On the Computability of Solomonoff Induction and AIXI

Jan Leike[1] and Marcus Hutter

Australian National University

Abstract

How could we solve the machine learning and the artificial intelligence problem if we had infinite computation? Solomonoff induction and the reinforcement learning agent AIXI are proposed answers to this question. Both are known to be incomputable. We quantify this using the arithmetical hierarchy, and prove upper and in most cases corresponding lower bounds for incomputability. Moreover, we show that AIXI is not limit computable, thus it cannot be approximated using finite computation. However there are limit computable ε-optimal approximations to AIXI. We also derive computability bounds for knowledge-seeking agents, and give a limit computable weakly asymptotically optimal reinforcement learning agent.

Keywords: Solomonoff induction, AIXI, general reinforcement learning, knowledge-seeking agents, computability, arithmetical hierarchy.

1. Introduction

Given infinite computation power, many traditional AI problems become trivial: playing chess, go, or backgammon can be solved by exhaustive expansion of the game tree. Yet other problems seem difficult still; for example, predicting the stock market, driving a car, or babysitting your nephew. How can we solve these problems in theory? A proposed answer to this question is the agent AIXI (Hutter, 2000, 2005). As a *reinforcement learning agent*, its goal is to maximize cumulative (discounted) rewards obtained from the environment (Sutton and Barto, 1998).

[1]Now at DeepMind.

The basis of AIXI is Solomonoff's theory of learning (Solomonoff, 1964, 1978; Li and Vitányi, 2008), also called *Solomonoff induction*. It arguably solves the induction problem (Rathmanner and Hutter, 2011): for data drawn from a computable measure μ, Solomonoff induction will converge to the correct belief about any hypothesis (Blackwell and Dubins, 1962; Rathmanner and Hutter, 2011). Moreover, convergence is extremely fast in the sense that Solomonoff induction will make a total of at most $E + O(\sqrt{E})$ errors when predicting the next data points, where E is the number of errors of the informed predictor that knows μ (Hutter, 2001). While learning the environment according to Solomonoff's theory, AIXI selects actions by running an expectimax-search for maximum cumulative discounted rewards. However, AIXI's trade-off between exploration and exploitation includes insufficient exploration to get rid of the prior's bias (Leike and Hutter, 2015c), which is why the universal agent AIXI does not achieve asymptotic optimality (Orseau, 2013). It is clear that AIXI can only serve as an ideal, yet recently it has inspired some impressive applications (Veness et al., 2011).

For extra exploration, we can resort to Orseau's *knowledge-seeking agents*. Instead of rewards, knowledge-seeking agents maximize entropy gain (Orseau, 2014) or expected information gain (Orseau et al., 2013). These agents are apt explorers, and asymptotically they fully learn their environment.

A reinforcement learning agent is *weakly asymptotically optimal* if the value of its policy converges to the optimal value in Cesàro mean (Lattimore and Hutter, 2011). Weak asymptotic optimality stands out because it currently is the only known nontrivial objective notion of optimality for general reinforcement learners (Lattimore and Hutter, 2011; Leike and Hutter, 2015c). Lattimore defines the agent BayesExp by grafting a knowledge-seeking component on top of AIXI and shows that BayesExp is a weakly asymptotically optimal agent in the class of all stochastically computable environments (Lattimore, 2013, Ch. 5).

Both Solomonoff induction and AIXI are known to be incomputable. But not all incomputabilities are equal. The *arithmetical hierarchy* specifies different levels of computability based on *oracle machines*: each level in the arithmetical hierarchy is computed by a Turing machine which may query a halting oracle for the respective lower level.

The purpose of models such as Solomonoff induction, AIXI, and knowledge-seeking agents is to answer the question of how to solve (reinforcement) learning *in theory*. These answers are useless if they cannot be approximated in practice, i.e., by a regular Turing machine. Therefore we posit that any ideal for a 'perfect agent' needs to be *limit computable* (Δ^0_2). The class of limit computable functions is the class of functions that admit an *anytime algorithm*.

P	$\{(x,q) \in \mathcal{X}^* \times \mathbb{Q} \mid P(x) > q\}$	$\{(x,y,q) \in \mathcal{X}^* \times \mathcal{X}^* \times \mathbb{Q} \mid P(xy \mid x) > q\}$
M	$\Sigma^0_1 \setminus \Delta^0_1$	$\Delta^0_2 \setminus (\Sigma^0_1 \cup \Pi^0_1)$
M_{norm}	$\Delta^0_2 \setminus (\Sigma^0_1 \cup \Pi^0_1)$	$\Delta^0_2 \setminus (\Sigma^0_1 \cup \Pi^0_1)$
$\overline{M}$	$\Pi^0_2 \setminus \Delta^0_2$	$\Delta^0_3 \setminus (\Sigma^0_2 \cup \Pi^0_2)$
$\overline{M}_{\text{norm}}$	$\Delta^0_3 \setminus (\Sigma^0_2 \cup \Pi^0_2)$	$\Delta^0_3 \setminus (\Sigma^0_2 \cup \Pi^0_2)$

Table 1: The computability results on M, M_{norm}, $\overline{M}$, and $\overline{M}_{\text{norm}}$ proved in Section 3. Lower bounds on the complexity of $\overline{M}$ and $\overline{M}_{\text{norm}}$ are given only for specific universal Turing machines.

In Section 3 we consider various different flavors of Solomonoff induction: the universal (Solomonoff) prior M specifies the a priori probability that the universal monotone Turing machine generates x when fed with uniformly random bits as input. It is only a semimeasure and not a measure: it assigns positive probability that the observed string has only finite length. This can be circumvented by normalizing M. Solomonoff's normalization M_{norm} preserves the ratio $M(x1)/M(x0)$ and is limit computable (Δ^0_2-computable). If instead we mix only over programs that compute infinite strings, we get a semimeasure $\overline{M}$, which can be normalized to $\overline{M}_{\text{norm}}$. Moreover, when predicting a sequence, we are primarily interested in the conditional probability $M(xy \mid x)$ (respectively $M_{\text{norm}}(xy \mid y)$, $\overline{M}(xy \mid x)$, or $\overline{M}_{\text{norm}}(xy \mid x)$) that the currently observed string x is continued with y. We show that both M and M_{norm} are limit computable (Δ^0_2), while $\overline{M}$ and $\overline{M}_{\text{norm}}$ are not. Table 1 summarizes our computability results for Solomonoff induction.

For MDPs, planning is already P-complete for finite and infinite horizons (Papadimitriou and Tsitsiklis, 1987). In POMDPs, planning is undecidable (Madani et al., 1999, 2003). The existence of a policy whose expected value exceeds a given threshold is PSPACE-complete (Mundhenk et al., 2000), even for purely epistemic POMDPs in which actions do not change the hidden state (Sabbadin et al., 2007). In Section 4 we derive hardness results for planning in general semicomputable environments; this environment class is even more general than POMDPs. We show that optimal policies are Π^0_2-hard and ε-optimal policies are undecidable.

Moreover, we show that by default, AIXI is not limit computable (Δ^0_2). When picking the next action, two or more actions might have the same value (expected future rewards). The choice between them is easy, but determining whether such a tie exists is difficult. This problem can be circumvented by settling for an ε-optimal policy; we get a limit-computable agent with infinite horizon. However,

Agent	Optimal	ε-Optimal
AIMU	Δ^0_2	Δ^0_1
AINU	Δ^0_3, Π^0_2-hard	Δ^0_2, Σ^0_1-hard
AIXI	Δ^0_3, Σ^0_1-hard	Δ^0_2, Σ^0_1-hard
Entropy-seeking	Δ^0_3	Δ^0_2
Information-seeking	Δ^0_3	Δ^0_2
BayesExp	Δ^0_3	Δ^0_2

Table 2: Computability results for different agent models derived in Section 4, Section 6, and Section 7. AIMU denotes the optimal policy in a computable environment and AINU denotes the optimal policy in a semicomputable environment (see Section 2.6). Hardness results for AIXI are with respect to a specific universal Turing machine; hardness results for ν-optimal policies are with respect to a specific environment $\nu \in \mathcal{M}$. Results for entropy-seeking and information-seeking policies are only for finite horizons.

these results rely on the recursive definition of the value function. In contrast, Hutter (2005) defines the value function as the limit of the iterative value function. In Section 5 we compare these two definitions and show that the recursive definition correctly maximizes expected rewards and has better computability properties.

In Section 6 we show that for finite horizons both the entropy-seeking and the information-seeking agent are Δ^0_3-computable and have limit-computable ε-optimal policies. The weakly asymptotically optimal agent BayesExp relies on optimal policies that are generally not limit computable (Δ^0_2). In Section 7 we give a weakly asymptotically optimal agent based on BayesExp that is limit computable. Table 2 summarizes our results on the computability of these agents.

This journal paper combines the results from Leike and Hutter (2015a) and Leike and Hutter (2015b). We unified notation and results and added more explanations. We also rewrote everything in terms of the recursive value function (which is arguably the correct one), which also removes the distinction between finite and infinite lifetime discounting. Lastly, we added figures for all environments constructed in proofs which should make them easier to understand. A list of notation can be found on page 41.

2. Preliminaries

2.1. The Arithmetical Hierarchy

A set $A \subseteq \mathbb{N}$ is Σ^0_n iff there is a quantifier-free formula η such that

$$k \in A \iff \exists k_1 \forall k_2 \ldots Q_n k_n \; \eta(k, k_1, \ldots, k_n) \tag{1}$$

where $Q_n = \forall$ if n is even, $Q_n = \exists$ if n is odd (Nies, 2009, Def. 1.4.10). (We can also think of η as a computable relation.) A set $A \subseteq \mathbb{N}$ is Π_n^0 iff its complement $\mathbb{N} \setminus A$ is Σ_n^0. The formula η on the right side of (1) is a Σ_n^0*-formula* and its negation is a Π_n^0*-formula*. It can be shown that we can add any bounded quantifiers and duplicate quantifiers of the same type without changing the classification of A. The set A is Δ_n^0 iff A is Σ_n^0 and A is Π_n^0. We get that Σ_1^0 as the class of recursively enumerable sets, Π_1^0 as the class of co-recursively enumerable sets and Δ_1^0 as the class of recursive sets.

We say the set $A \subseteq \mathbb{N}$ is Σ_n^0*-hard* (Π_n^0*-hard*, Δ_n^0*-hard*) iff for any set $B \in \Sigma_n^0$ ($B \in \Pi_n^0$, $B \in \Delta_n^0$), B is many-one reducible to A, i.e., there is a computable function f such that $k \in B \leftrightarrow f(k) \in A$ (Nies, 2009, Def. 1.2.1). We get $\Sigma_n^0 \subset \Delta_{n+1}^0 \subset \Sigma_{n+1}^0 \subset \ldots$ and $\Pi_n^0 \subset \Delta_{n+1}^0 \subset \Pi_{n+1}^0 \subset \ldots$. This hierarchy of subsets of natural numbers is known as the *arithmetical hierarchy*.

By Post's Theorem (Nies, 2009, Thm. 1.4.13), a set is Σ_n^0 if and only if it is recursively enumerable on an oracle machine with an oracle for a Σ_{n-1}^0-complete set.

2.2. Strings

Let $\mathcal{X}$ be some finite set called *alphabet*. The set $\mathcal{X}^* := \bigcup_{n=0}^{\infty} \mathcal{X}^n$ is the set of all finite strings over the alphabet $\mathcal{X}$, the set $\mathcal{X}^\infty$ is the set of all infinite strings over the alphabet $\mathcal{X}$, and the set $\mathcal{X}^\sharp := \mathcal{X}^* \cup \mathcal{X}^\infty$ is their union. The empty string is denoted by ϵ, not to be confused with the small positive real number ε. Given a string $x \in \mathcal{X}^*$, we denote its length by $|x|$. For a (finite or infinite) string x of length $\geq k$, we denote with $x_{1:k}$ the first k characters of x, and with $x_{<k}$ the first $k-1$ characters of x. The notation $x_{1:\infty}$ stresses that x is an infinite string. We write $x \sqsubseteq y$ iff x is a prefix of y, i.e., $x = y_{1:|x|}$.

2.3. Computability of Real-valued Functions

We fix some encoding of rational numbers into binary strings and an encoding of binary strings into natural numbers. From now on, this encoding will be done implicitly wherever necessary.

Definition 1 (Σ_n^0-, Π_n^0-, Δ_n^0-computable)**.** A function $f : \mathcal{X}^* \to \mathbb{R}$ is called Σ_n^0*-computable* (Π_n^0*-computable*, Δ_n^0*-computable*) iff the set $\{(x, q) \in \mathcal{X}^* \times \mathbb{Q} \mid f(x) > q\}$ is Σ_n^0 (Π_n^0, Δ_n^0).[2]

[2] The results presented here also hold for a slightly different definition of computability, see Valenti (2016).

	$\{(x, q) \mid f(x) > q\}$	$\{(x, q) \mid f(x) < q\}$
f is computable	Δ^0_1	Δ^0_1
f is lower semicomputable	Σ^0_1	Π^0_1
f is upper semicomputable	Π^0_1	Σ^0_1
f is limit computable	Δ^0_2	Δ^0_2
f is Δ^0_n-computable	Δ^0_n	Δ^0_n
f is Σ^0_n-computable	Σ^0_n	Π^0_n
f is Π^0_n-computable	Π^0_n	Σ^0_n

Table 3: Connection between the computability of real-valued functions and the arithmetical hierarchy.

A Δ^0_1-computable function is called *computable*, a Σ^0_1-computable function is called *lower semicomputable*, and a Π^0_1-computable function is called *upper semicomputable*. A Δ^0_2-computable function f is called *limit computable*, because there is a computable function ϕ such that

$$\lim_{k \to \infty} \phi(x, k) = f(x).$$

The program ϕ that limit computes f can be thought of as an *anytime algorithm* for f: we can stop ϕ at any time k and get a preliminary answer. If the program ϕ ran long enough (which we do not know), this preliminary answer will be close to the correct one.

Limit-computable sets are the highest level in the arithmetical hierarchy that can be approached by a regular Turing machine. Above limit-computable sets we necessarily need some form of halting oracle. See Table 3 for the definition of lower/upper semicomputable and limit-computable functions in terms of the arithmetical hierarchy.

Lemma 2 (Computability of Arithmetical Operations)**.** *Let $n > 0$ and let $f, g : \mathcal{X}^* \to \mathbb{R}$ be two Δ^0_n-computable functions. Then*

(i) $\{(x, y) \mid f(x) > g(y)\}$ *is* Σ^0_n*,*

(ii) $\{(x, y) \mid f(x) \leq g(y)\}$ *is* Π^0_n*,*

(iii) $f + g$*,* $f - g$*, and* $f \cdot g$ *are* Δ^0_n*-computable, and*

(iv) f/g *is* Δ^0_n*-computable if* $g(x) \neq 0$ *for all* x*.*

(v) $\log f$ *is* Δ^0_n*-computable if* $f(x) > 0$ *for all* x.

Proof. We only prove this for $n > 1$. Since f, g are Δ^0_n-computable, they are limit computable on a level $n-1$ oracle machine. Let ϕ be the function limit computing f on the oracle machine, and let ψ be the function limit computing g on the oracle machine:

$$f(x) = \lim_{k\to\infty} \phi(k, x) \quad \text{and} \quad g(y) = \lim_{k\to\infty} \psi(k, y).$$

By assumption, both ϕ and ψ are Δ^0_{n-1}-computable.

(i) Let $G := \{(x, y, q) \mid g(y) < q\}$, and let $F := \{(x, y, q) \mid q \leq f(x)\}$, both of which are in Δ^0_n by assumption. Hence there are Σ^0_n-formulas φ_G and φ_F such that

$$\begin{aligned}(x, y, q) \in G &\iff \varphi_G(x, y, q)\\ (x, y, q) \in F &\iff \varphi_F(x, y, q)\end{aligned}$$

Now $f(x) > g(y)$ if and only if $\exists q.\ (x, y, q) \in G \cap F$, which is equivalent to the Σ^0_n-forumla

$$\exists q.\ \varphi_G(x, y, q) \ \wedge \ \varphi_F(x, y, q).$$

(ii) Follows from (i).

(iii) Addition, subtraction, and multiplication are continuous operations.

(iv) Division is discontinuous only at $g(x) = 0$. We show this explicitly. By assumption, for any $\varepsilon > 0$ there is a k_0 such that for all $k > k_0$

$$|\phi(x, k) - f(x)| < \varepsilon \quad \text{and} \quad |\psi(x, k) - g(x)| < \varepsilon.$$

We assume without loss of generality that $\varepsilon < |g(x)|$, since $g(x) \neq 0$ by assumption.

$$\begin{aligned}&\left|\frac{\phi(x,k)}{\psi(x,k)} - \frac{f(x)}{g(x)}\right|\\ =&\left|\frac{\phi(x,k)g(x) - f(x)\psi(x,k)}{\psi(x,k)g(x)}\right|\\ \leq&\ \frac{|\phi(x,k)g(x) - f(x)g(x)| + |f(x)g(x) - f(x)\psi(x,k)|}{|\psi(x,k)g(x)|}\\ <&\ \frac{\varepsilon|g(x)| + |f(x)|\varepsilon}{|\psi(x,k)g(x)|}\end{aligned}$$

with $|\psi(x,k)g(x)| = |\psi(x,k)| \cdot |g(x)| > (|g(x)| - \varepsilon)|g(x)|$,

$$< \varepsilon \cdot \frac{|g(x)| + |f(x)|}{(|g(x)| - \varepsilon)|g(x)|} \xrightarrow{\varepsilon \to 0} 0,$$

therefore $f(x)/g(x) = \lim_{k\to\infty} \phi(x,k)/\psi(x,k)$.

(v) Follows from the fact that the logarithm is computable. □

2.4. Algorithmic Information Theory

The following notion of a (semi-)measure is particular to algorithmic information theory.

Definition 3 (Semimeasure (Li and Vitányi, 2008, Def. 4.2.1))**.** A *semimeasure* over the alphabet $\mathcal{X}$ is a function $\nu : \mathcal{X}^* \to [0,1]$ such that

(i) $\nu(\epsilon) \leq 1$, and

(ii) $\nu(x) \geq \sum_{a\in\mathcal{X}} \nu(xa)$ for all $x \in \mathcal{X}^*$.

A semimeasure is called *(probability) measure* iff equalities hold in (i) and (ii) for all x.

Semimeasures are not probability measures in the classical measure theoretic sense. However, semimeasures correspond canonically to classical probability measures on the probability space $\mathcal{X}^\sharp := \mathcal{X}^* \cup X^\infty$ whose σ-algebra is generated by the cylinder sets $\Gamma_x := \{xz \mid z \in \mathcal{X}^\sharp\}$ (Li and Vitányi, 2008, Ch. 4.2).

For a semimeasure ν and a string x with $\nu(x) > 0$ we use $\nu(a \mid x) := \nu(xa)/\nu(x)$ to denote the conditional probability of the continuation a given the string x.

Solomonoff's prior M (Solomonoff, 1964) assigns to a string x the probability that the reference universal monotone Turing machine U computes a string starting with x when fed with uniformly random bits as input. Formally,

$$M(x) := \sum_{p:\, x\sqsubseteq U(p)} 2^{-|p|}, \tag{2}$$

Equivalently, the Solomonoff prior M can be defined as a mixture over all lower semicomputable semimeasures (Wood et al., 2011). The function M is a lower semicomputable semimeasure, but not computable and not a measure (Li and Vitányi, 2008, Lem. 4.5.3).

The distribution M dominates all lower semicomputable semimeasures ν in the sense that there is a constant $w(\nu) > 0$ such that $M(x) \geq w(\nu)\nu(x)$ for all strings $x \in \mathcal{X}^*$. This property is the crucial ingredient to show that Bayesian prediction based on M is very successful:

Theorem 4 (Solomonoff (1978, Thm. 3))**.** *For every computable measure μ,*

$$\mathbb{E}_\mu \left[\sum_{t=1}^{\infty} \sum_{a \in \mathcal{X}} (M(a \mid x_{<t}) - \mu(a \mid x_{<t}))^2 \right] \leq K(\mu) \ln \sqrt{2}.$$

This theorem implies in particular that $M(a \mid x_{<t}) \to \mu(a \mid x_{<t})$ as $t \to \infty$. Moreover, if there is a $\varepsilon > 0$ such that $|\mu(a \mid x_{<t}) - \mu(b \mid x_{<t})| > \varepsilon$ for all $a \neq b \in \mathcal{X}$, then the maximum-likelihood predictor based on M, $\hat{x}_t^M := \arg\max_{a \in \mathcal{X}} M(a \mid x_{<t})$ will only make finitely many more mistakes than a μ-based predictor. See Hutter (2001) for details and additional properties.

A semimeasure ν can be turned into a measure ν_{norm} using *Solomonoff normalization*: $\nu_{\text{norm}}(\epsilon) := 1$ and for all $x \in \mathcal{X}^*$ and $a \in \mathcal{X}$,

$$\nu_{\text{norm}}(xa) := \nu_{\text{norm}}(x) \frac{\nu(xa)}{\sum_{b \in \mathcal{X}} \nu(xb)}. \tag{3}$$

By definition, ν_{norm} is a measure (Li and Vitányi, 2008, Sec. 4.5.3). Moreover, since $M_{\text{norm}} \geq M$, normalization preserves universal dominance: $M \geq w(\nu)\nu$ for all lower semicomputable semimeasures ν, and therefore M_{norm} predicts just as well as M.

The *measure mixture* $\overline{M}$ (Gács, 1983, p. 74) removes the contribution of programs that do not compute infinite strings:

$$\overline{M}(x) := \lim_{n \to \infty} \sum_{y \in \mathcal{X}^n} M(xy). \tag{4}$$

The measure mixture $\overline{M}$ is the same as M except that the contributions by programs that do not produce infinite strings are removed: for any such program p, let k denote the length of the finite string generated by p. Then for $|xy| > k$, the program p does not contribute to $M(xy)$, hence it is excluded from $\overline{M}(x)$.

Similarly to M, the measure mixture $\overline{M}$ is not a (probability) measure since $\overline{M}(\epsilon) < 1$; but in this case normalization (3) is just multiplication with the constant $1/\overline{M}(\epsilon)$, leading to the *normalized measure mixture* $\overline{M}_{\text{norm}}$.

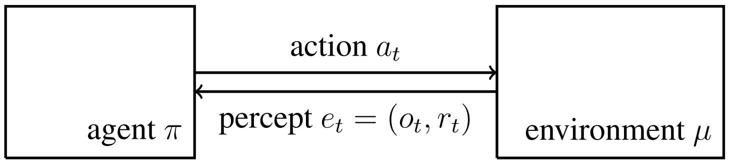


Figure 1: The agent model used in reinforcement learning.

2.5. General Reinforcement Learning

In reinforcement learning the agent interacts with an environment in cycles: at time step t the agent chooses an *action* $a_t \in \mathcal{A}$ and receives a *percept* $e_t = (o_t, r_t) \in \mathcal{E}$ consisting of an *observation* $o_t \in \mathcal{O}$ and a real-valued *reward* $r_t \in [0, 1]$; the cycle then repeats for $t + 1$ (see Figure 1). A *history* is an element of $(\mathcal{A} \times \mathcal{E})^*$. We use $æ \in \mathcal{A} \times \mathcal{E}$ to denote one interaction cycle, and $æ_{1:t}$ to denote a history of length t. The goal in reinforcement learning is to maximize total discounted rewards. A *policy* is a function $\pi : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$ mapping each history to the action taken after seeing this history.

The environment can be stochastic, but is assumed to be semicomputable. In accordance with the AIXI literature (Hutter, 2005), we model environments as lower semicomputable *chronological conditional semimeasures* (LSCCCSs). A *conditional semimeasure* ν takes a sequence of actions $a_{1:\infty}$ as input and returns a semimeasure $\nu(\,\cdot \parallel a_{1:\infty})$ over $\mathcal{E}^\sharp$. A conditional semimeasure ν is *chronological* iff percepts at time t do not depend on future actions, i.e., $\nu(e_{1:t} \parallel a_{1:\infty}) = \nu(e_{1:t} \parallel a'_{1:\infty})$ whenever $a_{1:t} = a'_{1:t}$. Therefore we can write $\nu(e_{1:t} \parallel a_{1:t})$ instead of $\nu(e_{1:t} \parallel a_{1:\infty})$. Despite their name, conditional semimeasures do *not* specify conditional probabilities; the environment ν is *not* a joint probability distribution over actions and percepts. Here we only care about the computability of the environment ν; for our purposes, chronological conditional semimeasures behave just like semimeasures.

We fix a *discount function* $\gamma : \mathbb{N} \to \mathbb{R}$ with $\gamma(t) \geq 0$ and $\sum_{t=1}^{\infty} \gamma(t) < \infty$ and make the following assumptions.

Assumption 5. *(a) The discount function γ is lower semicomputable.*

(b) Rewards are bounded between 0 *and* 1.

(c) The set of actions $\mathcal{A}$ and the set of percepts $\mathcal{E}$ are both finite.

Assumption 5b could be relaxed to bounded rewards because we can rescale rewards $r \mapsto cr + d$ for any $c, d \in \mathbb{R}$ without changing optimal policies if the environment ν is a measure. However, for our value-related results, we require that rewards are nonnegative.

We define the *discount normalization factor* $\Gamma_t := \sum_{i=t}^{\infty} \gamma(i)$. There is no requirement that $\Gamma_t > 0$. In fact, we use γ for both, discounted infinite horizon ($\Gamma_t > 0$ for all t), and finite lifetime m. The *effective horizon* $H_t(\varepsilon)$ is a horizon that is long enough to encompass all but an ε of the discount function's mass:

$$H_t(\varepsilon) := \min\{H \mid \Gamma_{t+H}/\Gamma_t \leq \varepsilon\} \tag{5}$$

We illustrate the environments used in the proofs of our theorems in the form of flowcharts. They should be read as follows. Circles denote *stochastic nodes*, rectangles denote *environment nodes*, and diamonds denote the agent's *choice nodes*. Transitions out of stochastic nodes are labeled with transition probabilities, transitions out of environment nodes are labeled with percepts (observations and rewards), and transitions out of choice nodes are labeled with actions. The initial node is marked with a small incoming arrow (see for example Figure 4). By Assumption 5b the worst possible outcome is getting reward 0 forever, thus we label such states as *hell*. Analogously, getting reward 1 forever is the best possible outcome, thus we label such states as *heaven*.

2.6. The Universal Agent AIXI

Our environment class $\mathcal{M}$ is the class of all LSCCCSs. Typically, Bayesian agents such as AIXI only function well if the true environment is in their hypothesis class. Since the hypothesis class $\mathcal{M}$ is extremely large, the assumption that it contains the true environment is rather weak. We fix a *universal prior* $w : \mathcal{M} \to [0, 1]$ with $w(\nu) > 0$ for all $\nu \in \mathcal{M}$ and $\sum_{\nu \in \mathcal{M}} w(\nu) \leq 1$, given by the reference machine U by $w(\nu) \propto 2^{-K_U(\nu)}$. The universal prior w gives rise to the *universal mixture* ξ, which is a convex combination of all LSCCCSs $\mathcal{M}$:

$$\xi(e_{<t} \parallel a_{<t}) := \sum_{\nu \in \mathcal{M}} w(\nu)\nu(e_{<t} \parallel a_{<t})$$

It is analogous to the Solomonoff prior M but defined for reactive environments. Like M, the universal mixture ξ is lower semicomputable (Hutter, 2005, Sec. 5.10).

Definition 6 (Value Function)**.** The *value* of a policy π in an environment ν given history $æ_{<t}$ is

$$V_\nu^\pi(æ_{<t}) := \frac{1}{\Gamma_t} \sum_{k=t}^{\infty} \sum_{e_{t:k}} \gamma(k) r_k \nu(e_{1:k} \mid e_{<t} \parallel a_{1:k})$$

if $\Gamma_t > 0$ and $V^\pi_\nu(æ_{<t}) := 0$ if $\Gamma_t = 0$ where $a_i := \pi(æ_{<i})$ for all $i \geq t$. The *optimal value* is defined as $V^*_\nu(h) := \sup_\pi V^\pi_\nu(h)$.

This value function is also called *recursive value*, in contrast to iterative value that we discuss in Section 5. This name stems from the following identity (analogously to Hutter (2005, Eq. 4.12)):

$$V^\pi_\nu(æ_{<t}) = \frac{1}{\Gamma_t} \sum_{e_t} \big(\gamma(t) r_t + \Gamma_{t+1} V^\pi_\nu(æ_{1:t})\big) \nu(e_t \mid e_{<t} \parallel a_{1:t})$$

where $a_t := \pi(æ_{<t})$.

An explicit expression for the optimal value in environment ν is

$$V^*_\nu(æ_{<t}) = \frac{1}{\Gamma_t} \lim_{m\to\infty} \max_{a_t \in \mathcal{A}} \sum_{e_t \in \mathcal{E}} \ldots \max_{a_m \in \mathcal{A}} \sum_{e_m \in \mathcal{E}} \sum_{k=t}^{m} \gamma(k) r_k \nu(e_{1:k} \mid e_{<t} \parallel a_{1:k}). \quad (6)$$

Let $æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*$ be some history. We extend the value functions V^π_ν to include initial interactions (in reinforcement learning literature on MDPs these are called Q-values), $V^\pi_\nu(æ_{<t}a_t) := V^{\pi'}_\nu(æ_{<t})$ where π' is the policy π except that it takes action a_t next, i.e., $\pi'(æ_{<t}) := a_t$ and $\pi'(h) := \pi(h)$ for all $h \neq æ_{<t}$. We define $V^*_\nu(æ_{<t}a_t) := \sup_\pi V^\pi_\nu(æ_{<t}a_t)$ analogously.

Definition 7 (Optimal Policy (Hutter, 2005, Def. 5.19 & 5.30))**.** A policy π is *optimal in environment ν (ν-optimal)* iff for all histories the policy π attains the optimal value: $V^\pi_\nu(h) = V^*_\nu(h)$ for all $h \in (\mathcal{A} \times \mathcal{E})^*$.

The discount function is summable, rewards are bounded (Assumption 5b), and actions and percepts spaces are both finite (Assumption 5c), so an optimal policy exists for every environment $\nu \in \mathcal{M}$ (Lattimore and Hutter, 2014, Thm. 10).

For an environment $\nu \in \mathcal{M}$ (an LSCCCS), AINU is defined as a ν-optimal policy $\pi^*_\nu = \arg\max_\pi V^\pi_\nu(\epsilon)$. To stress that the environment is given by a measure $\mu \in \mathcal{M}$ (as opposed to a semimeasure), we use AIMU. If we knew the true environment $\nu \in \mathcal{M}$, we would choose the agent AINU that maximizes ν-expected value. Since we do not know the true environment, we use the universal mixture ξ over all environments in $\mathcal{M}$ instead. This yields the Bayesian agent AIXI: it weighs every environment $\nu \in \mathcal{M}$ according to its prior probability $w(\nu)$. Formally, AIXI is defined as a ξ-optimal policy π^*_ξ for the universal mixture ξ (Hutter, 2005, Ch. 5). Since $\xi \in \mathcal{M}$ and every measure $\mu \in \mathcal{M}$ is also a semimeasure, both AIMU and AIXI are a special case of AINU. However, AIXI is not a special case of AIMU since the mixture ξ is not a measure.

$$M(xy \mid x) > q \iff \forall m \exists k.\ \tfrac{\phi(xy,k)}{\phi(x,m)} > q \iff \exists k \exists m_0 \forall m \geq m_0.\ \tfrac{\phi(xy,k)}{\phi(x,m)} > q$$

Figure 2: A Π^0_2-formula and an equivalent Σ^0_2-formula defining conditional M. Here $\phi(x,k)$ denotes a computable function that lower semicomputes $M(x)$.

Because there can be more than one optimal policy, the definitions of AINU, AIMU and AIXI are not unique. More specifically, a ν-optimal policy maps a history h to

$$\pi^*_\nu(h) :\in \arg\max_{a \in \mathcal{A}} V^*_\nu(ha). \tag{7}$$

If there are multiple actions $\alpha, \beta \in \mathcal{A}$ that attain the optimal value, $V^*_\nu(h\alpha) = V^*_\nu(h\beta)$, we say there is an *argmax tie*. Which action we settle on in case of a tie (how we break the tie) is irrelevant and can be arbitrary.

We also use the following properties.

Lemma 8 (Linearity of V^π_ν in ν (Hutter, 2005, Thm. 5.31))**.** *Let $\nu = q\rho + q'\rho'$ for some $q, q' \geq 0$. Then for all policies π and all histories $æ_{<t}$,*

$$V^\pi_\nu(æ_{<t}) \;=\; q\frac{\rho(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^\pi_\rho(æ_{<t}) + q'\frac{\rho'(e_{<t} \parallel a_{<t})}{\nu(e_{<t} \parallel a_{<t})} V^\pi_{\rho'}(æ_{<t}).$$

Theorem 9 (On-Policy Value Convergence (Hutter, 2005, Thm. 5.36))**.** *Let μ be any environment and π be any policy. Then*

$$V^\pi_\xi(æ_{<t}) - V^\pi_\mu(æ_{<t}) \to 0 \text{ as } t \to \infty\ \mu\text{-almost surely.}$$

Lemma 10 (Mixing Mixtures (Leike and Hutter, 2015c, Lem. 1))**.** *Let $q, q' \in \mathbb{Q}$ such that $q > 0$, $q' \geq 0$, and $q + q' \leq 1$. Let ξ be a universal mixture, and let ρ be an LSCCCS. Then $\xi' := q\xi + q'\rho$ is an LSCCCS and a universal mixture.*

3. The Complexity of Solomonoff Induction

When using the Solomonoff prior M (or one of its sisters M_{norm}, $\overline{M}$, or $\overline{M}_{\text{norm}}$) for sequence prediction, we need to compute the conditional probability $M(xy \mid x) := M(xy)/M(x)$ for finite strings $x, y \in \mathcal{X}^*$. Because $M(x) > 0$ for all finite strings $x \in \mathcal{X}^*$, this quotient is well-defined.

Theorem 11 (Complexity of M, M_{norm}, $\overline{M}$, and $\overline{M}_{\text{norm}}$)**.**

(i) $M(x)$ *is lower semicomputable*

(ii) $M(xy \mid x)$ *is limit computable*

(iii) $M_{\text{norm}}(x)$ *is limit computable*

(iv) $M_{\text{norm}}(xy \mid x)$ *is limit computable*

(v) $\overline{M}(x)$ *is* Π^0_2*-computable*

(vi) $\overline{M}(xy \mid x)$ *is* Δ^0_3*-computable*

(vii) $\overline{M}_{\text{norm}}(x)$ *is* Δ^0_3*-computable*

(viii) $\overline{M}_{\text{norm}}(xy \mid x)$ *is* Δ^0_3*-computable*

Proof. (i) By Li and Vitányi (2008, Thm. 4.5.2). Intuitively, we can run all programs in parallel and get monotonely increasing lower bounds for $M(x)$ by adding $2^{-|p|}$ every time a program p has completed outputting x.

(ii) From (i) and Lemma 2 (iv) since $M(x) > 0$ (see also Figure 2).

(iii) By Lemma 2 (iii) and (iv), and $M(x) > 0$.

(iv) By (iii), Lemma 2 (iv) since $M_{\text{norm}}(x) \geq M(x) > 0$.

(v) Let ϕ be a computable function that lower semicomputes M. Since M is a semimeasure, $M(xy) \geq \sum_z M(xyz)$, hence $\sum_{y \in \mathcal{X}^n} M(xy)$ is nonincreasing in n and thus $\overline{M}(x) > q$ iff $\forall n \exists k \sum_{y \in \mathcal{X}^n} \phi(xy, k) > q$.

(vi) From (v) and Lemma 2 (iv) since $\overline{M}(x) > 0$.

(vii) From (v) and Lemma 2 (iv).

(viii) From (vi) and Lemma 2 (iv) since $\overline{M}_{\text{norm}}(x) \geq \overline{M}(x) > 0$. □

We proceed to show that these bounds are in fact the best possible ones. If M were Δ^0_1-computable, then so would be the conditional semimeasure $M(\,\cdot \mid \cdot\,)$. Thus we could compute the M-adversarial sequence $z_1 z_2 \ldots$ defined by

$$z_t := \begin{cases} 0 & \text{if } M(1 \mid z_{<t}) > \frac{1}{2}, \\ 1 & \text{otherwise.} \end{cases}$$

The sequence $z_1 z_2 \ldots$ corresponds to a computable deterministic measure μ. However, we have $M(z_{1:t}) \leq 2^{-t}$ by construction, so dominance $M(x) \geq w(\mu)\mu(x)$ with $w(\mu) > 0$ yields a contradiction with $t \to \infty$:

$$2^{-t} \geq M(z_{1:t}) \geq w(\mu)\mu(z_{1:t}) = w(\mu) > 0$$

By the same argument, the normalized Solomonoff prior M_{norm} cannot be Δ^0_1-computable. However, since it is a measure, Σ^0_1- or Π^0_1-computability would entail Δ^0_1-computability.

For $\overline{M}$ and $\overline{M}_{\text{norm}}$ we prove the following two lower bounds for specific universal Turing machines. In the following theorem, we use $\overline{M}_{U'}$ to denote the measure mixture $\overline{M}$ defined on the universal Turing U' instead of U.

Theorem 12 ($\overline{M}$ is not Limit Computable)**.** *There is a universal Turing machine U' such that the set $\{(x,q) \mid \overline{M}_{U'}(x) > q\}$ is not in Δ^0_2.*

Proof. Assume the contrary and let $A \in \Pi^0_2 \setminus \Delta^0_2$ and η be a quantifier-free first-order formula such that

$$n \in A \quad \Longleftrightarrow \quad \forall k \exists i.\ \eta(n,k,i). \tag{8}$$

For each $n \in \mathbb{N}$, we define the program p_n as follows.

```
1: procedure $p_n$
2:     output $1^{n+1}0$
3:     $k \leftarrow 0$
4:     while true do
5:         $i \leftarrow 0$
6:         while not $\eta(n,k,i)$ do
7:             $i \leftarrow i+1$
8:         $k \leftarrow k+1$
9:         output $0$
```

Each program p_n always outputs $1^{n+1}0$. Furthermore, the program p_n outputs the infinite string $1^{n+1}0^\infty$ if and only if $n \in A$ by (8). We define U' as follows using our reference machine U.

- $U'(1^{n+1}0)$: Run p_n.
- $U'(00p)$: Run $U(p)$.
- $U'(01p)$: Run $U(p)$ and bitwise invert its output.

By construction, U' is a universal Turing machine. No p_n outputs a string starting with $0^{n+1}1$, therefore $\overline{M}_{U'}(0^{n+1}1) = \frac{1}{4}\big(\overline{M}_U(0^{n+1}1) + \overline{M}_U(1^{n+1}0)\big)$. Hence

$$\begin{aligned}\overline{M}_{U'}(1^{n+1}0) &= 2^{-n-2}\mathbb{1}_A(n) + \tfrac{1}{4}\overline{M}_U(1^{n+1}0) + \tfrac{1}{4}\overline{M}_U(0^{n+1}1) \\ &= 2^{-n-2}\mathbb{1}_A(n) + \overline{M}_{U'}(0^{n+1}1)\end{aligned}$$

If $n \notin A$, then $\overline{M}_{U'}(1^{n+1}0) = \overline{M}_{U'}(0^{n+1}1)$. Otherwise, we have $|\overline{M}_{U'}(1^{n+1}0) - \overline{M}_{U'}(0^{n+1}1)| = 2^{-n-2}$.

Now we assume that $\overline{M}_{U'}$ is limit computable (Δ^0_2-computable), i.e., there is a computable function $\phi : \mathcal{X}^* \times \mathbb{N} \to \mathbb{Q}$ such that $\lim_{k\to\infty} \phi(x, k) = \overline{M}_{U'}(x)$. We get that

$$n \in A \iff \lim_{k\to\infty} \phi(0^{n+1}1, k) - \phi(1^{n+1}0, k) \geq 2^{-n-2},$$

thus A is limit computable, a contradiction. □

Corollary 13 ($\overline{M}_{\text{norm}}$ is not Σ^0_2- or Π^0_2-computable)**.** *There is a universal Turing machine U' such that $\{(x, q) \mid \overline{M}_{\text{norm}U'}(x) > q\}$ is not in Σ^0_2 or Π^0_2.*

Proof. Since $\overline{M}_{\text{norm}} = c \cdot \overline{M}$, there exists a $k \in \mathbb{N}$ such that $2^{-k} < c$ (even if we do not know the value of k). We can show that the set $\{(x, q) \mid \overline{M}_{\text{norm}U'}(x) > q\}$ is not in Δ^0_2 analogously to the proof of Theorem 12, using

$$n \in A \iff \lim_{k\to\infty} \phi(0^{n+1}1, k) - \phi(1^{n+1}0, k) \geq 2^{-k-n-2}.$$

If $\overline{M}_{\text{norm}}$ were Σ^0_2-computable or Π^0_2-computable, this would imply that $\overline{M}_{\text{norm}}$ is Δ^0_2-computable since $\overline{M}_{\text{norm}}$ is a measure, a contradiction. □

Since $M(\epsilon) = 1$, we have $M(x \mid \epsilon) = M(x)$, so the conditional probability $M(xy \mid x)$ has at least the same complexity as M. Analogously for M_{norm} and $\overline{M}_{\text{norm}}$ since they are measures. For $\overline{M}$, we have that $\overline{M}(x \mid \epsilon) = \overline{M}_{\text{norm}}(x)$, so Corollary 13 applies. All that remains to prove is that conditional M is not lower semicomputable.

Theorem 14 (Conditional M is not Lower Semicomputable)**.** *The set $\{(x, xy, q) \mid M(xy \mid x) > q\}$ is not recursively enumerable.*

We gave a different, more complicated proof in Leike and Hutter (2015b). The following, much simpler and more elegant proof is due to Sterkenburg (2016, Prop. 3).

Proof. Assume to the contrary that $M(xy \mid x)$ is lower semicomputable. Let $a \neq b \in \mathcal{X}$. We construct an infinite string x by defining its initial segments $\epsilon =: x(0) \sqsubset x(1) \sqsubset x(2) \sqsubset \ldots \sqsubset x$. At every step n, we enumerate strings $y \in \mathcal{X}^*$ until one is found satisfying $M(a \mid x(n)y) \geq 1/2$; then set $x(n+1) := x(n)yb$. This implies that for infinitely many t there is an n such that $M(b \mid x_{<t}) = M(b \mid x(n)y) \leq 1 - M(a \mid x(n)y) \leq 1/2$. Since we assumed $M(\cdot \mid \cdot)$ to be lower semicomputable, the infinite string x is computable, and hence $M(x_t \mid x_{<t}) \to 1$ by Theorem 4. But this contradicts $M(b \mid x_{<t}) \leq 1/2$ infinitely often. □

4. The Complexity of AINU, AIMU, and AIXI

4.1. Upper Bounds

In this section, we derive upper bounds on the computability of AINU, AIMU, and AIXI. Except for Corollary 22, all results in this section apply generally to any LSCCCS $\nu \in \mathcal{M}$, hence they apply to AIXI even though they are stated for AINU.

In order to position AINU in the arithmetical hierarchy, we represent policies as relations over $(\mathcal{A} \times \mathcal{E})^* \times \mathcal{A}$. These relations are easily identified with sets of natural numbers by encoding the history into one natural number. From now on this translation of policies into sets of natural numbers will be done implicitly wherever necessary.

Lemma 15 (Policies are in Δ_n^0)**.** *If a policy π is Σ_n^0 or Π_n^0, then π is Δ_n^0.*

Proof. Let φ be a Σ_n^0-formula (Π_n^0-formula) defining π, i.e., $\varphi(h, a)$ holds iff $\pi(h) = a$. We define the formula φ',

$$\varphi'(h, a) := \bigwedge_{a' \in \mathcal{A} \setminus \{a\}} \neg\varphi(h, a').$$

The set of actions $\mathcal{A}$ is finite, hence φ' is a Π_n^0-formula (Σ_n^0-formula). Moreover, φ' is equivalent to φ. $\square$

One way to compute the optimal policy is through computing the optimal value function. Hence we can provide upper bounds on the computability of the optimal policy if we have upper bounds on the computability of the optimal value function. The following lemma gives such an upper bound on the computability of the value function for environments in $\mathcal{M}$.

Lemma 16 (Complexity of V_ν^*)**.** *For every LSCCCS $\nu \in \mathcal{M}$, and every lower semicomputable discount function γ, the function V_ν^* is Δ_2^0-computable.*

Proof. The explicit form of the value function (6) has numerator

$$\lim_{m\to\infty} \max_{a_t \in \mathcal{A}} \sum_{e_t \in \mathcal{E}} \ldots \max_{a_m \in \mathcal{A}} \sum_{e_m \in \mathcal{E}} \sum_{i=t}^{m} \gamma(i) r_i \nu(e_{1:i} \parallel a_{1:i}),$$

and denominator $\nu(e_{<t} \parallel a_{<t}) \cdot \Gamma_t$. The numerator is nondecreasing in m because we assumed rewards to be nonnegative (Assumption 5b). Hence both numerator and denominator are lower semicomputable functions, so Lemma 2 (iv) implies that V_ν^* is Δ_2^0-computable. $\square$

From the optimal value function V^*_ν we get the optimal policy π^*_ν according to (7). However, in cases where there is more than one optimal action, we have to break an argmax tie. This happens iff $V^*_\nu(h\alpha) = V^*_\nu(h\beta)$ for two potential actions $\alpha \neq \beta \in \mathcal{A}$. This equality test is more difficult than determining which is larger in cases where they are unequal. Thus we get the following upper bound.

Theorem 17 (Complexity of Optimal Policies)**.** *For any environment $\nu \in \mathcal{M}$, if V^*_ν is Δ^0_n-computable, then there is an optimal policy π^*_ν for the environment ν that is Δ^0_{n+1}.*

Proof. To break potential ties, we pick an (arbitrary) total order $\succ$ on $\mathcal{A}$ that specifies which actions should be preferred in case of a tie. We define

$$\begin{aligned}\pi_\nu(h) = a \;:\Longleftrightarrow \quad & \bigwedge_{a' : a' \succ a} V^*_\nu(ha) > V^*_\nu(ha') \\ \land & \bigwedge_{a' : a \succ a'} V^*_\nu(ha) \geq V^*_\nu(ha').\end{aligned} \tag{9}$$

Then π_ν is a ν-optimal policy according to (7). By assumption, V^*_ν is Δ^0_n-computable. By Lemma 2 (i) and (ii) $V^*_\nu(ha) > V^*_\nu(ha')$ is Σ^0_n and $V^*_\nu(ha) \geq V^*_\nu(ha')$ is Π^0_n. Therefore the policy π_ν defined in (9) is a conjunction of a Σ^0_n-formula and a Π^0_n-formula and thus Δ^0_{n+1}. □

Corollary 18 (Complexity of AINU)**.** *AINU is Δ^0_3 for every environment $\nu \in \mathcal{M}$.*

Proof. From Lemma 16 and Theorem 17. □

Usually we do not mind taking slightly suboptimal actions. Therefore actually trying to determine if two actions have the exact same value seems like a waste of resources. In the following, we consider policies that attain a value that is always within some $\varepsilon > 0$ of the optimal value.

Definition 19 (ε-Optimal Policy)**.** A policy π is *ε-optimal in environment ν* iff $V^*_\nu(h) - V^\pi_\nu(h) < \varepsilon$ for all histories $h \in (\mathcal{A} \times \mathcal{E})^*$.

Theorem 20 (Complexity of ε-Optimal Policies)**.** *For any environment $\nu \in \mathcal{M}$, if V^*_ν is Δ^0_n-computable, then there is an ε-optimal policy π^ε_ν for the environment ν that is Δ^0_n.*

Proof. Let $\varepsilon > 0$ be given. Since the value function $V^*_\nu(h)$ is Δ^0_n-computable, the set $V_\varepsilon := \{(ha, q) \mid |q - V^*_\nu(ha)| < \varepsilon/2\}$ is in Δ^0_n according to Definition 1. Hence we compute the values $V^*_\nu(ha')$ until we get within $\varepsilon/2$ for every $a' \in \mathcal{A}$ and then choose the action with the highest value so far. Formally, let $\succ$ be an arbitrary total order on $\mathcal{A}$ that specifies which actions should be preferred in case of a tie. Without loss of generality, we assume $\varepsilon = 1/k$, and define Q to be an $\varepsilon/2$-grid on $[0, 1]$, i.e., $Q := \{0, 1/2k, 2/2k, \ldots, 1\}$. We define

$$\begin{aligned}\pi^\varepsilon_\nu(h) = a \;:\Longleftrightarrow\; \exists(q_{a'})_{a'\in\mathcal{A}} \in Q^{\mathcal{A}}.\; & \bigwedge_{a'\in\mathcal{A}} (ha', q_{a'}) \in V_\varepsilon \\ & \wedge \bigwedge_{a':a'\succ a} q_a > q_{a'} \wedge \bigwedge_{a':a\succ a'} q_a \geq q_{a'} \\ & \wedge \text{ the tuple } (q_{a'})_{a'\in\mathcal{A}} \text{ is minimal with} \\ & \quad \text{respect to the lex. ordering on } Q^{\mathcal{A}}.\end{aligned} \tag{10}$$

This makes the choice of a unique. Moreover, $Q^{\mathcal{A}}$ is finite since $\mathcal{A}$ is finite, and hence (10) is a Δ^0_n-formula. □

Corollary 21 (Complexity of ε-Optimal AINU)**.** *For any environment $\nu \in \mathcal{M}$, there is an ε-optimal policy for AINU that is Δ^0_2.*

Proof. From Lemma 16 and Theorem 20. □

If the environment $\nu \in \mathcal{M}$ is a measure, i.e., ν assigns zero probability to finite strings, then we get computable ε-optimal policies.

Corollary 22 (Complexity of AIMU)**.** *If the environment $\mu \in \mathcal{M}$ is a measure and the discount function γ is computable, then* AIMU *is limit computable (Δ^0_2), and ε-optimal* AIMU *is computable (Δ^0_1).*

Proof. Let $\varepsilon > 0$ be the desired accuracy. We can truncate the limit $m \to \infty$ in (6) at the $\varepsilon/2$-effective horizon $H_t(\varepsilon/2)$, since everything after $H_t(\varepsilon/2)$ can contribute at most $\varepsilon/2$ to the value function. Any lower semicomputable measure is computable (Li and Vitányi, 2008, Lem. 4.5.1). Therefore V^*_μ as given in (6) is composed only of computable functions, hence it is computable according to Lemma 2. The claim now follows from Theorem 17 and Theorem 20. □

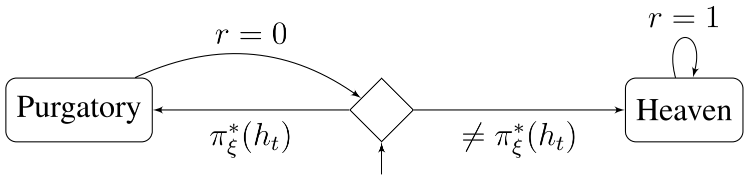


Figure 3: The environment μ from the proof of Theorem 23. The agent gets reward 0 as long as it follows AIXI's policy π^*_ξ that is assumed to be computable. Once the agent deviates from π^*_ξ, it gets reward 1. We get a contradiction because AIXI can learn this environment, so it will eventually decide to take an action that leads to heaven.

4.2. Lower Bounds

We proceed to show that the bounds from the previous section are the best we can hope for. In environment classes where ties have to be broken, AINU has to solve Π^0_2-hard problems (Theorem 24). These lower bounds are stated for particular environments $\nu \in \mathcal{M}$. Throughout this section, we assume that $\Gamma_t > 0$ for all t.

We also construct universal mixtures that yield bounds on ε-optimal policies. There is an ε-optimal AIXI that solves Σ^0_1-hard problems (Theorem 25). For arbitrary universal mixtures, we prove the following weaker statement that only guarantees incomputability.

Theorem 23 (No AIXI is computable)**.** AIXI *is not computable for any universal Turing machine U.*

This theorem follows from the incomputability of Solomonoff induction. Since AIXI uses an analogue of Solomonoff's prior for learning, it succeeds to predict the environment's behavior for its own policy (Theorem 9). If AIXI were computable, then there would be computable environments more powerful than AIXI: they can simulate AIXI and anticipate its prediction, which leads to a contradiction.

Proof. Assume there is a computable policy π^*_ξ that is optimal in the mixture ξ. We define a deterministic environment μ, the *adversarial environment* to π^*_ξ. The environment μ gives rewards 0 as long as the agent follows the policy π^*_ξ, and rewards 1 once the agent deviates. Formally, we ignore observations by setting

$\mathcal{O} := \{0\}$, and define

$$\mu(r_{1:t} \parallel a_{1:t}) := \begin{cases} 1 & \text{if } \forall k \leq t.\, a_k = \pi^*_\xi((ar)_{<k}) \text{ and } r_k = 0 \\ 1 & \text{if } \forall k \leq t.\, r_k = \mathbb{1}_{k \geq i} \\ & \quad \text{where } i := \min\{j \mid a_j \neq \pi^*_\xi((ar)_{<j})\} \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 3 for an illustration of this environment. The environment μ is computable because the policy π^*_ξ was assumed to be computable. Suppose π^*_ξ acts in μ to generate the history $æ_{<t}$, then by Theorem 9 AIXI learns to predict perfectly *on policy*:

$$V^*_\xi(æ_{<t}) = V^{\pi^*_\xi}_\xi(æ_{<t}) \to V^{\pi^*_\xi}_\mu(æ_{<t}) = 0 \text{ as } t \to \infty,$$

since both π^*_ξ and μ are deterministic. Therefore we find a t large enough such that $V^*_\xi(æ_{<t}) < w(\mu)$ where $æ_{<t}$ is the interaction history of π^*_ξ in μ. A policy π with $\pi(æ_{<t}) \neq \pi^*_\xi(æ_{<t})$, gets a reward of 1 in environment μ for all time steps after t, hence $V^\pi_\mu(æ_{<t}) = 1$. With linearity of $V^\pi_\xi(æ_{<t})$ in ξ (Lemma 8),

$$V^\pi_\xi(æ_{<t}) \geq w(\mu)\tfrac{\mu(e_{<t} \| a_{<t})}{\xi(e_{<t} \| a_{<t})} V^\pi_\mu(æ_{<t}) \geq w(\mu),$$

since $\mu(e_{<t} \parallel a_{<t}) = 1$ (μ is deterministic), $V^\pi_\mu(æ_{<t}) = 1$, and $\xi(e_{<t} \parallel a_{<t}) \leq 1$. Now we get a contradiction:

$$w(\mu) > V^*_\xi(æ_{<t}) = \sup_{\pi'} V^{\pi'}_\xi(æ_{<t}) \geq V^\pi_\xi(æ_{<t}) \geq w(\mu) \qquad \square$$

For the remainder of this section, we fix the action space to be $\mathcal{A} := \{\alpha, \beta\}$ with action α favored in ties. The percept space is fixed to a tuple of binary observations and rewards, $\mathcal{E} := \mathcal{O} \times \{0, 1\}$ with $\mathcal{O} := \{0, 1\}$.

Theorem 24 (AINU is Π^0_2-hard)**.** *There is an environment $\nu \in \mathcal{M}$ such that AINU is Π^0_2-hard.*

Proof. Let A be a any Π^0_2-set, and let η be a quantifier-free formula such that

$$n \in A \iff \forall i\, \exists k\, \eta(n, i, k). \tag{11}$$

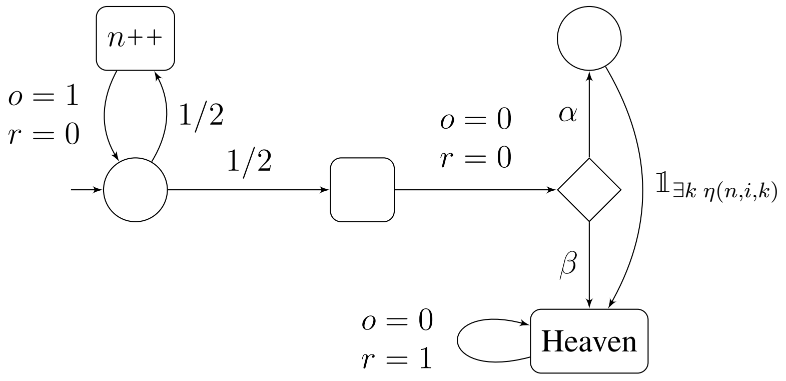


Figure 4: The environment ρ_i from the proof of Theorem 24. The mixture ν over class of environments $\mathcal{M}' := \{\rho_0, \rho_1, \ldots\} \subset \mathcal{M}$ forces AINU to solve Π^0_2-hard problems: Action α is preferred (because of a tie) iff it leads to heaven, which is the case iff $\exists k\ \eta(n, i, k)$.

We define a class of environments $\mathcal{M}' := \{\rho_1, \rho_2, \ldots\}$ where each ρ_i is defined as follows.

$$\rho_i((or)_{1:m} \parallel a_{1:m}) := \begin{cases} 2^{-m}, & \text{if } o_{1:m} = 1^m \text{ and } \forall t \leq m.\ r_t = 0 \\ 2^{-n-1}, & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:m} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \alpha \\ & \quad \text{and } r_t = \mathbb{1}_{t>n+1} \text{ and } \exists k\ \eta(n, i, k) \\ 2^{-n-1}, & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:m} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \beta \\ & \quad \text{and } r_t = \mathbb{1}_{t>n+1} \\ 0, & \text{otherwise.} \end{cases}$$

See Figure 4 for an illustration of these environments. Every ρ_i is a chronological conditional semimeasure by definition, so $\mathcal{M}' \subseteq \mathcal{M}$. Furthermore, every ρ_i is lower semicomputable since η is quantifier-free.

We define our environment ν as a mixture over $\mathcal{M}'$,

$$\nu := \sum_{i \in \mathbb{N}} 2^{-i-1} \rho_i;$$

the choice of the weights on the environments ρ_i is arbitrary but positive. Let π^*_ν be an optimal policy for the environment ν and recall that the action α is preferred in ties. We claim that for the ν-optimal policy π^*_ν,

$$n \in A \iff \pi^*_\nu(1^n 0) = \alpha. \tag{12}$$

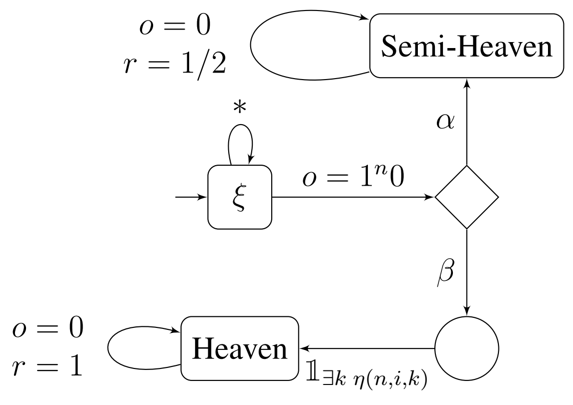


Figure 5: The environment ν from the proof of Theorem 25, which forces AIXI to solve Σ^0_1-hard problems. It functions just like ξ until the observation history is 1^n0. Then, action α is preferred iff heaven is accessible, i.e., iff $\exists k\ \eta(n, i, k)$.

This enables us to decide whether $n \in A$ given the policy π^*_ν, hence proving (12) concludes this proof.

Let $n, i \in \mathbb{N}$ be given, and suppose we are in environment ρ_i and observe 1^n0. Taking action β next yields reward 1 forever; taking action α next yields a reward of 1 if there is a k such that $\eta(n, i, k)$ holds. If this is the case, then

$$V^*_{\rho_i}(1^n0\alpha) = \Gamma_{n+2} = V^*_{\rho_i}(1^n0\beta),$$

and otherwise

$$V^*_{\rho_i}(1^n0\alpha) = 0 < \Gamma_{n+2} = V^*_{\rho_i}(1^n0\beta)$$

(omitting the first $n+1$ actions and rewards in the argument of the value function). We can now show (12): By (11), $n \in A$ if and only if for all i there is a k such that $\eta(n, i, k)$, which happens if and only if $V^*_{\rho_i}(1^n0\alpha) = \Gamma_{n+2}$ for all $i \in \mathbb{N}$, which is equivalent to $V^*_\nu(1^n0\alpha) = \Gamma_{n+2}$, which in turn is equivalent to $\pi^*_\mu(1^n0) = \alpha$ since $V^*_\nu(1^n0\beta) = \Gamma_{n+2}$ and action α is favored in ties. $\square$

Theorem 25 (Some ε-optimal AIXI are Σ^0_1-hard)**.** *There is a universal Turing machine U' and an $\varepsilon > 0$ such that any ε-optimal policy for* AIXI *is Σ^0_1-hard.*

Proof. Let A be a Σ^0_1-set and η be a quantifier-free formula such that $n + 1 \in A$

iff $\exists k\ \eta(n,k)$. We define the environment

$$\nu((or)_{1:t} \parallel a_{1:t}) := \begin{cases} \xi((or)_{1:n} \parallel a_{1:n}), & \text{if } \exists n.\ o_{1:n} = 1^{n-1}0 \\ & \text{ and } a_n = \alpha \\ & \text{ and } \forall t' > n.\ o_{t'} = 0 \wedge r_{t'} = \frac{1}{2} \\ \xi((or)_{1:n} \parallel a_{1:n}), & \text{if } \exists n.\ o_{1:n} = 1^{n-1}0 \\ & \text{ and } a_n = \beta \\ & \text{ and } \forall t' > n.\ o_t = 0 \wedge r_t = 1 \\ & \text{ and } \exists k\ \eta(n-1,k). \\ \xi((or)_{1:t} \parallel a_{1:t}), & \text{if } \nexists n.\ o_{1:n} = 1^{n-1}0 \\ 0, & \text{otherwise.} \end{cases}$$

See Figure 5 for an illustration. The environment ν mimics the universal environment ξ until the observation history is $1^{n-1}0$. Taking the action α next gives rewards $1/2$ forever. Taking the action β next gives rewards 1 forever if $n \in A$, otherwise the environment ν ends at some future time step. Therefore we want to take action β if and only if $n \in A$. We have that ν is an LSCCCS since ξ is an LSCCCS and η is quantifier-free.

We define $\xi' := \frac{1}{2}\nu + \frac{1}{8}\xi$. By Lemma 10 ξ' is a universal lower semicomputable semimeasure. Let $n \in A$ be given and let $h \in (\mathcal{A} \times \mathcal{E})^n$ be any history with observations $o_{1:n} = 1^{n-1}0$. Since $\nu(1^{n-1}0 \mid a_{1:n}) = \xi(1^{n-1}0 \mid a_{1:n})$ by definition, the posterior weights of ν and ξ in ξ' are equal to the prior weights, analogously to the proof of Leike and Hutter (2015c, Thm. 7). In the following, we use the linearity of $V_\rho^{\pi^*_{\xi'}}$ in ρ (Lemma 8), and the fact that values are bounded between 0 and 1 (Assumption 5b). If there is a k such that $\eta(n-1,k)$ holds,

$$\begin{aligned} V^*_{\xi'}(h\beta) - V^*_{\xi'}(h\alpha) &= \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\beta) - \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\alpha) + \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\beta) - \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\alpha) \\ &\geq \tfrac{1}{2} - \tfrac{1}{4} + 0 - \tfrac{1}{8} = \tfrac{1}{8}, \end{aligned}$$

and similarly if there is no k such that $\eta(n-1,k)$ holds, then

$$\begin{aligned} V^*_{\xi'}(h\alpha) - V^*_{\xi'}(h\beta) &= \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\alpha) - \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\beta) + \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\alpha) - \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\beta) \\ &\geq \tfrac{1}{4} - 0 + 0 - \tfrac{1}{8} = \tfrac{1}{8}. \end{aligned}$$

In both cases $|V^*_{\xi'}(h\beta) - V^*_{\xi'}(h\alpha)| > 1/9$. Hence we pick $\varepsilon := 1/9$ and get for every ε-optimal policy $\pi^\varepsilon_{\xi'}$ that $\pi^\varepsilon_{\xi'}(h) = \beta$ if and only if $n \in A$. $\square$

Note the differences between Theorem 23 and Theorem 25: the former talks about optimal policies and shows that they are not computable, but is agnostic towards the underlying universal Turing machine. The latter talks about ε-optimal policies and gives a stronger hardness result, at the cost of depending on one particular universal Turing machine.

5. Iterative Value Function

Historically, AIXI's value function has been defined slightly differently to Definition 6, using a limit extension of an iterative definition of the value function. This definition is the more straightforward to come up with in AI: it is the natural adaptation of (optimal) minimax search in zero-sum games to the (optimal) expectimax algorithm for stochastic environments. In this section we discuss the problems with this definition.

To avoid confusion with the recursive value function V_ν^π, we denote it W_ν^π.[3]

Definition 26 (Iterative Value Function (Hutter, 2005, Def. 5.30))**.** The *iterative value* of a policy π in an environment ν given history $æ_{<t}$ is

$$W_\nu^\pi(æ_{<t}) := \frac{1}{\Gamma_t} \lim_{m\to\infty} \sum_{e_{t:m}} \Big(\nu(e_{1:m} \mid e_{<t} \parallel a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k \Big)$$

if $\Gamma_t > 0$ and $W_\nu^\pi(æ_{<t}) := 0$ if $\Gamma_t = 0$ where $a_i := \pi(e_{<i})$ for all $i \geq t$. The *optimal iterative value* is defined as $W_\nu^*(h) := \sup_\pi W_\nu^\pi(h)$.

Analogously to (6), we can write W_ν^* using alternating $\max$ and $\sum$ operators:

$$W_\nu^*(æ_{<t}) = \frac{1}{\Gamma_t} \lim_{m\to\infty} \max_{a_t\in\mathcal{A}} \sum_{e_t\in\mathcal{E}} \dots \max_{a_m\in\mathcal{A}} \sum_{e_m\in\mathcal{E}} \nu(e_{1:m} \mid e_{<t} \parallel a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k \tag{13}$$

We use *iterative AINU* for the ν-optimal policy according to the iterative value function, and *iterative AIXI* for the ξ-optimal policy according to the iterative value function. Note that iterative AIMU coincides with AIMU since μ is a measure by convention.

[3]Note that in Leike and Hutter (2015a) the use of the symbols V and W is reversed.

Agent	Optimal	ε-Optimal
Iterative AINU	Δ^0_4, Σ^0_3-hard	Δ^0_3, Π^0_2-hard
Iterative AIXI	Δ^0_4, Π^0_2-hard	Δ^0_3, Π^0_2-hard
Iterative AIMU	Δ^0_2	Δ^0_1

Table 4: Computability results for different agent models that use the iterative value function derived in Section 5. Hardness results for AINU are with respect to a specific environment $\nu \in \mathcal{M}$.

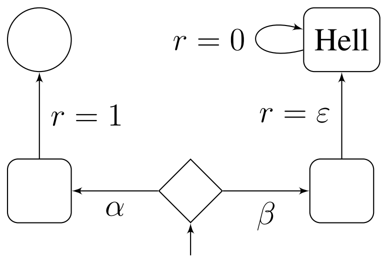


Figure 6: The environment ν from the proof of Proposition 27. Action α yields reward 1, but subsequently the environment ends. Action β yields reward ε and the environment continues forever. Iterative AINU will prefer the suboptimal action β, because it conditions on surviving forever.

Generally, our environment $\nu \in \mathcal{M}$ is only a semimeasure and not a measure, i.e., there is a history $æ_{<t}a_t$ such that

$$1 > \sum_{e_t \in \mathcal{E}} \nu(e_t \mid e_{<t} \parallel a_{1:t}).$$

In such cases, with positive probability the environment ν does not produce a new percept e_t. If this occurs, we shall use the informal interpretation that the environment ν *ended*, but our formal argument does not rely on this interpretation.

The following proposition shows that for a semimeasure $\nu \in \mathcal{M}$ that is not a measure, iterative AINU does not maximize ν-expected rewards. Recall that $\gamma(1)$ states the discount of the first reward. In the following, we assume without loss of generality that $\gamma(1) > 0$, i.e., we are not indifferent about the reward received in time step 1.

Proposition 27 (Iterative AINU is not a ν-Expected Reward Maximizer)**.** *For any $\varepsilon > 0$ there is an environment $\nu \in \mathcal{M}$ that is not a measure and a policy π*

that receives a total of $\gamma(1)$ rewards in ν, but iterative AINU receives only $\varepsilon\gamma(1)$ rewards in ν.

Informally, the environment ν is defined as follows. In the first time step, the agent chooses between the two actions α and β. Taking action α gives a reward of 1, and subsequently the environment ends. Action β gives a reward of ε, but the environment continues forever. There are no other rewards in this environment. See Figure 6. From the perspective of ν-expected reward maximization, it is better to take action α, however iterative AINU takes action β.

Proof of Proposition 27. Let $\varepsilon > 0$. We ignore observations and set $\mathcal{E} := \{0, \varepsilon, 1\}$, $\mathcal{A} := \{\alpha, \beta\}$. The environment ν is formally defined by

$$\nu(r_{1:t} \parallel a_{1:t}) := \begin{cases} 1 & \text{if } a_1 = \alpha \text{ and } r_1 = 1 \text{ and } t = 1 \\ 1 & \text{if } a_1 = \beta \text{ and } r_1 = \varepsilon \text{ and } r_k = 0 \ \forall 1 < k \leq t \\ 0 & \text{otherwise.} \end{cases}$$

Taking action α first, we have $\nu(r_{1:t} \parallel \alpha a_{2:t}) = 0$ for $t > 1$ (the environment ν ends in time step 2 given history α). Hence we conclude

$$V_\nu^*(\alpha) = \frac{1}{\Gamma_t} \lim_{m\to\infty} \sum_{r_{1:m}} \nu(r_{1:m} \parallel \alpha a_{2:m}) \sum_{k=1}^{m} \gamma(k) r_k = 0.$$

Taking action β first we get

$$V_\nu^*(\beta) = \frac{1}{\Gamma_t} \lim_{m\to\infty} \sum_{r_{1:m}} \nu(r_{1:m} \parallel \beta a_{2:m}) \sum_{k=1}^{m} \gamma(k) r_k = \frac{\gamma(1)}{\Gamma_1}\varepsilon.$$

Since $\gamma(1) > 0$ and $\varepsilon > 0$, we have $V_\nu^*(\beta) > V_\nu^*(\alpha)$, and thus iterative AINU will use a policy that plays action β first, receiving a total discounted reward of $\varepsilon\gamma(1)$. In contrast, any policy π that takes action α first receives a larger total discounted reward of $\gamma(1)$. □

Whether it is reasonable to assume that our environment has a nonzero probability of ending is a debate we do not want to engage in here; instead we refer the reader to Martin et al. (2016). Instead, we have a different motivation to use the recursive over the iterative value function: the latter has worse computability properties. Concretely, we show that ε-optimal iterative AIXI has to solve Π_2^0-hard problems and that there is an environment $\nu \in \mathcal{M}$ such that iterative AINU has to solve Σ^0_3-hard problems. In contrast, using the recursive value function, ε-optimal AIXI is Δ^0_2 according to Corollary 21 and AINU is Δ^0_3 according to Corollary 18.

The central difference between V^π_ν and W^π_ν is that for V^π_ν all obtained rewards matter, but for W^π_ν only the rewards in timelines that continue indefinitely. In this sense the value function W^π_ν conditions on surviving forever. If the environment μ is a measure, then the history is infinite with probability one, and so V^π_ν and W^π_ν coincide. Hence this distinction is not relevant for AIMU, only for AINU and AIXI.

Lemma 28 (Complexity of W^*_ν)**.** *For every LSCCCS $\nu \in \mathcal{M}$, the function W^*_ν is Π^0_2-computable.*

Proof. Multiplying (13) with $\Gamma_t \nu(e_{<t} \parallel a_{<t})$ yields $W^*_\nu(æ_{<t}) < q$ if and only if

$$\lim_{m\to\infty} \max_{a_t \in \mathcal{A}} \sum_{e_t \in \mathcal{E}} \ldots \max_{a_m \in \mathcal{A}} \sum_{e_m \in \mathcal{E}} \nu(e_{1:m} \parallel a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k < q\, \Gamma_t\, \nu(e_{<t} \parallel a_{<t}). \tag{14}$$

The inequality's right side is lower semicomputable, hence there is a computable function ψ such that $\psi(\ell) \nearrow q\, \Gamma_t\, \nu(e_{<t} \parallel a_{<t}) =: q'$ as $\ell \to \infty$. For a fixed m, the left side is also lower semicomputable, therefore there is a computable function ϕ such that as $k \to \infty$,

$$\phi(m,k) \nearrow \max_{a_t \in \mathcal{A}} \sum_{e_t \in \mathcal{E}} \ldots \max_{a_m \in \mathcal{A}} \sum_{e_m \in \mathcal{E}} \nu(e_{1:m} \parallel a_{1:m}) \sum_{k=t}^{m} \gamma(k) r_k =: f(m)$$

(In contrast to the recursive value function V^*_ν, this quantity is not nondecreasing in m.) We already know that the limit of $f(m)$ for $m \to \infty$ exists uniquely since optimal policies exist (Lattimore and Hutter, 2014, Thm. 10). Hence using that $\phi(m,k)$ is nondecreasing in k and that $\psi(\ell)$ is nondecreasing in ℓ we can write (14) as

$$\begin{aligned} & \lim_{m\to\infty} f(m) < q' \\ \iff & \exists m_0 \, \forall m \geq m_0.\ f(m) < q' \\ \iff & \exists m_0 \, \forall m \geq m_0 \, \forall k.\ \phi(m,k) < q' \\ \iff & \exists \ell \, \exists m_0 \, \forall m \geq m_0 \, \forall k.\ \phi(m,k) < \psi(\ell), \end{aligned}$$

which is a Σ^0_2-formula. □

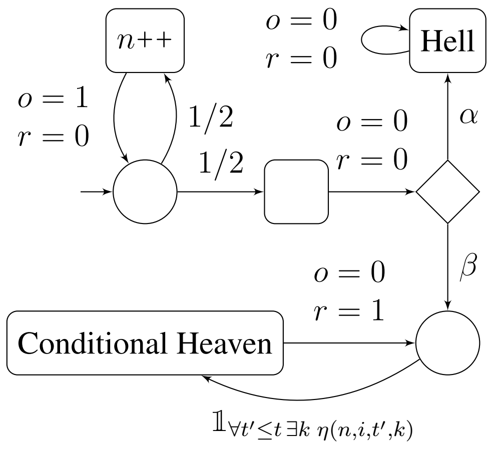


Figure 7: The environment ρ_i from the proof of Theorem 30. The mixture ν over class of environments $\mathcal{M}' := \{\rho_0, \rho_1, \ldots\} \subset \mathcal{M}$ forces iterative AINU to solve Σ^0_3-hard problems. 'Conditional Heaven' is a node that yields reward 1 until $\neg\exists k\; \eta(n,i,t,k)$, at which point the environment ends. Hence action β is preferred in environment ρ_i iff conditional heaven lasts forever (because otherwise $\nu(\ldots) = 0$ and hence $V^*_\nu(\ldots) = 0$) which is the case iff $\forall t\; \exists k\; \eta(n,i,t,k)$.

Note that in the finite lifetime case where m is fixed, the value function W^*_ν is Δ^0_2-computable by Lemma 2 (iv), since $W^*_\nu(æ_{<t}) = f(m)/q'$. In this case, we get the same computability results for iterative AINU as we did in Section 4.1.

Corollary 29 (Complexity of Iterative AINU)**.** *For any environment $\nu \in \mathcal{M}$, iterative AINU is Δ^0_4 and there is an ε-optimal iterative AINU that is Δ^0_3.*

Proof. From Theorem 17, Theorem 20, and Lemma 28. □

We proceed to show corresponding lower bounds as in Section 4.2. For the rest of this section we assume $\Gamma_t > 0$ for all t.

Theorem 30 (Iterative AINU is Σ^0_3-hard)**.** *There is an environment $\nu \in \mathcal{M}$ such that iterative AINU is Σ^0_3-hard.*

Proof. The proof is analogous to the proof of Theorem 24. Let A be any Σ^0_3 set, then there is a quantifier-free formula η such that

$$n \in A \iff \exists i\; \forall t\; \exists k\; \eta(n,i,t,k).$$

We define the environments ρ_i similar to the proof of Theorem 24, except for two changes:

- We replace $\exists k\ \eta(n,i,k)$ with $\forall t' \leq t\ \exists k\ \eta(n,i,t',k)$.
- We switch actions α and β: action β 'checks' the formula η and action α gives a sure reward of 0.

Formally,

$$\rho_i((or)_{1:t} \parallel a_{1:t}) := \begin{cases} 2^{-t}, & \text{if } o_{1:t} = 1^t \text{ and } \forall t' \leq t.\ r_{t'} = 0 \\ 2^{-n-1}, & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:t} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \alpha \\ & \text{ and } \forall t' \leq t.\ r_{t'} = 0 \\ 2^{-n-1}, & \text{if } \exists n.\ 1^n 0 \sqsubseteq o_{1:t} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \beta \\ & \text{ and } \forall t' \leq t.\ r_{t'} = \mathbb{1}_{t' > n+1} \\ & \text{ and } \forall t' \leq t\, \exists k\ \eta(n,i,t',k) \\ 0, & \text{otherwise.} \end{cases}$$

See Figure 7 for an illustration of the environment ρ_i. Every ρ_i is a chronological conditional semimeasure by definition, so $\mathcal{M}' := \{\rho_0, \rho_1, \ldots\} \subseteq \mathcal{M}$. Furthermore, every ρ_i is lower semicomputable since η is quantifier-free.

We define our environment ν as a mixture over $\mathcal{M}'$,

$$\nu := \sum_{i \in \mathbb{N}} 2^{-i-1} \rho_i;$$

the choice of the weights on the environments ρ_i is arbitrary but positive. We get for the ν-optimal policy π^*_ν analogously to the proof of Theorem 24

$$\pi^*_\nu(1^n 0) = \beta \iff \exists i\, \forall t' \leq t\, \exists k\ \eta(n,i,t',k) \iff n \in A,$$

since action α is preferred in ties. ☐

Analogously to Theorem 23, we can show that iterative AIXI is not computable. We also get the following lower bound.

Theorem 31 (Some ε-optimal iterative AIXI are Π^0_2-hard)**.** *There is a universal mixture ξ' and an $\varepsilon > 0$ such that any policy that is ε-optimal according to the iterative value for environment ξ' is Π^0_2-hard.*

Proof. Let A be a Π^0_2-set and η a quantifier-free formula such that

$$n \in A \iff \forall t\, \exists k\ \eta(n,t,k).$$

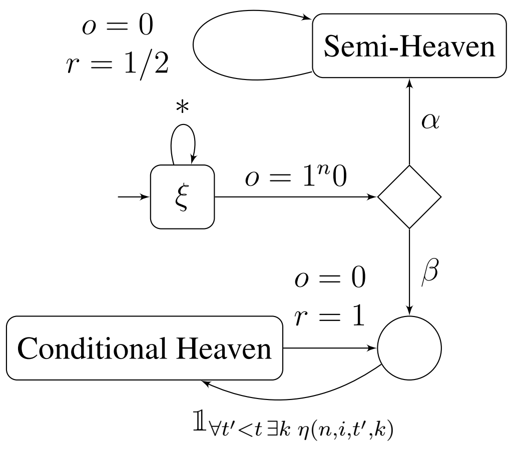


Figure 8: The environment ν from the proof of Theorem 31, which forces ε-optimal iterative AIXI to solve Π^0_2-hard problems. It functions just like ξ until the observation history is 1^n0. Then, action α is preferred iff conditional heaven never ends, i.e., iff $\forall t\, \exists k\; \eta(n, t, k)$.

We proceed analogous to the proof of Theorem 25 except that we choose $\forall t' \leq t\, \exists k\; \eta(n, t, k)$ as a condition for reward 1 after playing action β.

Define the environment

$$\nu((or)_{1:t} \parallel a_{1:t}) := \begin{cases} \xi((or)_{1:n+1} \parallel a_{1:n+1}), & \text{if } \exists n.\ 1^n0 \sqsubseteq o_{1:t} \sqsubseteq 1^n0^\infty \\ & \ \text{and } a_{n+1} = \alpha \\ & \ \text{and } \forall n+1 < k \leq t.\ r_k = 1/2 \\ \xi((or)_{1:n+1} \parallel a_{1:n+1}), & \text{if } \exists n.\ 1^n0 \sqsubseteq o_{1:t} \sqsubseteq 1^n0^\infty \\ & \ \text{and } a_{n+1} = \beta \\ & \ \text{and } \forall n+1 < k \leq t.\ r_k = 1 \\ & \ \text{and } \forall t' \leq t\, \exists k\; \eta(n, t, k) \\ \xi((or)_{1:t} \parallel a_{1:t}), & \text{if } \nexists n.\ 1^n0 \sqsubseteq o_{1:t} \sqsubseteq 1^n0^\infty \\ 0, & \text{otherwise.} \end{cases}$$

See Figure 8 for an illustration of the environment ν. The environment ν mimics the universal environment ξ until the observation history is 1^n0. The next action α always gives rewards $1/2$ forever, while action β gives rewards 1 forever iff $n \in A$. We have that ν is a lower semicomputable semimeasure since ξ is a lower semicomputable semimeasure and η is quantifier-free. We define $\xi' = \frac{1}{2}\nu + \frac{1}{8}\xi$.

By Lemma 10, ξ' is a universal lower semicomputable semimeasure. Let $n \in A$ be given and let $h \in (\mathcal{A} \times \mathcal{O})^{n+1}$ be any history with observations $o_{1:n+1} = 1^n 0$. In the following, we use the linearity of W^*_ρ in ρ (analogously to Lemma 8). If $\forall t \exists k\ \eta(n,t,k)$, then

$$\begin{aligned} W^*_{\xi'}(h\beta) - W^*_{\xi'}(h\alpha) &= \tfrac{1}{2} W^*_\nu(h\beta) - \tfrac{1}{2} W^*_\nu(h\alpha) + \tfrac{1}{8} W^*_\xi(h\beta) - \tfrac{1}{8} W^*_\xi(h\alpha) \\ &\geq \tfrac{1}{2} - \tfrac{1}{4} + 0 - \tfrac{1}{8} = \tfrac{1}{8}, \end{aligned}$$

and similarly if $\neg\forall t \exists k\ \eta(n,t,k)$, then

$$\begin{aligned} W^*_{\xi'}(h\alpha) - W^*_{\xi'}(h\beta) &= \tfrac{1}{2} W^*_\nu(h\alpha) - \tfrac{1}{2} W^*_\nu(h\beta) + \tfrac{1}{8} W^*_\xi(h\alpha) - \tfrac{1}{8} W^*_\xi(h\beta) \\ &\geq \tfrac{1}{4} - 0 + 0 - \tfrac{1}{8} = \tfrac{1}{8}. \end{aligned}$$

In both cases $|W^*_{\xi'}(h\beta) - W^*_{\xi'}(h\alpha)| > 1/9$, hence with $\varepsilon := 1/9$ we have for an ε-optimal policy $\pi^\varepsilon_{\xi'}$ that $\pi^\varepsilon_{\xi'}(h) = \beta$ if and only if $n \in A$. $\square$

6. The Complexity of Knowledge-Seeking

In this section we discuss two variants of knowledge-seeking agents: entropy-seeking agents (Orseau, 2011, 2014) and information-seeking agents (Orseau et al., 2013). The entropy-seeking agent maximizes the Shannon entropy gain, while the information-seeking agent maximizes the expected Bayesian information gain (based on the KL-divergence) in the universal mixture ξ. These quantities are expressed in different value functions. To contrast them with the value function V^π_ν defined in Definition 6, we call the latter *reward-seeking value*.

In this section we use a finite horizon m (possibly dependent on time step t): the knowledge-seeking agent maximizes entropy/information received up to and including time step m. We assume that m (as a function of t) is computable.

Definition 32 (Entropy-Seeking Value Function (Orseau, 2014, Sec. 6))**.** The *entropy-seeking value* of a policy π given history $æ_{<t}$ is

$$V^\pi_{\text{Ent}}(æ_{<t}) := \sum_{e_{t:m}} -\xi_{\text{norm}}(e_{1:m} \mid e_{<t} \parallel a_{1:m}) \log_2 \xi_{\text{norm}}(e_{1:m} \mid e_{<t} \parallel a_{1:m})$$

where $a_i := \pi(æ_{<i})$ for all $i \geq t$.

Definition 33 (Information-Seeking Value Function (Orseau et al., 2013, Def. 1))**.** The *information-seeking value* of a policy π given history $æ_{<t}$ is

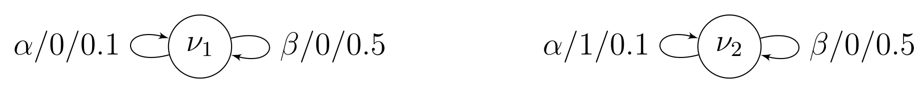
$$V^\pi_{\text{IG}}(æ_{<t}) := \sum_{e_{t:m}} \sum_{\nu \in \mathcal{M}} w(\nu) \frac{\nu(e_{1:m} \parallel a_{1:m})}{\xi_{\text{norm}}(e_{<t} \parallel a_{<t})} \log_2 \frac{\nu(e_{1:m} \mid e_{<t} \parallel a_{1:m})}{\xi_{\text{norm}}(e_{1:m} \mid e_{<t} \parallel a_{1:m})}$$

where $a_i := \pi(æ_{<i})$ for all $i \geq t$.

Analogously to before we define $V^*_{\text{Ent}} := \sup_\pi V^\pi_{\text{Ent}}$ and $V^*_{\text{IG}} := \sup_\pi V^\pi_{\text{IG}}$. We use V^π and V^* in places where either of the entropy-seeking or the information-seeking value function can be substituted. An optimal entropy-seeking policy is defined as $\pi^*_{\text{Ent}} :\in \arg\max_\pi V^\pi_{\text{Ent}}$ and an optimal information-seeking policy is defined as $\pi^*_{\text{IG}} :\in \arg\max_\pi V^\pi_{\text{IG}}$.

The entropy-seeking agent does not work well in stochastic environments because it gets distracted by noise in the environment rather than trying to distinguish environments (Orseau et al., 2013). Moreover, the unnormalized knowledge-seeking agents may fail to seek knowledge in deterministic semimeasures as the following example demonstrates.

Example 34 (Unnormalized Entropy-Seeking)**.** Suppose we use ξ instead of ξ_{norm} in Definition 32. Fix $\mathcal{A} := \{\alpha, \beta\}$, $\mathcal{E} := \{0, 1\}$, and $m := 1$ (we only care about the entropy of the next percept). We illustrate the problem on a simple class of environments $\{\nu_1, \nu_2\}$:



where transitions are labeled with action/percept/probability. Both ν_1 and ν_2 return a percept deterministically or nothing at all (the environment ends). Only action α distinguishes between the environments. With the prior $w(\nu_1) := w(\nu_2) := 1/2$, we get a mixture ξ for the entropy-seeking value function V^π_{Ent}. Then $V^*_{\text{Ent}}(\alpha) \approx 0.432 < 0.5 = V^*_{\text{Ent}}(\beta)$, hence action β is preferred over α by the entropy-seeking agent. But taking action β yields percept 0 (if any), hence nothing is learned about the environment! $\diamondsuit$

Solomonoff's universal prior is extremely good at learning, as stated in Theorem 4. AIXI is a Bayesian reinforcement learning agent that uses this prior to learn the value of its own policy asymptotically (Theorem 9). However, generally it does not learn the result of counterfactual actions that it does not take. Knowledge-seeking agents learn the environment more effectively, because they don't have to balance between exploration and exploitation: they can focus solely on exploration. Both the entropy-seeking agent and the information-seeking agent are *strongly asymptotically optimal* in the class of all deterministic computable environments (Orseau, 2014; Orseau et al., 2013, Thm. 5): the (entropy-seeking/information-seeking) value of their policy converges to the optimal value. Moreover, the information-seeking agent also learns to predict the result of counterfactual actions (Orseau et al., 2013, Thm. 7).

Using the results from Section 4 we can show that ε-optimal knowledge-seeking agents are limit computable (Δ_2^0), and optimal knowledge-seeking agents are Δ_3^0.

Corollary 35 (Computability of Knowledge-Seeking Values). *For fixed m, the value functions V_{Ent}^* and V_{IG}^* are limit computable (Δ_2^0).*

Proof. This follows from Lemma 2 (iii-v) since ξ, ν, and w are lower semicomputable. □

Corollary 36 (Computability of Knowledge-Seeking Policies). *For entropy-seeking and information-seeking agents there are limit-computable ε-optimal policies and Δ_3^0-computable optimal policies.*

Proof. Follows from Corollary 35, Theorem 17, and Theorem 20. □

Note that if we used an infinite horizon with discounting in Definition 32 or Definition 33, then we cannot retain this computability result without further assumptions: we would need that the value functions increase monotonically as $m \to \infty$, as they do for the recursive value function from Definition 6. However, entropy is not a monotone function and may decrease if there are events whose probability converges to something $\geq 1/2$. For the entropy-seeking value function this happens for histories drawn from a deterministic environment μ since $\xi_{\text{norm}} \to \mu$, so the conditionals converge to 1. Similarly, for the information-seeking value function, the posterior belief in one (deterministic) environment might become larger than $1/2$ (depending on the prior and the environment class). Therefore we generally only get that discounted versions of V_{Ent}^* and V_{IG}^* are Δ_3^0 analogously to Lemma 28. Hence optimal discounted entropy-seeking and optimal discounted information-seeking policies are in Δ_4^0 by Theorem 17 and their corresponding ε-optimal siblings are Δ_3^0 by Theorem 20.

7. A Limit Computable Weakly Asymptotically Optimal Agent

In this section, we turn to an objective optimality notion for general reinforcement learning.

Definition 37 (Weak Asymptotic Optimality (Lattimore and Hutter, 2011, Def. 7)). A policy π is *weakly asymptotically optimal* in the class of environments $\mathcal{M}$

Algorithm 1 BayesExp (Lattimore, 2013, Alg. 2).

```
1: while true do
2:     lifetime $m \leftarrow t + H_t(\varepsilon_t)$
3:     if $V^*_{\text{IG}}(æ_{<t}) > \varepsilon_t$ then
4:         follow $\pi^*_{\text{IG}}$ for $m$ steps
5:     else
6:         follow $\pi^*_\xi$ for 1 step
```

iff the reward-seeking value converges to the optimal value on-policy in Cesàro mean, i.e.,

$$\frac{1}{t}\sum_{k=1}^{t}\big(V^*_\nu(æ_{<k}) - V^\pi_\nu(æ_{<k})\big) \xrightarrow{t\to\infty} 0 \quad \nu\text{-almost surely for all } \nu \in \mathcal{M}.$$

Not all discount functions admit weakly asymptotically optimal policies (Lattimore and Hutter, 2011, Thm. 8); a necessary condition is that the effective horizon grows sublinearly (Lattimore, 2013, Thm. 5.5). This is satisfied by geometric discounting, but not by harmonic or power discounting (Hutter, 2005, Tab. 5.41).

This condition is also sufficient: Lattimore (2013, Thm. 5.6) defines a weakly asymptotically optimal agent called *BayesExp*. BayesExp alternates between phases of exploration and phases of exploitation: if the optimal information-seeking value is larger than ε_t, then BayesExp starts an exploration phase, otherwise it starts an exploitation phase. During an exploration phase, BayesExp follows an optimal information-seeking policy for an ε_t-effective horizon. During an exploitation phase, BayesExp follows an ξ-optimal reward-seeking policy for one step (see Algorithm 1).

According to Theorem 24, optimal reward-seeking policies are generally Π^0_2-hard, and for optimal knowledge-seeking policies Corollary 36 shows that they are Δ^0_3. Therefore we get that BayesExp is Δ^0_3:

Corollary 38 (BayesExp is Δ^0_3)**.** *For any universal mixture ξ, BayesExp is Δ^0_3.*

Proof. From Corollary 18, Corollary 35, and Corollary 36. □

However, we do not know BayesExp to be limit computable, and we expect it not to be. However, we can approximate it using ε-optimal policies preserving weak asymptotic optimality.

Theorem 39 (A Limit-Computable Weakly Asymptotically Optimal Agent)**.** *If there is a nonincreasing computable sequence of positive reals* $(\varepsilon_t)_{t \in \mathbb{N}}$ *such that* $\varepsilon_t \to 0$ *and* $H_t(\varepsilon_t)/(t\varepsilon_t) \to 0$ *as* $t \to \infty$*, then there is a limit-computable policy that is weakly asymptotically optimal in the class of all computable stochastic environments.*

Proof. By Corollary 18, there is a limit-computable 2^{-t}-optimal reward-seeking policy π^t_ξ for the universal mixture ξ. By Corollary 36 there are limit-computable $\epsilon_t/2$-optimal information-seeking policies π^t_{IG} with lifetime $t + H_t(\varepsilon_t)$. We define a policy π analogously to Algorithm 1 with π^t_{IG} and π^t_ξ instead of the optimal policies. From Corollary 35 we get that V^*_{IG} is limit computable (Δ^0_2-computable), so the policy π is limit computable. Furthermore, π^t_ξ is 2^{-t}-optimal and $2^{-t} \to 0$, so $V^{\pi^t_\xi}_\xi(æ_{<t}) \to V^*_\xi(æ_{<t})$ as $t \to \infty$.

Now we can proceed analogously to the proof of Lattimore (2013, Thm. 5.6), which consists of three parts. First, it is shown that the value of the ξ-optimal reward-seeking policy π^*_ξ converges to the optimal value for exploitation time steps (line 6 in Algorithm 1) in the sense that $V^{\pi^*_\xi}_\mu \to V^*_\mu$. This carries over to the 2^{-t}-optimal policy π^t_ξ, since the key property is that on exploitation steps, $V^*_{\text{IG}} < \varepsilon_t$; i.e., π only exploits if potential knowledge-seeking value is low. In short, we get for exploitation steps

$$V^{\pi^t_\xi}_\xi(æ_{<t}) \to V^{\pi^*_\xi}_\xi(æ_{<t}) \to V^{\pi^*_\xi}_\mu(æ_{<t}) \to V^*_\mu(æ_{<t}) \text{ as } t \to \infty.$$

Second, it is shown that the density of exploration steps vanishes. This result carries over since the condition $V^*_{\text{IG}}(æ_{<t}) > \varepsilon_t$ that determines exploration steps is exactly the same as for BayesExp and π^t_{IG} is $\varepsilon_t/2$-optimal.

Third, the results of part one and two are used to conclude that π is weakly asymptotically optimal. This part carries over to our proof. □

8. Discussion

8.1. Summary

When using Solomonoff's prior for induction, we need to evaluate conditional probabilities. We showed that conditional M and M_{norm} are limit computable (Theorem 11), and that $\overline{M}$ and $\overline{M}_{\text{norm}}$ are not limit computable (Theorem 12 and Corollary 13). Table 1 on page 3 summarizes our computability results on various versions of Solomonoff's prior. Theses results implies that we can approximate M or M_{norm} for prediction, but not the measure mixture $\overline{M}$ or $\overline{M}_{\text{norm}}$.

In some cases, normalized priors have advantages. As illustrated in Example 34, unnormalized priors can make the entropy-seeking agent mistake the entropy gained from the probability assigned to finite strings for knowledge. From $M_{\text{norm}} \geq M$ we get that M_{norm} predicts just as well as M, and by Theorem 11 we can use M_{norm} without losing limit computability.

Table 2 on page 4 summarizes our computability results for the agents AINU, AIXI, and AINU: AINU is Δ^0_3 and restricting to ε-optimal policies decreases the level by one (Corollary 18 and Corollary 21). For environments that almost surely continue forever (semimeasure that are measures), AIMU is limit-computable and ε-optimal AIMU is computable (Corollary 22). In Section 4.2 we proved that these computability bounds on AINU are generally unimprovable (Theorem 24 and Theorem 25). Additionally, we proved weaker lower bounds for AIXI independent of the universal Turing machine (Theorem 23) and for ε-optimal AIXI for specific choices of the universal Turing machine (Theorem 25).

When the environment ν has nonzero probability of not producing a new percept, the iterative definition of AINU (Definition 26) originally given by Hutter (2005, Def. 5.30) fails to maximize ν-expected rewards (Proposition 27). Moreover, the policies are one level higher in the arithmetical hierarchy (see Table 4 on page 26). We proved upper (Corollary 29) and lower bounds (Theorem 30 and Theorem 31). The difference between the recursive value function V and the iterative value function W is readily exposed in the difference between the universal prior M and the measure mixture $\overline{M}$: Just like W conditions on surviving forever, so does $\overline{M}$ eliminate the weight of programs that do not produce infinite strings. Both $\overline{M}$ and W are not limit computable (Δ^0_2-computable) for this reason.

We considered ε-optimality to avoid having to determine argmax ties. This ε does not have to be constant over time, we may let $\varepsilon \to 0$ as $t \to \infty$ at any computable rate. With this we retain the computability results of ε-optimal policies and get that the value of the $\varepsilon(t)$-optimal policy $\pi^{\varepsilon(t)}_\nu$ converges rapidly to the ν-optimal value:

$$V^*_\nu(æ_{<t}) - V^{\pi^{\varepsilon(t)}_\nu}_\nu(æ_{<t}) \to 0 \text{ as } t \to \infty.$$

In Section 2.5 we defined the environment as a lower semicomputable chronological semimeasure over percepts given actions. When determining the probability of the next percept e_t in an environment ν, we have to compute $\nu(e_{1:t} \mid e_{<t} \parallel a_{1:t})$. Alternatively, we could have defined the environment as a lower semicomputable mapping from histories $æ_{<t}a_t$ to probabilities over the next percept e_t. For

the proof of Lemma 16 and Lemma 28 we only need that $\nu(e_{1:t} \parallel a_{1:t})$ is lower semicomputable computable. While this new definition makes no difference for the computability of AINU, it matters for AIXI because in the mixture ξ over all of these environments is no longer lower semicomputable.

Any method that tries to tackle the reinforcement learning problem has to balance between exploration and exploitation. AIXI strikes this balance in the Bayesian way. However, it has been argued that this does not lead to enough exploration (Orseau, 2013; Leike and Hutter, 2015c). To counteract this, we can add an explorative component to the agent, akin to Orseau's knowledge-seeking agents. Instead of maximizing rewards, knowledge-seeking agents maximize the expected information gained about the environment (Orseau, 2011, 2014; Orseau et al., 2013). In Section 6 we show that ε-optimal knowledge-seeking agents are limit computable (Δ^0_2-computable) if we use the recursive definition of the value function.

We set out with the goal of finding an ideal reinforcement learning agent that is limit computable (Δ^0_2-computable). The Bayesian agent AIXI could be considered one suitable candidate, despite the problems just mentioned. However, we showed only an ε-approximation to be limit computable. Another suitable candidates are asymptotically optimal agents, which in contrast to AIXI are optimal in an objective sense (Leike and Hutter, 2015c): they converge to optimal behavior asymptotically. We discussed Lattimore's BayesExp (Lattimore, 2013, Ch. 5), which relies on Solomonoff induction to learn its environment and on a knowledge-seeking component for extra exploration. Our results culminated in a limit-computable weakly asymptotically optimal agent based on Lattimore's BayesExp (Theorem 39). In this sense our goal has been achieved.

8.2. Open Questions

The lower bounds in Theorem 25, Theorem 31, Theorem 12, and Corollary 13 were only provided with respect to one specifically chosen universal Turing machine. This implies that without further assumption on our reference machine, we cannot improve the lower bound. However, this result could be strengthened if it turned out that these lower bounds hold for *all* universal Turing machines. Below we give a list of the questions that we have left unanswered.

1. Is there a universal mixture $\xi \in \mathcal{M}$ such that iterative AIXI's argmax operator ties infinitely often? This would increase the lower bound for (ε-optimal) iterative AIXI.

2. Is there a universal mixture $\xi \in \mathcal{M}$ such that iterative AIXI's argmax operator ties only finitely many times? This would decrease the upper bound for (ε-optimal) iterative AIXI.

3. Do the lower bounds on $\overline{M}$ and $\overline{M}_{\text{norm}}$ given in Theorem 12 and Corollary 13 hold for every universal Turing machine?

Acknowledgements.. We thank Tom Sterkenburg for feedback on the proof of Theorem 14, Daniel Filan for pointing out a problem in the definition of LSC-CCSs, and Manlio Valenti for pointing out a problem in the proof of Lemma 28, both of which have been fixed in this version. This work was supported by ARC grant DP150104590.

References

David Blackwell and Lester Dubins. Merging of opinions with increasing information. *The Annals of Mathematical Statistics*, pages 882–886, 1962.

Péter Gács. On the relation between descriptional complexity and algorithmic probability. *Theoretical Computer Science*, 22(1):71–93, 1983.

Marcus Hutter. A theory of universal artificial intelligence based on algorithmic complexity. Technical report, 2000. http://arxiv.org/abs/cs.AI/0004001.

Marcus Hutter. New error bounds for Solomonoff prediction. *Journal of Computer and System Sciences*, 62(4):653–667, 2001.

Marcus Hutter. *Universal Artificial Intelligence*. Springer, 2005.

Tor Lattimore. *Theory of General Reinforcement Learning*. PhD thesis, Australian National University, 2013.

Tor Lattimore and Marcus Hutter. Asymptotically optimal agents. In *Algorithmic Learning Theory*, pages 368–382. Springer, 2011.

Tor Lattimore and Marcus Hutter. General time consistent discounting. *Theoretical Computer Science*, 519:140–154, 2014.

Jan Leike and Marcus Hutter. On the computability of AIXI. In *Uncertainty in Artificial Intelligence*, pages 464–473, 2015a.

Jan Leike and Marcus Hutter. On the computability of Solomonoff induction and knowledge-seeking. In *Algorithmic Learning Theory*, pages 364–378, 2015b.

Jan Leike and Marcus Hutter. Bad universal priors and notions of optimality. In *Conference on Learning Theory*, pages 1244–1259, 2015c.

Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Texts in Computer Science. Springer, 3rd edition, 2008.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI*, pages 541–548, 1999.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1):5–34, 2003.

Jarryd Martin, Tom Everitt, and Marcus Hutter. Death and suicide in universal artificial intelligence. In *Artificial General Intelligence*, 2016.

Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

André Nies. *Computability and Randomness*. Oxford University Press, 2009.

Laurent Orseau. Universal knowledge-seeking agents. In *Algorithmic Learning Theory*, pages 353–367. Springer, 2011.

Laurent Orseau. Asymptotic non-learnability of universal agents with computable horizon functions. *Theoretical Computer Science*, 473:149–156, 2013.

Laurent Orseau. Universal knowledge-seeking agents. *Theoretical Computer Science*, 519:127–139, 2014.

Laurent Orseau, Tor Lattimore, and Marcus Hutter. Universal knowledge-seeking agents for stochastic environments. In *Algorithmic Learning Theory*, pages 158–172. Springer, 2013.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Samuel Rathmanner and Marcus Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

Régis Sabbadin, Jérôme Lang, and Nasolo Ravoanjanahry. Purely epistemic Markov decision processes. In *AAAI*, pages 1057–1062, 2007.

Ray Solomonoff. A formal theory of inductive inference. Parts 1 and 2. *Information and Control*, 7(1):1–22 and 224–254, 1964.

Ray Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, 24(4):422–432, 1978.

Tom F Sterkenburg. Putnam's diagonal argument and the impossibility of a universal learning machine. Technical report, Centrum Wiskunde & Informatica, 2016. `http://philsci-archive.pitt.edu/12096/`.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Manlio Valenti. On the notion of Kolmogorov complexity and the computability of learning agents. Master's thesis, University of Trento, 2016.

Joel Veness, Kee Siong Ng, Marcus Hutter, William Uther, and David Silver. A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40(1):95–142, 2011.

Ian Wood, Peter Sunehag, and Marcus Hutter. (Non-)equivalence of universal priors. In *Solomonoff 85th Memorial Conference*, pages 417–425. Springer, 2011.

List of Notation

$:=$	defined to be equal
$:\in$	defined to be an element of
$\nearrow$	a monotone increasing limit
$\mathbb{N}$	the natural numbers, starting with 0
$\#A$	the cardinality of the set A, i.e., the number of elements

$\mathbb{1}_A$	the characteristic function that is 1 if its argument is an element of the set A and 0 otherwise
$\mathcal{X}^*$	the set of all finite strings over the alphabet $\mathcal{X}$
$\mathcal{X}^\infty$	the set of all infinite strings over the alphabet $\mathcal{X}$
$\mathcal{X}^\sharp$	$\mathcal{X}^\sharp := \mathcal{X}^* \cup \mathcal{X}^\infty$, the set of all finite and infinite strings over the alphabet $\mathcal{X}$
x, y	finite or infinite strings, $x, y \in \mathcal{X}^\sharp$
$x \sqsubseteq y$	the string x is a prefix of the string y
$\mathcal{A}$	the (finite) set of possible actions
$\mathcal{O}$	the (finite) set of possible observations
$\mathcal{E}$	the (finite) set of possible percepts, $\mathcal{E} \subset \mathcal{O} \times \mathbb{R}$
α, β	two different actions, $\alpha, \beta \in \mathcal{A}$
a_t	the action in time step t
e_t	the percept in time step t
o_t	the observation in time step t
r_t	the reward in time step t, bounded between 0 and 1
$æ_{<t}$	a history of length $t-1$, $æ_{<t} = a_1 e_1 a_2 e_2 \dots a_{t-1} e_{t-1}$
h	a history, $h \in (\mathcal{A} \times \mathcal{E})^*$
γ	the discount function $\gamma : \mathbb{N} \to \mathbb{R}_{\geq 0}$
Γ_t	a discount normalization factor, $\Gamma_t := \sum_{i=t}^\infty \gamma(i)$
$H_t(\varepsilon)$	ε-effective horizon, defined in (5)
π	a policy, i.e., a function $\pi : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$
π^*_ν	an optimal policy for environment ν
π^ε_ν	an ε-optimal policy for environment ν
V^π_ν	recursive (ν-expected) value function of the policy π
W^π_ν	iterative value function of the policy π in environment ν
ϕ, ψ	computable functions
φ, η	first-order formulas of Peano arithmetic, η is quantifier-free
n, k	natural numbers
t	(current) time step
i	time step, natural number
m	lifetime of the agent
$\mathcal{M}$	the class of all lower semicomputable chronological conditional semimeasures; our environment class
ν, ρ	lower semicomputable chronological conditional semimeasures, $\nu, \rho \in \mathcal{M}$
μ	a computable chronological conditional measure, $\mu \in \mathcal{M}$
ξ	the universal mixture over all environments in $\mathcal{M}$

A, B	sets of natural numbers
ε	a small positive real number
ϵ	the empty string, the history of length 0
$K_U(x)$	Kolmogorov complexity of the string x on the universal Turing machine U
$K_U(\nu)$	Kolmogorov complexity of the index of ν in the enumeration of all lower semicomputable semimeasures on the universal Turing machine U