Reward-Punishment Symmetric Universal Intelligence

Samuel Allen Alexander[1(✉)] and Marcus Hutter[2]

[1] The U.S. Securities and Exchange Commission, New York City, USA
[2] DeepMind & AMU, London, UK
https://philpeople.org/profiles/samuel-alexander/publications,
http://www.hutter1.net/

Abstract. Can an agent's intelligence level be negative? We extend the Legg-Hutter agent-environment framework to include punishments and argue for an affirmative answer to that question. We show that if the background encodings and Universal Turing Machine (UTM) admit certain Kolmogorov complexity symmetries, then the resulting Legg-Hutter intelligence measure is symmetric about the origin. In particular, this implies reward-ignoring agents have Legg-Hutter intelligence 0 according to such UTMs.

Keywords: Universal intelligence · Intelligence measures · Reinforcement learning

1 Introduction

In their paper [11], Legg and Hutter write:

> "As our goal is to produce a definition of intelligence that is as broad and encompassing as possible, the space of environments used in our definition should be as large as possible."

So motivated, we investigate what would happen if we extended the universe of environments to include environments with rewards from $\mathbb{Q} \cap [-1, 1]$ instead of just from $\mathbb{Q} \cap [0, 1]$ as in Legg and Hutter's paper. In other words, we investigate what would happen if environments are not only allowed to reward agents but also to punish agents (a punishment being a negative reward).

We discovered that when negative rewards are allowed, this introduces a certain algebraic structure into the agent-environment framework. The main objection we anticipate to our extended framework is that it implies the negative intelligence of certain agents[1]. We would argue that this makes perfect sense

[1] Thus, this paper falls under the broader program of advocating for intelligence measures having different ranges than the nonnegative reals. Alexander has advocated more extreme extensions of the range of intelligence measures [1,2]; by contrast, here we merely question the assumption that intelligence never be negative, leaving aside the question of whether intelligence should be real-valued.

B. Goertzel et al. (Eds.): AGI 2021, LNAI 13154, pp. 1–10, 2022.
https://doi.org/10.1007/978-3-030-93758-4_1

when environments are capable of punishing agents: if the intelligence level of a reinforcement learning agent is a measure of its ability to extract large rewards on average across many environments, then an agent who instead extracts large punishments should have a negative intelligence level.

This paper advances the practical pursuit of AGI by suggesting (in Sect. 4) certain symmetry constraints which would narrow down the space of background UTMs, thereby refining at least one approach to intelligence measurement. In particular these constraints are one answer to Leike and Hutter, who asked: "But what are other desirable properties of a UTM?" [13].

The structure of the paper is as follows:

- In Sect. 2, we give preliminary definitions.
- In Sect. 3, we introduce what we call the dual of an agent and of an environment, and prove some algebraic theorems about these.
- In Sect. 4, we show the existence of UTMs yielding Kolmogorov complexities with certain symmetries, and show that the resulting Legg-Hutter intelligence measures are symmetric too.
- In Sect. 5 we consider the absolute value of Legg-Hutter intelligence as an alternative intelligence measure.
- In Sect. 6, we summarize and make concluding remarks, including remarks about how these ideas might be applied to certain other intelligence measures.

2 Preliminaries

In defining agent and environment below, we attempt to follow Legg and Hutter [11] as closely as possible, except that we permit environments to output rewards from $\mathbb{Q} \cap [-1, 1]$ rather than just $\mathbb{Q} \cap [0, 1]$ (and, accordingly, we modify which well-behaved environments to restrict our attention to).

Throughout the paper, we implicitly fix a finite set $\mathcal{A}$ of *actions*, a finite set $\mathcal{O}$ of *observations*, and a finite set $\mathcal{R} \subseteq \mathbb{Q} \cap [-1, 1]$ of *rewards* (so each reward is a rational number between -1 and 1 inclusive), with $|\mathcal{A}| > 0$, $|\mathcal{O}| > 0$, $|\mathcal{R}| > 0$. We assume that $\mathcal{R}$ has the following property: whenever $\mathcal{R}$ contains any reward r, then $\mathcal{R}$ also contains $-r$. We assume $\mathcal{A}$, $\mathcal{O}$, and $\mathcal{R}$ are mutually disjoint (i.e., no reward is an action, no reward is an observation, and no action is an observation). By $\langle\rangle$ we mean the empty sequence.

Definition 1 *(Agents, environments, etc.).*

1. *By $(\mathcal{ORA})^*$ we mean the set of all finite sequences starting with an observation, ending with an action, and following the pattern "observation, reward, action, ...". We include $\langle\rangle$ in this set.*
2. *By $(\mathcal{ORA})^*\mathcal{OR}$ we mean the set of all sequences of the form $s \frown o \frown r$ where $s \in (\mathcal{ORA})^*$, $o \in \mathcal{O}$ and $r \in \mathcal{R}$ ($\frown$ denotes concatenation).*
3. *By an* agent*, we mean a function π with domain $(\mathcal{ORA})^*\mathcal{OR}$, which assigns to every sequence $s \in (\mathcal{ORA})^*\mathcal{OR}$ a $\mathbb{Q}$-valued probability measure, written $\pi(\bullet|s)$, on $\mathcal{A}$. For every such s and every $a \in \mathcal{A}$, we write $\pi(a|s)$ for $(\pi(\bullet|s))(a)$. Intuitively, $\pi(a|s)$ is the probability that agent π will take action a in response to history s.*

4. *By an* environment*, we mean a function μ with domain $(\mathcal{ORA})^*$, which assigns to every $s \in (\mathcal{ORA})^*$ a $\mathbb{Q}$-valued probability measure, written $\mu(\bullet|s)$, on $\mathcal{O} \times \mathcal{R}$. For every such s and every $(o, r) \in \mathcal{O} \times \mathcal{R}$, we write $\mu(o, r|s)$ for $(\mu(\bullet|s))(o, r)$. Intuitively, $\mu(o, r|s)$ is the probability that environment μ will issue observation o and reward r to the agent in response to history s.*
5. *If π is an agent, μ is an environment, and $n \in \mathbb{N}$, we write $V^\pi_{\mu,n}$ for the expected value of the sum of the rewards which would occur in the sequence $(o_0, r_0, a_0, \ldots, o_n, r_n, a_n)$ randomly generated as follows:*
   - *(a) $(o_0, r_0) \in \mathcal{O} \times \mathcal{R}$ is chosen randomly based on the probability measure $\mu(\bullet|\langle\rangle)$.*
   - *(b) $a_0 \in \mathcal{A}$ is chosen randomly based on the probability measure $\pi(\bullet|o_0, r_0)$.*
   - *(c) For each $i > 0$, $(o_i, r_i) \in \mathcal{O} \times \mathcal{R}$ is chosen randomly based on the probability measure $\mu(\bullet|o_0, r_0, a_0, \ldots, o_{i-1}, r_{i-1}, a_{i-1})$.*
   - *(d) For each $i > 0$, $a_i \in \mathcal{A}$ is chosen randomly based on the probability measure $\pi(\bullet|o_0, r_0, a_0, \ldots, o_{i-1}, r_{i-1}, a_{i-1}, o_i, r_i)$.*
6. *If π is an agent and μ is an environment, let $V^\pi_\mu = \lim_{n\to\infty} V^\pi_{\mu,n}$. Intuitively, V^π_μ is the expected total reward which π would extract from μ.*

Note that it is possible for V^π_μ to be undefined. For example, if μ is an environment which always issues reward $(-1)^n$ in response to the agent's nth action, then V^π_μ is undefined for every agent π. This would not be the case if rewards were required to be ≥ 0, so this is one way in which allowing punishments complicates the resulting theory.

Definition 2. *An environment μ is* well-behaved *if μ is computable and the following condition holds: for every agent π, V^π_μ exists and $-1 \leq V^\pi_\mu \leq 1$.*

Note that reward-space $[0, 1]$ can be transformed into punishment-space $[-1, 0]$ either via $r \mapsto -r$ or via $r \mapsto r - 1$. An advantage of $r \mapsto -r$ is that it preserves well-behavedness of environments (we prove this below in Corollary 7)[2].

3 Dual Agents and Dual Environments

In the Introduction, we promised that by allowing environments to punish agents, we would reveal algebraic structure not otherwise present. The key to this additional structure is the following definition.

[2] It is worth mentioning another difference between these two transforms. The hypothetical agent AI_μ with perfect knowledge of the environment's reward distribution would not change its behavior in response to $r \mapsto r - 1$ (nor indeed in response to any positive linear scaling $r \mapsto ar + b$, $a > 0$), but it would generally change its behavior in response to $r \mapsto -r$. Interestingly, this behavior invariance with respect to $r \mapsto r - 1$ would not hold if AI_μ were capable of "suicide" (deliberately ending the environmental interaction): one should never quit a slot machine that always pays between 0 and 1 dollars, but one should immediately quit a slot machine that always pays between -1 and 0 dollars. The agent AIXI also changes behavior in response to $r \mapsto r - 1$, and it was recently argued that this can be interpreted in terms of suicide/death: AIXI models its environment using a mixture distribution over a countable class of semimeasures, and AIXI's behavior can be interpreted as treating the complement of the domain of each semimeasure as death, see [14].

Definition 3 *(Dual Agents and Dual Environments).*

1. *For each sequence s, let $\overline{s}$ be the sequence obtained by replacing every reward r in s by $-r$.*
2. *Suppose π is an agent. We define a new agent $\overline{\pi}$, the* dual *of π, as follows: for each $s \in (\mathcal{ORA})^*\mathcal{OR}$, for each action $a \in \mathcal{A}$,*

$$\overline{\pi}(a|s) = \pi(a|\overline{s}).$$

3. *Suppose μ is an environment. We define a new environment $\overline{\mu}$, the* dual *of μ, as follows: for each $s \in (\mathcal{ORA})^*$, for each observation $o \in \mathcal{O}$ and reward $r \in \mathcal{R}$,*

$$\overline{\mu}(o, r|s) = \mu(o, -r|\overline{s}).$$

Lemma 4 *(Double Negation). If x is a sequence, agent, or environment, then $\overline{\overline{x}} = x$.*

Proof. Follows from the fact that for every real number r, $--r = r$. □

Theorem 5. *Suppose μ is an environment and π is an agent. Then*

$$V^{\overline{\pi}}_{\overline{\mu}} = -V^{\pi}_{\mu}$$

(and the left-hand side is defined if and only if the right-hand side is defined).

Proof. By Definition 1 part 6, it suffices to show that for each $n \in \mathbb{N}$, $V^{\overline{\pi}}_{\overline{\mu},n} = -V^{\pi}_{\mu,n}$. For that, it suffices to show that for every $s \in ((\mathcal{ORA})^*) \cup ((\mathcal{ORA})^*\mathcal{OR})$, the probability X of generating s using π and μ (as in Definition 1 part 5) equals the probability X' of generating $\overline{s}$ using $\overline{\pi}$ and $\overline{\mu}$. We will show this by induction on the length of s.

Case 1: s is empty. Then $X = X' = 1$.

Case 2: s terminates with an action. Then $s = t \frown a$ for some $t \in (\mathcal{ORA})^*\mathcal{OR}$. Let Y (resp. Y') be the probability of generating t (resp. $\overline{t}$) using π and μ (resp. $\overline{\pi}$ and $\overline{\mu}$). We reason: $X = \pi(a|t)Y = \pi(a|\overline{\overline{t}})Y = \overline{\pi}(a|\overline{t})Y$ by definition of $\overline{\pi}$. By induction, $Y = Y'$, so $X = \overline{\pi}(a|\overline{t})Y'$, which by definition is X'.

Case 3: s terminates with a reward. Similar to Case 2. □

Corollary 6. *For every agent π and environment μ,*

$$V^{\pi}_{\overline{\mu}} = -V^{\overline{\pi}}_{\mu}$$

(and the left-hand side is defined if and only if the right-hand side is defined).

Proof. If neither side is defined, then there is nothing to prove. Assume the left-hand side is defined. Then

$$\begin{aligned} V^{\pi}_{\overline{\mu}} &= V^{\overline{\overline{\pi}}}_{\overline{\mu}} && \text{(Lemma 4)} \\ &= -V^{\overline{\pi}}_{\mu}, && \text{(Theorem 5)} \end{aligned}$$

as desired. A similar argument holds if we assume the right-hand side is defined. □

Corollary 7. *For every environment μ, μ is well-behaved if and only if $\overline{\mu}$ is well-behaved.*

Proof. We prove the $\Rightarrow$ direction, the other is similar. Since μ is well-behaved, μ is computable, so clearly $\overline{\mu}$ is computable. Let π be any agent. Since μ is well-behaved, $V_{\mu}^{\overline{\pi}}$ is defined and $-1 \leq V_{\mu}^{\overline{\pi}} \leq 1$. By Corollary 6, $V_{\overline{\mu}}^{\pi} = -V_{\mu}^{\overline{\pi}}$ is defined, implying $-1 \leq V_{\overline{\mu}}^{\pi} \leq 1$. By arbitrariness of π, this shows $\overline{\mu}$ is well-behaved. $\square$

4 Symmetric Intelligence

Agent $\overline{\pi}$ acts as agent π would act if π confused punishments with rewards and rewards with punishments. Whatever ingenuity π applies to maximize rewards, $\overline{\pi}$ applies that same ingenuity to maximize punishments. Thus, if Υ measures intelligence as performance averaged in some way[3], it seems natural that we might expect the following property to hold $(*)$: that whenever $\Upsilon(\pi) \neq 0$, then $\Upsilon(\pi) \neq \Upsilon(\overline{\pi})$. Indeed, one could argue it would be strange to hold that π manages to extract (say) positive rewards on average, and at the same time hold that $\overline{\pi}$ (which uses π to seek punishments) extracts the exact same positive rewards on average. To be clear, we do not declare $(*)$ is an absolute law, we merely opine that $(*)$ seems reasonable and natural. Now, assuming $(*)$, we can offer an informal argument for a stronger-looking symmetry property $(**)$: that $\Upsilon(\pi) = -\Upsilon(\overline{\pi})$ for all π. The informal argument is as follows. Let π be any agent. Imagine a new agent ρ which, at the start of every environmental interaction, flips a coin and commits to act as π for that whole interaction if the coin lands heads, or to act as $\overline{\pi}$ for the whole interaction if the coin lands tails. Probabilistic intuition suggests $\Upsilon(\rho) = \frac{1}{2}(\Upsilon(\pi) + \Upsilon(\overline{\pi}))$, so if $\Upsilon(\rho) = 0$ then $\Upsilon(\pi) = -\Upsilon(\overline{\pi})$. But maybe the reader doubts $\Upsilon(\rho) = 0$. In that case, define ρ' in the same way except swap "heads" and "tails". It seems there is no way to meaningfully distinguish ρ from ρ', so it seems we ought to have $\Upsilon(\rho) = \Upsilon(\rho')$. But to swap "heads" and "tails" is the same as to swap "π" and "$\overline{\pi}$". Thus $\rho' = \overline{\rho}$. Thus $\Upsilon(\rho) \neq 0$ would contradict $(*)$. In conclusion, while we do not declare it an absolute law, we do consider $(**)$ natural and reasonable, at least if Υ measures intelligence as performance averaged in some way. In this section, we will show that Legg and Hutter's universal intelligence measure satisfies $(**)$, provided a background UTM and encoding are suitably chosen.

We write 2^* for the set of finite binary strings. We write $f :\subseteq A \to B$ to indicate that f has codomain B and that f's domain is some subset of A.

Definition 8 *(Prefix-free universal Turing machines).*

1. *A partial computable function $f :\subseteq 2^* \to 2^*$ is* prefix-free *if the following requirement holds: $\forall p, p' \in 2^*$, if p is a strict initial segment of p', then $f(p)$ and $f(p')$ are not both defined.*

[3] Note that measuring intelligence as averaged performance might conflict with certain everyday uses of the word "intelligent", see Sect. 5.

2. *A* prefix-free universal Turing machine *(or* PFUTM*) is a prefix-free partial computable function $U :\subseteq 2^* \to 2^*$ such that the following condition holds. For every prefix-free partial computable function $f :\subseteq 2^* \to 2^*$, $\exists y \in 2^*$ such that $\forall x \in 2^*$, $f(x) = U(y \frown x)$. In this case, we say y is a* computer program for f in programming language U*.*

Environments do not have domain $\subseteq 2^*$, and they do not have codomain 2^*. Rather, their domain and codomain are $(\mathcal{ORA})^*$ and the set of $\mathbb{Q}$-valued probability measures on $\mathcal{O} \times \mathcal{R}$, respectively. Thus, in order to talk about their Kolmogorov complexities, one must encode said inputs and outputs. This low-level detail is usually implicit, but we will need (in Theorem 11) to distinguish between different kinds of encodings, so we must make the details explicit.

Definition 9. *By an* RL-encoding *we mean a computable function $\sqcap : (\mathcal{ORA})^* \cup M \to 2^*$ (where M is the set of $\mathbb{Q}$-valued probability-measures on $\mathcal{O} \times \mathcal{R}$) such that for all $x, y \in (\mathcal{ORA})^* \cup M$ (with $x \neq y$), $\sqcap(x)$ is not an initial segment of $\sqcap(y)$. We say $\sqcap$ is* suffix-free *if for all $x, y \in (\mathcal{ORA})^* \cup M$ (with $x \neq y$), $\sqcap(x)$ is not a terminal segment of $\sqcap(y)$. We write $\ulcorner x \urcorner$ for $\sqcap(x)$.*

Note that in Definition 9, it makes sense to encode M because $\mathcal{O}$ and $\mathcal{R}$ are finite (Sect. 2). Notice that suffix-freeness is, in some sense, the reverse of prefix-freeness. The existence of encodings that are simultaneously prefix-free and suffix-free is well-known. For example, elements of the range of $\sqcap$ could be composed of 8-bit blocks (bytes), such that every element of the range of $\sqcap$ begins and ends with the ASCII closed-bracket characters [and], respectively, and such that these closed-brackets do not appear anywhere in the middle.

Definition 10 *(Kolmogorov Complexity). Suppose U is a PFUTM and $\sqcap$ is an RL-encoding.*

1. *For each computable environment μ, the* Kolmogorov complexity of μ given by $U, \sqcap$*, written $K^{\sqcap}_U(\mu)$, is the smallest $n \in \mathbb{N}$ such that there is some computer program of length n, in programming language U, for some function $f :\subseteq 2^* \to 2^*$ such that for all $s \in (\mathcal{ORA})^*$, $f(\ulcorner s \urcorner) = \ulcorner \mu(\bullet|s) \urcorner$ (note this makes sense since the domain of $\sqcap$ in Definition 9 is $(\mathcal{ORA})^* \cup M$).*
2. *We say U is* symmetric in its $\sqcap$-encoded-environment cross-section *(or simply that U is* $\sqcap$-symmetric*) if $K^{\sqcap}_U(\mu) = K^{\sqcap}_U(\overline{\mu})$ for every computable environment μ.*

Theorem 11. *For every suffix-free RL-encoding $\sqcap$, there exists a $\sqcap$-symmetric PFUTM.*

Proof. Let U_0 be a PFUTM, we will modify U_0 to obtain a $\sqcap$-symmetric PFUTM. For readability's sake, write POS for 0 and NEG for 1. Thinking of U_0 as a programming language, we define a new programming language U as follows. Every program in U must begin with one of the keywords POS or NEG. Outputs of U are defined as follows.

– $U(\text{POS} \frown x) = U_0(x)$.

– To compute $U(\mathrm{NEG} \frown x)$, find $s \in (\mathcal{ORA})^*$ such that $x = y \frown \ulcorner s \urcorner$ for some y (if no such s exists, diverge). Note that s is unique by suffix-freeness of $\sqcap$. If $U_0(y \frown \ulcorner \overline{s} \urcorner) = \ulcorner m \urcorner$ for some $\mathbb{Q}$-valued probability-measure m on $\mathcal{O} \times \mathcal{R}$, then let $U(\mathrm{NEG} \frown x) = \ulcorner \overline{m} \urcorner$ where $\overline{m}(o, r) = m(o, -r)$. Otherwise, diverge.
  - Informally: If x appears to be an instruction to plug s into computer program y to get a probability measure $\mu(\bullet|s)$, then instead plug $\overline{s}$ into y and flip the resulting probability measure so that the output ends up being the flipped version of $\mu(\bullet|\overline{s})$, i.e., $\overline{\mu}(\bullet|s)$.

By construction, whenever $\mathrm{POS} \frown y$ is a U-computer program for a function f satisfying $f(\ulcorner s \urcorner) = \ulcorner \mu(\bullet|s) \urcorner$, $\mathrm{NEG} \frown y$ is an equal-length U-computer program for a function g satisfying $g(\ulcorner s \urcorner) = \ulcorner \overline{\mu}(\bullet|s) \urcorner$, and vice versa. It follows that U is $\sqcap$-symmetric. □

The proof of Theorem 11 proves more than required: any PFUTM can be modified to make a $\sqcap$-symmetric PFUTM if $\sqcap$ is suffix-free. In some sense, the construction in the proof of Theorem 11 works by eliminating bias: reinforcement learning itself is implicitly biased in its convention that rewards be positive and punishments negative. We can imagine a pessimistic parallel universe where RL instead follows the opposite convention, and the RL in that parallel universe is no less valid than the RL in our own. To be unbiased in this sense, a computer program defining an environment should specify which of the two RL conventions it is operating under (hence the POS and NEG keywords). This trick of using an initial bit to indicate reward-reversal was previously used in [12].

Definition 12. *Let W be the set of all well-behaved environments. Let $\overline{W} = \{\overline{\mu} : \mu \in W\}$.*

Definition 13. *For every PFUTM U, RL-encoding $\sqcap$, and agent π, the* Legg-Hutter universal intelligence of π given by $U, \sqcap$, *written $\Upsilon_U^{\sqcap}(\pi)$, is*

$$\Upsilon_U^{\sqcap}(\pi) = \sum_{\mu \in W} 2^{-K_U^{\sqcap}(\mu)} V_\mu^\pi.$$

The sum defining $\Upsilon_U^{\sqcap}(\pi)$ is absolutely convergent by comparison with the summands defining Chaitin's constant (hence the prefix-free UTM requirement). Thus a well-known theorem from calculus says the sum does not depend on which order the $\mu \in W$ are enumerated.

Legg-Hutter intelligence has been accused of being subjective because of its UTM-sensitivity [7,9,13]. More optimistically, UTM-sensitivity could be considered a feature, reflecting the existence of many kinds of intelligence. It could be used to measure intelligence in various contexts, by choosing UTMs appropriately. One could even use it to measure, say, chess intelligence, by choosing a UTM where chess-related environments are easiest to program.

Theorem 14 *(Symmetry about the origin). For every RL-encoding $\sqcap$, every $\sqcap$-symmetric PFUTM U, and every agent π,*

$$\Upsilon_U^{\sqcap}(\overline{\pi}) = -\Upsilon_U^{\sqcap}(\pi).$$

Proof. By Corollary 6,

$$\Upsilon_U^{\sqcap}(\overline{\pi}) = \sum_{\mu\in W} 2^{-K_U^{\sqcap}(\mu)} V_\mu^{\overline{\pi}} = -\sum_{\mu\in W} 2^{-K_U^{\sqcap}(\mu)} V_{\overline{\mu}}^{\pi}.$$

By $\sqcap$-symmetry, we can rewrite this as

$$-\sum_{\mu\in W} 2^{-K_U^{\sqcap}(\overline{\mu})} V_{\overline{\mu}}^{\pi} = -\sum_{\mu\in \overline{W}} 2^{-K_U^{\sqcap}(\mu)} V_\mu^{\pi}.$$

By Corollary 7, $W = \overline{W}$, so this expression equals $-\sum_{\mu\in W} 2^{-K_U^{\sqcap}(\mu)} V_\mu^\pi$, which is $-\Upsilon_U^{\sqcap}(\pi)$ by Definition 13. □

The following corollary addresses another obvious desideratum. This corollary is foreshadowed in [12].

Corollary 15. *Let $\sqcap$ be an RL-encoding, let U be a $\sqcap$-symmetric PFUTM and suppose π is an agent which ignores rewards (by which we mean that $\pi(\bullet|s)$ does not depend on the rewards in s). Then $\Upsilon_U^{\sqcap}(\pi) = 0$.*

Proof. The hypothesis implies $\pi = \overline{\pi}$, so by Theorem 14, $\Upsilon_U^{\sqcap}(\pi) = -\Upsilon_U^{\sqcap}(\pi)$. □

Corollary 15 illustrates why it is appropriate, for purposes of Legg-Hutter universal intelligence, to choose a $\sqcap$-symmetric PFUTM[4]. Consider an agent π_a which blindly repeats a fixed action $a \in \mathcal{A}$. For any particular environment μ, where π_a earns total reward r by blind luck, that total reward should be cancelled by $\overline{\mu}$, where that blind luck becomes blind misfortune and π_a earns total reward $-r$ (Corollary 6). If $K_U^{\sqcap}(\mu) \neq K_U^{\sqcap}(\overline{\mu})$, the different weights $2^{-K_U^{\sqcap}(\mu)} \neq 2^{-K_U^{\sqcap}(\overline{\mu})}$ would prevent cancellation.

We conclude this section with an exercise, suggesting how the techniques of this paper can be used to obtain other structural results.

Exercise 16 *(Permutations).*

1. *For each permutation $P : \mathcal{A} \to \mathcal{A}$ of the action-space, for each sequence s, let Ps be the result of applying P to all the actions in s. For each agent π, let $P\pi$ be the agent defined by $P\pi(a|s) = \pi(Pa|Ps)$. For each environment μ, let $P\mu$ be the environment defined by $P\mu(o, r|s) = \mu(o, r|Ps)$. Show that in general $V_\mu^\pi = V_{P^{-1}\mu}^{P\pi}$ and $V_\mu^{P\pi} = V_{P\mu}^{\pi}$.*
2. *Say PFUTM U is* $\sqcap$-permutable *if $K_U^{\sqcap}(\mu) = K_U^{\sqcap}(P\mu)$ for every computable environment μ and permutation $P : \mathcal{A} \to \mathcal{A}$. Show that if $\sqcap$ is suffix-free then any given PFUTM can be transformed into a $\sqcap$-permutable PFUTM.*
3. *Show that if U is a $\sqcap$-permutable PFUTM, then $\Upsilon_U^{\sqcap}(P\pi) = \Upsilon_U^{\sqcap}(\pi)$ for every agent π and permutation $P : \mathcal{A} \to \mathcal{A}$.*
4. *Modify this exercise to apply to permutations of the observation-space.*

[4] An answer to Leike and Hutter's [13] "what are other desirable [UTM properties]?".

5 Whether to Take Absolute Values

Definition 13 assigns negative intelligence to agents who consistently minimize rewards. This is based on the desire to measure performance: agents who consistently minimize rewards have poor performance. One might, however, argue that $|\Upsilon_U^{\sqcap}(\pi)|$ would be a better measure of the agent's intelligence: if mathematical functions could have desires, one might argue that when $\Upsilon_U^{\sqcap}(\pi) < 0$, we should give π the benefit of the doubt, assume that π desires punishment, and conclude π is intelligent. This would more closely align with Bostrom's orthogonality thesis [3]. In the same way, a subject who answers every question wrong in a true-false IQ test might be considered intelligent: answering every question wrong is as hard as answering every question right, and we might give the subject the benefit of the doubt and assume they meant to answer wrong[5]. Rather than take a side and declare one of $\Upsilon_U^{\sqcap}$ or $|\Upsilon_U^{\sqcap}|$ to be the better measure, we consider them to be two equally valid measures, one of which measures performance and one of which measures the agent's ability to consistently extremize rewards (whether consistently positively or consistently negatively).

If one knew that π's Legg-Hutter intelligence were negative, one could derive the same benefit from π as from $\overline{\pi}$: just flip rewards. This raises the question: given π, can one computably determine $\mathrm{sgn}(\Upsilon_U^{\sqcap}(\pi))$? Or more weakly, is there a procedure which outputs $\mathrm{sgn}(\Upsilon_U^{\sqcap}(\pi))$ when $\Upsilon_U^{\sqcap}(\pi) \neq 0$ (but, when $\Upsilon_U^{\sqcap}(\pi) = 0$, may output a wrong answer or get stuck in an infinite loop)? One can easily contrive non-$\sqcap$-symmetric PFUTMs where $\mathrm{sgn}(\Upsilon_U^{\sqcap}(\pi))$ is computable from π—in fact, without the $\sqcap$-symmetry requirement, one can arrange that $\Upsilon_U^{\sqcap}(\pi)$ is *always* positive, by arranging that $\Upsilon_U^{\sqcap}(\pi)$ is dominated by a low-K environment that blindly gives all agents $+1$ total reward. On the other hand, one can contrive a $\sqcap$-symmetric PFUTM such that $\mathrm{sgn}(\Upsilon_U^{\sqcap}(\pi))$ is not computable from π even in the weak sense[6]. We leave it an open question whether there is any $\sqcap$-symmetric PFUTM U where $\mathrm{sgn}(\Upsilon_U^{\sqcap}(\pi))$ is computable (in the strong or weak sense).

6 Conclusion

By allowing environments to punish agents, we found additional algebraic structure in the agent-environment framework. Using this, we showed that certain Kolmogorov complexity symmetries yield Legg-Hutter intelligence symmetry.

[5] To quote Socrates: "Don't you think the ignorant person would often involuntarily tell the truth when he wished to say falsehoods, if it so happened, because he didn't know; whereas you, the wise person, if you should wish to lie, would always consistently lie?" [15].

[6] Arrange that $\Upsilon_U^{\sqcap}$ is dominated by μ and $\bar{\mu}$ where μ is an environment that initially gives reward .01, then waits for the agent to input the code of a Turing machine T, then (if the agent does so), gives reward $-.51$, then gives rewards 0 while simulating T until T halts, finally giving reward 1 if T does halt. Then if $\mathrm{sgn}(\Upsilon_U^{\sqcap}(\pi))$ were computable (even in the weak sense), one could compute it for strategically-chosen agents and solve the Halting Problem.

In future work it would be interesting to explore how these symmetries manifest themselves in other Legg-Hutter-like intelligence measures [5,6,8]. The precise strategy we employ in this paper is not directly applicable to prediction-based intelligence measurement [2,4,10], but a higher-level idea still applies: an intentional mis-predictor underperforms a 0-intelligence blind guesser.

Acknowledgments. We acknowledge José Hernández-Orallo, Shane Legg, Pedro Ortega, and the reviewers for comments and feedback.

References

1. Alexander, S.A.: The Archimedean trap: why traditional reinforcement learning will probably not yield AGI. JAGI **11**(1), 70–85 (2020)
2. Alexander, S.A., Hibbard, B.: Measuring intelligence and growth rate: variations on Hibbard's intelligence measure. JAGI **12**(1), 1–25 (2021)
3. Bostrom, N.: The superintelligent will: motivation and instrumental rationality in advanced artificial agents. Minds Mach. **22**(2), 71–85 (2012)
4. Gamez, D.: Measuring intelligence in natural and artificial systems. J. Artif. Intell. Conscious. **08**(2), 285–302 (2021)
5. Gavane, V.: A measure of real-time intelligence. JAGI **4**(1), 31–48 (2013)
6. Goertzel, B.: Patterns, hypergraphs and embodied general intelligence. In: IJCNNP, IEEE (2006)
7. Hernández-Orallo, J.: C-tests revisited: back and forth with complexity. In: CAGI (2015)
8. Hernández-Orallo, J., Dowe, D.L.: Measuring universal intelligence: towards an anytime intelligence test. AI **174**(18), 1508–1539 (2010)
9. Hibbard, B.: Bias and no free lunch in formal measures of intelligence. JAGI **1**(1), 54 (2009)
10. Hibbard, B.: Measuring agent intelligence via hierarchies of environments. In: CAGI (2011)
11. Legg, S., Hutter, M.: Universal intelligence: a definition of machine intelligence. Minds Mach. **17**(4), 391–444 (2007)
12. Legg, S., Veness, J.: An approximation of the universal intelligence measure. In: Dowe, D.L. (ed.) Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence. LNCS, vol. 7070, pp. 236–249. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-44958-1_18
13. Leike, J., Hutter, M.: Bad universal priors and notions of optimality. In: Conference on Learning Theory, pp. 1244–1259. PMLR (2015)
14. Martin, J., Everitt, T., Hutter, M.: Death and suicide in universal artificial intelligence. In: CAGI (2016)
15. Plato: Lesser Hippias. In: Cooper, J.M., Hutchinson, D.S., et al. (eds.) Plato: Complete Works. Hackett Publishing, Indianapolis (1997)