Space-time embedded intelligence

Laurent Orseau, Mark Ring

Space-Time Embedded Intelligence

Laurent Orseau[1] and Mark Ring[2]

[1] AgroParisTech UMR 518 / INRA
16 rue Claude Bernard, 75005 Paris, France
laurent.orseau@agroparistech.fr
http://www.agroparistech.fr/mia/orseau/
[2] IDSIA / University of Lugano / SUPSI
Galleria 2, 6928 Manno-Lugano, Switzerland
mark@idsia.ch
http://www.markring.com

Abstract. This paper presents the first formal measure of intelligence for agents fully embedded within their environment. Whereas previous measures such as Legg's universal intelligence measure and Russell's bounded optimality provide theoretical insights into agents that interact with an external world, ours describes an intelligence that is computed by, can be modified by, and is subject to the time and space constraints of the environment with which it interacts. Our measure merges and goes beyond Legg's and Russell's, leading to a new, more realistic definition of artificial intelligence that we call *Space-Time Embedded Intelligence*.

Keywords: Intelligence measure, AIXI, bounded optimality, real-world assumptions.

1 Introduction

Artificial General Intelligence (AGI) is the field whose goal is to understand, design and build programs or machines that are or can become at least as intelligent as humans. We believe that this goal cannot be achieved without a formal, sound and *practical* theory of artificial intelligence. In the end we seek an equation of practical intelligence, the solution to which could be implemented on a computer and give rise to an artificial agent of genuine general intelligence.

Theoretical AGI may have begun with Solomonoff [12], who gave us the means for assigning a probability to any (stochastic) computable sequence. Hutter [3] used this universal probability distribution to define the optimally rational reinforcement-learning agent AIXI, the first formal and sound definition of universal artificial intelligence. Legg [4] turned AIXI inside-out to give the first universal measure of intelligence (rationality) of computable agents. None of this work, however, could be considered a *practical* theory of AI, because none of it takes into account the constraints of the real world, most importantly the limitation of computational resources.

Russell [10] introduced *bounded optimality*, which explicitly incorporates the constraints of real-world computer architectures and which can be easily extended to use Solomonoff's universal prior (see also Goertzel [2] for related ideas).

J. Bach, B. Goertzel, and M. Iklé (Eds.): AGI 2012, LNAI 7716, pp. 209–218, 2012.

However, in every case this previous work has adopted the traditional *agent framework*, in which the agent and environment interact as separate entities: the computation of the agent is in principle performed externally to that of the environment. Although quite successful as a working hypothesis, this framework, reminiscent of dualism in the theory of mind [9], can be problematic in the real world: for example, an agent that does not recognize that its computing device or memory might be altered by an external source may tend toward highly risky behaviors.

In previous work [5,8] we began examining the theoretical consequences of integrating an intelligent agent into its environment by, for example, allowing the environment to modify the agent's source code. And in a companion paper [6], we consider agents whose *memory* is integrated into and can be altered by the environment. In the present paper, we formulate a more generalized framework—more reminiscent of physicalism [13] than dualism—where agents are fully integrated into their environment: not just modifiable by it, but actually *computed* by it. While it marks a considerable departure from the traditional agent framework, this new formalization is surprisingly simple and deep.

After introducing notation and reviewing relevant background concepts, the paper proceeds in two steps: first generalizing the agent framework to *space-embedded* agents, which share computational storage with the environment; and second, enhancing the framework with *space-time* embedding, in which all the agent's computations are performed by the environment.

2 Notation

The notation is similar to that of Orseau & Ring [5,8,6]. At some time t the agent outputs actions $a_t \in A$ to the environment, which returns observations $o_t \in O$ to the agent. The sequence of all actions up to time t is written $a_{1:t}$, while the sequence $a_{1:t-1}$ is sometimes written $a_{\prec t}$, and similarly for other sequences. An action and observation from the same time step (an "interaction pair") is denoted $\overline{ao}_t$, and the history of interaction up to t is $\overline{ao}_{1:t}$. The empty sequence is denoted λ. Measures and semi-measures are denoted by Greek letters.

3 Legg's Measure of Intelligence

Legg [4] gave the first universal definition of intelligence, providing an equation to assign a value $\Upsilon(\pi) := V(\pi, \lambda)$ to each (here stochastic) policy π:[1]

$$V(\pi, \overline{ao}_{\prec t}) := \sum_{a_t} \pi(a_t \mid \overline{ao}_{\prec t}) \sum_{o_t} \xi^{\mathrm{RS}}(o_t \mid \overline{ao}_{\prec t} a_t) \Big[r_t + V(\pi, \overline{ao}_{1:t}) \Big]$$

$$\xi^{\mathrm{RS}}(o_{1:t} \mid a_{1:t}) := \sum_{\nu \in \mathcal{M}^{\mathrm{RS}}} 2^{-K(\nu)} \nu(o_{1:t} \mid a_{1:t}) \ ,$$

[1] A stochastic policy $\pi(a_t|\overline{ao}_{\prec t})$ specifies the probability that the agent chooses action a_t given the current interaction history $\overline{ao}_{\prec t}$.

where $r_t = r(o_t)$ is the reward output by the environment at step t; $\mathcal{M}^{\mathrm{RS}}$ is the set of all *reward summable* stochastic environments ν[2]; and $K(\nu)$ is the Kolmogorov complexity of ν. Considering a set of all computable such environments ensures the *generality* of the intelligence of the agent.

This measure of intelligence allows the comparison of various agents depending on the score they obtain in an infinite number of weighted environments. According to this measure, AIXI has the highest intelligence score; *i.e.*, AIXI$= \arg\max_{\pi \in \Pi} V(\pi, \lambda)$, where Π is the set of all approximable policies.

4 Russell's Bounded Optimality

Legg's definition ignores the agent's computational resource requirements and considers intelligence to be independent of such constraints. It is mathematically aesthetic and also useful, but because it does not include time and space constraints, an actual agent designed according to this measure (namely, AIXI), would compute forever, never taking any action at all.

In 1995 (before AIXI was defined), Russell [10] gave a definition of *bounded-optimality*, which does take real-world constraints into account, specifically the constraints of a given computing *architecture* (see Goetzel [2] for related ideas). A given architecture M (described as an interpreter) defines a set of policies $\pi \in \Pi^M$, subject to time, space and other possible constraints of the architecture M. At each interaction step, the policy is run for a single *time step* (*e.g.*, 0.01 seconds, measured in real-world time for a given architecture), continuing the computation of the last time step, and possibly failing to output an action for the current interaction step, in which case a default action is chosen.

The value of a policy is measured by a *utility function* u (to be defined for the task at hand) in a set of environments $q \in \mathcal{Q}$ with $V(\pi, q) := u(h(\pi, q))$ where $h(\pi, q)$ generates the interaction history $\overline{ao}_{1:\infty}$ of the policy π in the deterministic environment q. The value of a policy over the set $\mathcal{Q}$ of environments is defined by $V(\pi, \mathcal{Q}) := \sum_{q \in \mathcal{Q}} p(q) V(\pi, q)$, for a probability distribution p over the environments. The optimal agent π^* subject to the constraints of the architecture M is defined by $\pi^* := \arg\max_{\pi \in \Pi^M} V(\pi, \mathcal{Q})$.

Self-modifying Resource-Bounded Universal Intelligence

Although not explicitly stated, it seems reasonable to assume that Russell's definition of bounded optimality also includes self-modifiable policies, *i.e.*, those that can optimize every aspect of themselves, to be more efficient in both computation time and memory space. (This is in fact the core idea behind the Gödel Machine [11].)

It is straightforward to merge Legg's intelligence measure with Russell's bounded optimality, and the result is an optimal, self-modifying,

[2] A stochastic environment $\nu(o_{1:t}|a_{1:t})$ specifies the probability that the environment produces observation sequence $o_{1:t}$ given action sequence $a_{1:t}$. A reward-summable environment ensures the agent's total cumulated lifetime reward does not exceed 1.

resource-bounded universal agent. To do so we first define a set of policies $\Pi^{\tilde{t},\tilde{l}}$ based on a decomposition of architecture M into a reference machine U (like a universal Turing machine),a time $\tilde{t}$ of unitary computation steps per interaction, and a memory space $\tilde{l}$ that contains both the source code of the agent and the usable memory.

In the remainder of the paper, we will consider only stochastic policies. A self-modifying stochastic policy $\pi_{\in}\Pi^{\tilde{t},\tilde{l}}$ at time step t defines the probability $\pi_t(\langle a_t, \pi_{t+1}\rangle \mid o_{t-1})$ of outputting: (1) some action $a_t \in A$ at step t and (2) some stochastic policy $\pi_{t+1} \in \Pi^{\tilde{t},\tilde{l}}$ for use by the agent at $t+1$; both conditioned on the last observation o_{t-1} output by the environment, and this computation must be done within $\tilde{t}$ computation steps and $\tilde{l}$ bits of memory (otherwise some default values are output). The new code π_{t+1} might, for example, be the same as π_t or only slightly different, perhaps with o_{t-1} written somewhere in it.

The environment ρ outputs an observation o_t with a probability $\rho(o_t \mid \overline{ao}_{\prec t} a_t)$ depending on the current interaction history (not taking into account the sequence of policies of the agent). For generality, ρ is defined as a universal distribution[3] like ξ [15,3] (or ξ^{RS} above), such that $w_\rho(\nu) > 0$ for some prior weight $w_\rho(\nu)$ of any stochastic environment $\nu \in \mathcal{M}$ and $\sum_{\nu\in\mathcal{M}} w_\rho(\nu) \leq 1$. We use a utility function $u(\overline{ao}_{\prec t}) \in [0,1]$ that assigns a utility value to each interaction history, whose cumulated value over the future is discounted by a *horizon function* γ_t so that $\sum_{t=1}^{\infty} \gamma_t = 1$ to ensure convergence (in ξ^{RS} the horizon function is considered to be a part of the environment).

The optimal self-modifying, resource-bounded, universal agent π^* can now be defined:[4]

$$\pi^* := \arg\max_{\pi_1 \in \Pi^{\tilde{t},\tilde{l}}} V(\pi_1, \lambda)$$

$$V(\pi_t, \overline{ao}_{\prec t}) := \sum_{\langle a_t, \pi_{t+1}\rangle} \pi_t(\langle a_t, \pi_{t+1}\rangle \mid o_{t-1}) \quad \times$$

$$\sum_{o_t} \rho(o_t \mid \overline{ao}_{\prec t} a_t) \Big[\gamma_t u(\overline{ao}_{1:t}) + V(\pi_{t+1}, \overline{ao}_{1:t})\Big] .$$

This description shows that the optimal policy achieves greatest average weighted discounted utility in all possible futures by (a) choosing good actions within the time and space constraints $\tilde{t}$ and $\tilde{l}$, and (b) choosing good future policies for itself (within the same time and space constraints).

[3] ρ can be seen equivalently either as a single environment or a mixture of environments. The best way to think about it in the present case might be to consider ρ as a *universal* semi-measure (because we, humans, have no absolute certainty about what the true environment is), but biased with all the knowledge we can, or the knowledge we think is relevant. Thinking of ρ as a non-universal but accurate model of the real world is also acceptable (although arguably non-realistic).

[4] This is an uncomputable definition, but the solution of this equation is a computable optimal resource-bounded agent.

5 Embedded Resource-Bounded Intelligence

The self-modifying, resource-bounded intelligence just described does not take into account the fact that the environment may have read or even write access to the memory space of the agent, containing its memory of the past and its source code. We now propose a new definition of intelligence that extends self-modifying, resource-bounded intelligence in two ways. The first extension, which we call *space embedding* (Section 5.1), moves the code and memory of the agent into the environment. The second extension, which we call *space-time embedding* (Section 5.2), allows the environment itself (rather than an external, possibly infinite computational device) to compute the agent's code.

5.1 Space-Embedded Agents

In the traditional Reinforcement Learning (RL) framework, the agent is external to the environment [14,3]. It is immortal, and its resources are independent of the resources of the environment. In the real world, agents are *embedded* in the environment; *i.e.*, they can be modified and even destroyed by it. Such considerations were partially addressed in our definition of the optimal, self-modifying, universal agent [5], whose *source code* was part of the environment itself, both readable and modifiable by it. A companion paper [6] considers the consequences of doing the same with the agent's *memory of the past* (but not its source code).

In this section, we consider *space-embedded* agents, whose code *and* memory are modifiable by the environment. The space-embedded agent's code is calculated by an infinite computational device (or *oracle*) which yields the full results of the computation immediately and independently of the machine that computes the environment. However, the environment can modify the agent in its entirety—both its memory and its code, which together define the agent's *policy*.

At each time step, the space-embedded agent uses its current policy π_t to produce a *candidate* next policy π'_{t+1}, which is passed to the environment. The environment then produces the agent's actual next policy π_{t+1}, and the optimal agent is therefore defined as:

$$\pi^* := \arg\max_{\pi_1 \in \Pi} V(\pi_1, \lambda) \tag{1}$$

$$V(\pi_t, \overline{ao}_{\prec t}) := \sum_{a_t = \langle a'_t, \pi'_{t+1} \rangle} \pi_t(a_t \mid o'_{t-1}) \sum_{o_t = \langle o'_t, \pi_{t+1} \rangle} \rho(o_t \mid \overline{ao}_{\prec t} a_t) \Big[\gamma_t u(\overline{ao}_{1:t}) + V(\pi_{t+1}, \overline{ao}_{1:t}) \Big] \tag{2}$$

where the time $\tilde{t}$ and memory $\tilde{l}$ limits are not considered for now. Note that while Equation 2 shows the semantics of $a_t = \langle a'_t, \pi'_{t+1} \rangle$, neither a'_t nor π'_{t+1} are used in the equation. In fact, there is no need for an explicit action-observation interaction protocol anymore: the environment can read and write any information, including information about actions and observations, directly into the agent's space π_{t+1}. Thus, Equation (2) can be rewritten in various equivalent forms that have different interpretations:

$$V(\pi_t, \overline{ao}_{\prec t}) := \sum_{a_t} \pi_t(a_t) \sum_{o_t} \rho(o_t \mid \overline{ao}_{\prec t} a_t) \Big[\gamma_t u(\overline{ao}_{1:t}) + V(o_t, \overline{ao}_{1:t}) \Big] \tag{3}$$

$$V(\pi_t, \overline{a\pi}_{\prec t}) := \sum_{a_t} \pi_t(a_t) \sum_{\pi_{t+1}} \rho(\pi_{t+1} \mid \overline{\pi a}_{1:t}) \Big[\gamma_t u(\overline{a\pi}_{1:t} \pi_{t+1}) + V(\pi_{t+1}, \overline{a\pi}_{1:t}) \Big] \tag{4}$$

$$V(\pi_t, \overline{\pi\pi'}_{\prec t}) := \sum_{\pi'_t} \pi_t(\pi'_t) \sum_{\pi_{t+1}} \rho(\pi_{t+1} \mid \overline{\pi\pi'}_{1:t}) \Big[\gamma_t u(\overline{\pi\pi'}_{1:t} \pi_{t+1}) + V(\pi_{t+1}, \overline{\pi\pi'}_{1:t}) \Big] \tag{5}$$

In Equation (3) the action-observation protocol has been removed; additionally, π_{t+1} is shown as an implicit interpretation of o_t from the previous step. Equation (4) differs from Equation (3) in that the environment's output is always interpreted as a policy. Equation (5) then also renames action a as π' to emphasize that the agent and the environment share the agent's memory space. In all of these equations, the alphabets of the actions, observations, and policies are considered to be the same.

It is interesting to note that when the environment contains the agent's code, there is no RL agent that is asymptotically as good as every other agent in all environments: for each agent π, there is an environment $q_{\bar{\pi}}$ that always gives $r_t = 0 \, \forall t$ to that particular agent, and $r_t = 1 \, \forall t$ for all other agents.

With the space-embedded framework, the agent can in principle make predictions about its source code and memory (*e.g.*, that it will be updated by humans, who are part of the environment, to get new sensors or more efficient code). By contrast, neither AIXI, nor $\text{AIXI}_{\tilde{t},\tilde{l}}$, nor the Gödel Machine can make such predictions even in principle.[5]

5.2 Space-Time-Embedded Agents

The space-embedded agent's next action is computed independently from the environment by a separate machine. Its code can be incomputable (*cf.* AIXI [3]), and, unless an explicit time limit $\tilde{t}$ is introduced, is expected to run until completion (regardless how much computation might be involved) before the result is passed to the environment. In the real-world though, the agent cannot have more computing power or memory than what the environment has to offer.

To better model the interaction between the agent and the environment in the real world, we examine the case where the reference machine of the agent (*i.e.*, the computer on which it runs), is a part of the environment, and the agent is *computed by the environment*. Our proposal is not simply for the agent and environment to be computed by the same reference machine, but to actually make the agent be computed *by* the environment itself.

[5] In the case of the Gödel Machine, one could set up a special *protocol* whereby external agents could *propose* a new source code to the machine, possibly along with a proof that this leads to a better expected value. If the machine can verify the proof, it could adopt the new code. But this is a restricted form of external modification.

One specific advantage of this model is that it allows the agent through its actions (and predictions) to optimize not just its policy but also potentially the physical device on which its policy is computed.[6] An additional advantage is that there is no need anymore for a time parameter $\tilde{t}$, since it is the environment that determines how much computation the agent is allowed.

An alteration to Equation (5) describes the agent's computation as performed by ρ (even if ρ is only an estimate of the true environment):

$$\pi^* := \arg\max_{\pi_1 \in \Pi} V(\pi_1)$$

$$V(\overline{\pi\pi'}_{\prec t}\pi_t) := \sum_{\pi'_t} \rho'(\pi'_t \mid \overline{\pi\pi'}_{\prec t}\pi_t) \times \sum_{\pi_{t+1}} \rho(\pi_{t+1} \mid \overline{\pi\pi'}_{1:t}) \Big[\gamma_t u(\overline{\pi\pi'}_{1:t}\pi_{t+1}) + V(\overline{\pi\pi'}_{1:t}\pi_{t+1})\Big] \quad (6)$$

$$V(\overline{\pi\pi'}_{\prec t}\pi_t) := \sum_{\pi'_t\pi_{t+1}} \rho''(\pi'_t\pi_{t+1} \mid \overline{\pi\pi'}'_{\prec t}\pi_t) \Big[\gamma_t u(\overline{\pi\pi'}_{1:t}\pi_{t+1}) + V(\overline{\pi\pi'}_{1:t}\pi_{t+1})\Big], \quad (7)$$

where ρ' is defined appropriately for π', and ρ'' is the one-step combination of ρ and ρ'. Loosely renaming $\pi'_t\pi_{t+1}$ to π_{t+1} and ρ'' back to ρ, we can interpret Equation (7) as merging two interaction steps into a single one, and we obtain the value of the *space-time-embedded* agent:

$$V(\pi_{\prec t}) := \sum_{\pi_t} \rho(\pi_t \mid \pi_{\prec t}) \Big[\gamma_t u(\pi_{1:t}) + V(\pi_{1:t})\Big] \quad . \quad (8)$$

Equation (8) has one remaining problem, which is that for $t = 0$, when nothing yet is known about the environment, the optimal policy may have infinite length. Thus, we need to add a length constraint $\tilde{l}$ on the initial length that the program can have. It is a reasonable parameter for any real-world model. In our world it corresponds to the maximum number of bits that we, as programmers, are ready to use for the initial policy of the agent. Note, however, that after the very first step the actual length of the agent is determined by the computation of the environment, *i.e.*, $\pi_t, t > 1$ need not be of size less than $\tilde{l}$. Therefore, the final definition of the optimal bounded-length space-time-embedded agent is:

$$\boxed{\begin{aligned} \pi^* &:= \arg\max_{\pi_1 \in \Pi^{\tilde{l}}} V(\pi_1) \\ V(\pi_{\prec t}) &:= \sum_{\pi_t \in \Pi} \rho(\pi_t \mid \pi_{\prec t}) \Big[\gamma_t u(\pi_{1:t}) + V(\pi_{1:t})\Big] \end{aligned}} \quad .$$

Although simpler than Legg's definition, this equation has profound implications regarding the nature of the agent. In particular, it precisely represents the goal of those attempting to build an Artificial General Intelligence in our world.

[6] Such effects can be neither predicted nor controlled by AIXI, AIXI$_{\tilde{t},\tilde{l}}$, or the Gödel Machine using their current definition.

A Turing Machine Model. It is convenient to envision the space-time-embedded environment as a multi-tape Turing machine with a special tape for the agent. This tape is used by the environment just like any other working-memory tape. The read and write heads need not be synchronized and there is no external computing device, oracle or special interface for the agent. The agent's tape can be seen as a partial internal state of the environment, which is consistent with the intuition that agents do not have special status in the real world compared to the rest of the environment. This view extends easily to a multi-agent framework.

A Cellular-Automaton Survival Agent. It is also instructive to envision the environment as a cellular automaton (*e.g.*, the Game of Life [1]), in which an agent is represented as a particular set of cells whose initial state is specified by the initial policy π_1 (perhaps an $\sqrt{\tilde{l}} \times \sqrt{\tilde{l}}$ square of cells). The cells surrounding the agent, in possibly unbounded number, may be in any initial state.

As an example, consider a utility function whose value is 1 as long as some critical part of the agent's policy maintains some particular pattern (call it the agent's "heart"), and 0 otherwise. If the pattern can be destroyed by *gliders*[7] coming from outside the initial square of the agent, then the agent must find ways to avoid them. The optimal initial policy π^* is thus the one that maximizes the expected number of time steps that the heart pattern survives. To ensure its survival, the agent may need to learn about and react to its environment in sophisticated and intelligent ways (provided $\tilde{l}$ is large enough).

Note that while the utility and horizon functions are part of the definition of V, and thus are critical to defining π^*, they are not necessarily represented within π^* in any way. Note also that although π_1 is the entire initial state of the agent, as soon as the agent and environment interact, the boundary between them may quickly blur or disappear. Thus the notation π_t for $t > 2$ may be misleading, since it can also be viewed simply as a *window* onto that part of the environment used (by the utility and horizon functions) to assign a value to the agent. Since only the output of ρ can be used by the utility function, ρ can be defined so that $\pi_t, t > 2$ encompasses the entire environment, while π_1 remains limited to $\tilde{l}$ bits. Thus, in the case of a cellular automaton, for example, the agent and its heart pattern may drift away from its original cells; or, alternatively, the utility function may seek to maximize the number of "live" cells (*i.e.*, cells set to 1) in the entire environment.

New Kinds of Questions. The framework for space-time-embedded agents allows formal discussion of a range of questions that could not previously be formulated using the traditional RL framework.

Because the agent is computed by the environment, the agent's choices are the result of the computations made by the environment on the bits defining the agent's policy. Genuine choice is exercised only by those processes (*e.g.*, those programmers) that define the agent's initial program π_1 before interaction with the environment begins. Yet the programmers are also part of the environment, which implies that the agent is generated ultimately as a result of the initial

[7] A "glider" is a repeating pattern that can cross regions of the cellular automaton.

conditions of the environment in a generative process much like Solomonoff's Sequence Prediction [12].

If at some time t some set of bits implements an intelligent agent π, one might wonder by what process this agent was generated or, more precisely, what are the most probable environments that could have generated this set of bits. We do not seek to answer to this question here, but only to point out that the question itself can be discussed within the framework of space-time-embedded agents, in contrast to the traditional RL framework, in which the agent is not generated but is simply assumed to exist from the very first step.[8]

In fact, the framework now allows many questions to be discussed, such as: Who is the agent? (*i.e.*, what part of the global computation defines the identity of the agent; *e.g.*, π_1, u, γ?) What is an agent? (*i.e.*, where is the boundary between the agent and its environment?) What does it mean for an agent to live and to die? (Questions that depend deeply on agent identity and thus the boundary between the agent and the environment.)

To perform any computation, an embedded agent necessarily affects its environment, and thus the mere act of calculating the consequences of its own actions implies a self-referential computation. It may seem we would therefore need to define agents that deal explicitly with the problem of self reference, but our framework instead circumvents this issue entirely by simply computing each agent's value subject to the environment's computational constraints, and then selecting the agent with the highest value. This best agent may, or may not, deal with self reference, but does so to whatever extent is optimal for its environment.

6 Discussion and Conclusion

This paper has proposed a new definition of intelligence within a theoretical framework more realistic than previous definitions. In both Legg's definition of intelligence [4] (based on AIXI [3]) and Russell's bounded-optimality framework [10] (which embraces time and space constraints on the program of the agent), the agent is separated from the environment and therefore immortal.

Space-time-embedded intelligence formally describes agents as components of their environment. Such agents are thus limited to the computational resources (computation time and memory space) provided by their environment, can be fully modified by their environment (*i.e.*, the agents are mortal), and are even computed *by* the environment, making their computation dependent on the dynamics of the environment. Compared with previous definitions of intelligence, the resulting equation is surprisingly simple—deceptively simple, in fact, hiding considerable depth beneath its surface.

Our primary motivation was to find an equation of intelligence that, if solved, might lead to the actual building of real-world, artificially intelligent agents. We believe that this new formalization brings us a step closer to our goal, though we are by no means naive as to the difficulty of its solution. As much as anything else, we greatly lack insight into ρ (*i.e.*, the universe in which we live).

[8] Note, though, that in our companion paper [6] we address the issue of how to assess the probability of some particular memory state generated by an environment.

For example, our universe allows for the sustainability of intelligent agents—patterns that continue through large stretches of space and time (partly thanks to our protective skulls). We hope this work will help us develop a better understanding of the information we will need for eventually solving the equation.

When discussing *narrow AI*, our framework might be regarded as a step backward compared to Russell's definition, since the interaction loop is hidden in the embedding and makes reasoning about the actions of the agents less explicit. But when discussing true *AGI*, our framework captures critical aspects of reality that Russell's definition simplifies away: agents are computed by and can be modified by the environment, and are therefore mortal. Furthermore, these theoretical consequences of inhabiting one's environment turn out to be essential and practical considerations for all the intelligent agents who inhabit our own.

References

1. Conway, J.: The game of life. Scientific American 303(6), 43–44 (1970)
2. Goertzel, B.: Toward a Formal Characterization of Real-World General Intelligence. In: Proceedings of the 3rd Conference on Artificial General Intelligence, AGI 2010, pp. 19–24. Atlantis Press (2010)
3. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer, Berlin (2005)
4. Legg, S.: Machine Super Intelligence. Department of Informatics, University of Lugano (2008)
5. Orseau, L., Ring, M.: Self-Modification and Mortality in Artificial Agents. In: Schmidhuber, J., Thórisson, K.R., Looks, M. (eds.) AGI 2011. LNCS (LNAI), vol. 6830, pp. 1–10. Springer, Heidelberg (2011)
6. Orseau, L., Ring, M.: Memory Issues of Intelligent Agents. In: Bach, J., Goertzel, B., Iklé, M. (eds.) AGI 2012. LNCS (LNAI), vol. 7716, pp. 219–231. Springer, Heidelberg (2012)
7. Ortega, D.A., Braun, P.A.: Information, Utility and Bounded Rationality. In: Schmidhuber, J., Thórisson, K.R., Looks, M. (eds.) AGI 2011. LNCS (LNAI), vol. 6830, pp. 269–274. Springer, Heidelberg (2011)
8. Ring, M., Orseau, L.: Delusion, Survival, and Intelligent Agents. In: Schmidhuber, J., Thórisson, K.R., Looks, M. (eds.) AGI 2011. LNCS (LNAI), vol. 6830, pp. 11–20. Springer, Heidelberg (2011)
9. Robinson, H.: Dualism. In: The Stanford Encyclopedia of Philosophy. Winter 2011 edn. (2011)
10. Russell, S.J., Subramanian, D.: Provably Bounded-Optimal Agents. Perspective 2, 575–609 (1995)
11. Schmidhuber, J.: Ultimate Cognition à la Gödel. Cognitive Computation 1(2), 177–193 (2009)
12. Solomonoff, R.J.: A Formal Theory of Inductive Inference. Part I. Information and Control 7(1), 1–22 (1964)
13. Stoljar, D.: Physicalism. In: The Stanford Encyclopedia of Philosophy. Fall 2009 edn. (2009)
14. Sutton, R., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
15. Zvonkin, A.K., Levin, L.A.: The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. Russian Mathematical Surveys 25(6), 83–124 (1970)