbb{P}_n$ to the next $\mathbb{P}_{n+1}$ as a Bayesian update, since

$$\mathbb{P}_{n+1}(\phi) = \mathbb{P}_n(\phi|\mathbb{Q}_{n+1} = Q)$$

where Q notates the actual prices of $\mathbb{Q}_{n+1}$ (as rounded by D_{n+1}).

Notice that combining Eisenstat's constraints implies a version of the no-net-expected-update rule of probability [GBTC+16]:

$$\mathbb{P}_{n-1}(\phi) = \sum_{Q} \mathbb{P}_{n-1}(\mathbb{Q}_n = Q) \mathbb{P}_{n-1}(\phi|\mathbb{Q}_n = Q)$$

It is certainly possible to set beliefs that meet Eisenstat's constraints; for example, $\mathbb{P}_{n-1}(\mathbb{Q}_n = Q) = 1$ when Q accurately describes the beliefs of $\mathbb{Q}_{n-1}$ and 0 otherwise. However, this particular solution is unsatisfying, since we want to imagine that $\mathbb{P}$ is regularly updating on information about $\mathbb{Q}$'s prices. The proposed solution has $\mathbb{P}$'s large beliefs of the form $\mathbb{P}_{n-1}(\mathbb{Q}_n = Q)$ expecting that

[13] Meaning: the number of small sentences should grow faster than any polynomial (eg, grow exponentially).

prices will stay exactly the same as they are, so that any change in prices would be an update on a probability zero event. This is made worse by the fact that $\mathbb{P}$'s small beliefs probably *do not* expect the prices to stay exactly the same, at least not with 100% confidence.[14] We can see that there is still some sort of lack of coherence between small and large beliefs at play.[15]

We were also interested in a version which avoids the discretization D_n by replacing the sum in constraint 4 with an integral. We propose the following:

$$\mathbb{P}_{n-1}(\phi) = \lim_{\epsilon \to 0} \int_{Q \in [0,1]^{N_n}} \mathbb{P}_{n-1}(\mathrm{d}_{\mathrm{JS}}(Q, \widehat{Q}_n) \leq \epsilon) \cdot \mathbb{P}_{n-1}(\phi | \mathrm{d}_{\mathrm{JS}}(Q, \widehat{Q}_n) \leq \epsilon),$$

where d_{JS} is the Jensen-Shannon distance, given by the average of the KL divergences from the two distributions with respect to their midpoint, and $\widehat{Q}_n$ is a symbol for $\mathbb{Q}$'s true day-n price list.

The first factor corresponds to the first factor of the sum form; that is, $\mathbb{P}_{n-1}(\mathrm{d}_{\mathrm{JS}}(Q, \widehat{Q}_n) \leq \epsilon)$ is $\mathbb{P}$'s credence on day $n-1$ that $\mathbb{Q}$'s price list tomorrow (i.e., day n) will be within Jensen-Shannon distance ϵ of some chosen distribution Q.

The second factor corresponds to the second factor of the sum form; that is, $\mathbb{P}_{n-1}(\phi | \mathrm{d}_{\mathrm{JS}}(Q, \widehat{Q}_n) \leq \epsilon)$ is $\mathbb{P}$'s credence on day $n-1$ on some proposition ϕ, conditional on the guessed distribution Q being within the radius-ϵ sphere from the first term.

[14] Actually, logical inductors are allowed to make such mistakes finitely many times; a logical inductor can update away from 100% confidence. What's strictly true is that logical inductors cannot contain an efficiently computable sequence of such mistakes. Logical Inductors do in fact change their prices, so Logical Inductors must learn this fact.

[15] In a normal Logical Induction context, this would be fine; large beliefs don't need to be coherent at all, not even in the weak asymptotic ways that small beliefs are required to be coherent. However, Bayesian Logical Induction can be seen as an attempt to articulate meaningful coherence constraints for large sentences. Philosophically, then, Eisenstat's proposal seems to be incomplete in some sense. It articulates some important rationality constraints for large sentences, but it leaves room for more such important constraints to be articulated.