Avoiding Unintended AI Behaviors

Bill Hibbard
MIRI Research Associate
Emeritus Senior Scientist, SSEC, University of Wisconsin-Madison

Abstract

Artificial intelligence (AI) systems too complex for predefined environment models and actions will need to learn environment models and to choose actions that optimize some criteria. Several authors have described mechanisms by which such complex systems may behave in ways not intended in their designs. This paper describes ways to avoid such unintended behavior. For hypothesized powerful AI systems that may pose a threat to humans, this paper proposes a two-stage agent architecture that avoids some known types of unintended behavior. For the first stage of the architecture this paper shows that the most probable finite stochastic program to model a finite history is finitely computable, and that there is an agent that makes such a computation without any unintended instrumental actions.

Hibbard, Bill. 2012. "Avoiding Unintended AI Behaviors." In *Artificial General Intelligence: 5th International Conference, AGI 2012, Oxford, UK, December 8–11, 2012. Proceedings*, edited by Joscha Bach, Ben Goertzel, and Matthew Iklé, 107–116. Lecture Notes in Artificial Intelligence 7716. New York: Springer. doi:10.1007/978-3-642-35506-6_12

1. Introduction

Some scientists expect artificial intelligence (AI) to greatly exceed human intelligence during the twenty-first century (Kurzweil 2005). There has been concern about the possible harmful effect of intelligent machines on humans since at least Asimov's (1942) Laws of Robotics. More recently there has been interest in the ethical design of AI (Hibbard 2001; Bostrom 2003; Goertzel 2004; Yudkowsky 2004; Hibbard 2008; Omohundro 2008; Waser 2010, 2011; Muehlhauser and Helm 2012). Much of this work is closely reasoned but not mathematical. An AAAI Symposium on Machine Ethics (Anderson, Anderson, and Armen 2005) included some mathematical papers but focused almost exclusively on machine ethics in the context of the logic-based approach to AI rather than the learning-based approach (although one paper studied using feed forward neural networks to learn to classify moral decisions).

Hutter's (2005) theory of universal AI significantly advanced the mathematical theory of rational agents. This work defines a mathematical framework for agents and environments, in which agents learn models of their environments and pursue motives defined by utility functions to be maximized. Schmidhuber (2009) analyzed agents that had the option to modify their own code and concluded that they would not choose to modify their utility function in any way incompatible with their current utility function. In his work, the mathematics of rational agents was applied to a question relevant to whether AI would satisfy the intentions of its human designers.

The AGI-11 conference included three papers (Orseau and Ring 2011; Ring and Orseau 2011; Dewey 2011) that employed the mathematics of rational agents to analyze ways that AI agents may fail to satisfy the intentions of their designers. Omohundro (2008) and Bostrom (2012) described secondary AI motivations that are implied by a wide variety of primary motivations and that may drive unintended behaviors threatening humans. This paper proposes approaches for designing AI agents to avoid unintended behaviors, continuing the work of Hibbard (2012).

The next section presents a mathematical framework for reasoning about AI agents and possible unintended behaviors. The third section discusses sources of unintended behavior and approaches for avoiding them. The final section is a summary.

2. An Agent-Environment Framework

We assume that an agent interacts with an environment. At each of a discrete series of time steps $t \in \mathbf{N} = \{0, 1, 2, \ldots\}$ the agent sends an action $a_t \in A$ to the environment and receives an observation $o_t \in O$ from the environment, where A and O are finite sets. We assume that the environment is computable and we model it by programs $q \in Q$, where Q is some set of programs. Let $h = (a_1, o_1, ..., a_t, o_t) \in H$ be an interaction history where H is the set of all finite histories, and define $|h| = t$ as the length of the history h. Given a program $q \in Q$ we write $o(h) = U(q, a(h))$, where $o(h) = (o_1, ..., o_t)$ and $a(h) = (a_1, ..., a_t)$, to mean that q produces the observations o_i in response to the actions a_i for $1 \leq i \leq t$ (U is a program interpreter). Given a program q the probability $\rho(q) : Q \to [0, 1]$ is the agent's prior belief that q is a true model of the environment. The prior probability of history h, denoted $\rho(h)$, is computed from $\rho(q)$ (two ways of doing this are presented later in this section).

An agent is motivated according to a *utility function* $u : H \to [0, 1]$ which assigns utilities between 0 and 1 to histories. Future utilities are discounted according to a *geometric temporal discount* $0 < \gamma < 1$ (Sutton and Barto 1998). The value $v(h)$ of a possible future history h is defined recursively by:

$$v(h) = u(h) + \gamma \max_{a \in A} v(ha), \tag{1}$$

$$v(ha) = \sum_{o \in O} \rho(o|ha)\, v(hao). \tag{2}$$

Then the agent π is defined to take, after history h, the action:

$$\pi(h) := a_{|h|+1} = \arg\max_{a \in A} v(ha). \tag{3}$$

For Hutter's (2005) universal AI, Q is the set of programs for a deterministic prefix universal Turing machine (PUTM) U (Li and Vitányi 1997). The environment may be non-deterministic in which case it is modeled by a distribution of deterministic programs. The prior probability $\rho(q)$ of program q is $2^{-|q|}$ where $|q|$ is the length of q in bits, and the prior probability of history h is given by:

$$\rho(h) = \sum_{q:o(h)=U(q,a(h))} \rho(q). \tag{4}$$

Hutter's universal AI is a *reinforcement-learning* agent, meaning that the observation includes a reward r_t (i.e., $o_t = (\hat{o}_t, r_t)$) and $u(h) = r_{|h|}$. Hutter showed that his universal AI maximizes the expected value of future history, but it is not finitely computable.

As Hutter discussed, for real world agents single finite stochastic programs (limited to finite memory, for which the halting problem is decidable) such as Markov decision processes (MDPs) (Hutter 2009b; Sutton and Barto 1998) and dynamic Bayesian networks (DBNs) (Hutter 2009a) are more practical than distributions of PUTM programs for defining environment models. Modeling an environment with a single stochastic program rather than a distribution of deterministic PUTM programs requires a change to the way that $\rho(h)$ is computed in (4). Let Q be the set of all programs (these are bit strings in some language for defining MDPs, DBNs, or some other finite stochastic programming model), let $\rho(q) = 4^{-|q|}$ be the prior probability of program q where $|q|$ is the length of q in bits ($4^{-|q|}$ to ensure that $\sum_{q \in Q} \rho(q) \leq 1$ since program strings in Q are not prefix-free), and let $P(h|q)$ be the probability that q computes the history h.[1] Note $\rho(q)$ is a discrete distribution on individual program strings, not a measure on bit strings in the sense of Li and Vitányi (1997, p. 243). Then given a history h_0, the environment model is the single program that provides the most probable explanation of h_0, that is the q that maximizes $P(q|h_0)$. By Bayes theorem:

$$P(q|h_0) = \frac{P(h_0|q)\,\rho(q)}{P(h_0)}. \tag{5}$$

$P(h_0)$ is constant over all q so can be eliminated. Thus we define $\lambda(h_0)$ as the most probable program modeling h_0 by:

$$\lambda(h_0) := \operatorname*{arg\,max}_{q \in Q} \; P(h_0|q)\,\rho(q). \tag{6}$$

Proposition 1. Given a finite history h_0 the model $\lambda(h_0)$ can be finitely computed.

Proof. Given $h_0 = (a_1, o_1, ..., a_t, o_t)$ let q_{tl} be the program that produces observation o_i at time step i for $1 \leq i \leq t$ (such a finite "table-lookup" program can be written as an MDP, DBN, or in any other finite stochastic programming language with equivalent expressiveness) and let $n = |q_{tl}|$. Then, since the behavior of q_{tl} is deterministic, $P(h_0|q_{tl})\rho(q_{tl}) = 1 \times 4^{-n} = 4^{-n}$ so $P(h_0|\lambda(h_0))\rho(\lambda(h_0)) \geq 4^{-n}$. For any program q with $|q| > n$, $P(h_0|q)\rho(q) < 1 \times 4^{-n} = 4^{-n}$ so $\lambda(h_0) \neq q$. Thus one algorithm for finitely computing $\lambda(h_0)$ is an exhaustive search of the finite number of programs q with $|q| \leq n$ (there is no need here to consider the set of all programs that implement a given MDP). □

Given an environment model $q_0 = \lambda(h_0)$ the following can be used for the prior probability of an observation history h in place of (4):

$$\rho(h) = P(h|q_0). \tag{7}$$

According to current physics our universe is finite (Lloyd 2002). For finite environments, agents based on (6) and (7) are as optimal as those based on (4). Their prior

[1] $P(h|q)$ is the probability that q produces the observations o_i in response to the actions a_i for $1 \leq i \leq |h|$. For example let $A = \{a, b\}$, $O = \{0, 1\}$, $h = (a, 1, a, 0, b, 1)$ and let q generate observation 0 with probability 0.2 and observation 1 with probability 0.8, without any internal state or dependence on the agent's actions. Then the probability that the interaction history h is generated by program q is the product of the probabilities of the 3 observations in h: $P(h|q) = 0.8 \times 0.2 \times 0.8 = 0.128$. If the probabilities of observations generated by q depended on internal state or the agent's actions, then those would have to be taken into account.

probabilities better express algorithmic complexity if finite stochastic programs are expressed in an ordinary procedural programming language restricted to have only static array declarations, to have no recursive function definitions, and to include a source of truly random numbers.

3. Unintended AI Behaviors

Dewey (2011) employed the mathematics of rational agents to argue that reinforcement-learning agents will modify their environments so that they can maximize their utility functions without accomplishing the intentions of human designers. He discussed ways to avoid this problem with utility functions not conforming to the reinforcement-learning definition. Ring and Orseau (2011) argued that reinforcement-learning agents will self-delude, meaning they will choose to alter their own observations of their environment to maximize their utility function regardless of the actual state of the environment. In Hibbard (2012) I demonstrated by examples that agents with utility functions defined in terms of the agents' environment models can avoid self-delusion, and also proved that under certain assumptions agents will not choose to self-modify.

3.1. Model-Based Utility Functions

Given an environment model $q_0 = \lambda(h_0)$ derived from interaction history h_0, let Z be the set of finite histories of the internal states of q_0. Let h' be an observation and action history *extending* h_0 (defined as: h_0 is an initial subsequence of h'). Because q_0 is a stochastic program it may compute a set $Z_{h'} \subseteq Z$ of internal state histories that are *consistent* with h' (defined as: q_0 produces $o(h')$ in response to $a(h')$ when it follows state history $z' \in Z_h$) and terminating at time $|h'|$. Define $u_0(h', z')$ as a utility function in terms of the combined histories h' and $z' \in Z_{h'}$. The utility function $u(h')$ for use in (1) can be expressed as a sum of utilities of pairs (h', z') weighted by the probabilities $P(z'|h', q_0)$ that q_0 computes z' given h':

$$u(h') := \sum_{z' \in Z_{h'}} P(z'|h', q_0)\, u_0(h', z'). \tag{8}$$

The demonstration that the examples in Hibbard (2012) do not self-delude does not contradict the results in Ring and Orseau (2011), because model-based utility functions are defined from the history of observations and actions whereas the utility functions of self-deluding agents are defined from observations only. Self-delusion is an action by the agent and prohibiting actions from having any role in the utility function prevents the agent from accounting for its inability to observe the environment in evaluating the consequences of possible future actions. Agents can increase utility by sharpening the probabilities in (8), which implies a need to make more accurate estimates of the state of their environment model from their interaction history. And that requires that they continue to observe the environment. But note this logic only applies to stochastic environments because, once an agent has learned a model of a deterministic environment, it can predict environment state without continued observations and so its model-based utility function will not place higher value on continued observations.

3.2. Unintended Instrumental Actions

Omohundro (2008) and Bostrom (2012) describe how any of a broad range of primary AI motivations will imply secondary, unintended motivations for the AI to preserve its own existence, to eliminate threats to itself and its utility function, and to increase its own efficiency and computing resources. Bostrom discusses the example of an AI whose primary motive is to compute pi and may destroy the human species due to implied instrumental motivations (e.g., to eliminate threats and to increase its own computing resources).

Omohundro uses the term "basic AI drives" and Bostrom uses "instrumental goals." In the context of our agent-environment framework they should instead be called "unintended instrumental actions" because in that context there are no implied drives or goals; there are only a utility function, an environment model, and actions chosen to maximize the sum of future discounted utility function values. We might think that instrumental goals apply in some different framework. But von Neumann and Morgenstern (1944) showed that any set of value preferences that satisfy some basic probability axioms can be expressed as a utility function. And the framework in (1)–(3) maximizes the expected value of the sum of future discounted utility function values (Hay 2005) so any other framework is sub-optimal for value preferences consistent with the probability axioms. The utility function expresses the agent's entire motivation so it is important to avoid thinking of unintended instrumental actions as motivations independent of and possibly in conflict with the motivation defined by the utility function. But unintended instrumental actions can pose a risk, as in Bostrom's example of an AI whose motivation is to compute pi.

In analyzing the risk of a given unintended instrumental action, such as increasing the agent's physical computing resources by taking them from humans, the question is whether it increases a given utility function. If the utility function increases with the increasing health and well-being of humans, then it will not motivate any unintended instrumental action that decreases human health and well-being.

3.3. Learning Human Values

Several approaches to human-safe AI (Yudkowsky 2004; Hibbard 2008; Waser 2010; Muehlhauser and Helm 2012) suggest designing intelligent machines to share human values so that actions we dislike, such as taking resources from humans, violate the AI's motivations. However, Muehlhauser and Helm (2012) survey psychology literature to conclude that humans are unable to accurately write down their own values. Errors in specifying human values may motivate AI actions harmful to humans.

An analogy with automated language translation suggests an approach to accurately specifying human values. Translation algorithms based on rules written down by expert linguists have not been very accurate, but algorithms that learn language statistically from large samples of actual human language use are more accurate (Russell and Norvig 2010). This suggests that statistical algorithms may be able to learn human values. But to accurately learn human values will require powerful learning ability. This creates a chicken-and-egg problem for safe AI: learning human values requires powerful AI, but safe AI requires knowledge of human values.

A solution to this problem is a first stage agent, here called π_6, that can safely learn a model of the environment that includes models of the values of each human in the environment. An AI agent is defined by (1)–(3), (6) and (7), but (6) can be used alone to define the agent π_6 that learns a model $\lambda(h_0)$ from history h_0. In order for π_6 to learn an accurate model of the environment the interaction history h_0 in (6) should include agent actions, but for safety π_6 cannot be allowed to act. The resolution is for its actions to be made by many safe, human-level surrogate AI agents independent of π_6 and of each other. Actions of the surrogates include natural language and visual communication with each human. The agent π_6 observes humans and their interactions with the surrogates and physical objects in an interaction history h_0 for a time period set by π_6's designers, and then reports an environment model to the environment.

Proposition 2. The agent π_6 will report the model $\lambda(h_0)$ to the environment accurately and will not make any other, unintended instrumental actions.

Proof. Actions, utility function and predictions are defined in (1)–(3) and hence are not part of π_6. However, π_6 has an implicit utility function, $P(h_0|q)\rho(q)$, and an implicit action, reporting $\lambda(h_0) = \arg\max_{q \in Q} P(h_0|q)\rho(q)$ to the environment (π_6 also differs from the full framework in that it maximizes a single value of its implicit utility function rather than the sum of future discounted utility function values). The implicit utility function $P(h_0|q)\rho(q)$ depends only on h_0 and q. Since the interaction history h_0 occurs before the optimizing $\lambda(h_0)$ is computed and reported, there is no way for the action of reporting $\lambda(h_0)$ to the environment to affect h_0. So the only way for the agent π_6 to maximize its implicit utility function is to compute and report the most accurate model. Furthermore, while the history h_0 may give the agent π_6 the necessary information to predict the use that humans plan to make of the model $\lambda(h_0)$ that it will report to the environment, π_6 makes no predictions and so will not predict any effects of its report. □

This result may seem obvious but given the subtlety of unintended behaviors it is worth proving. The agent π_6 does not act in the world; that's the role of the agent described in the next section.

3.4. An AI Agent That Acts in the World

Muehlhauser and Helm (2012) describe difficult problems in using human values to define a utility function for an AI. This section proposes one approach to solving these problems, using the model $q_0 = \lambda(h_0)$ learned by π_6 as the basis for computing a utility function for use in (1)–(3) by a "mature" second stage agent π_m that acts in the environment (i.e., π_m does not use the surrogate agents that acted for π_6).

Let D_0 be the set of humans in the environment at time $|h_0|$ (when the agent π_m is created), defined by an explicit list compiled by π_m's designers. Let Z be the set of finite histories of the internal states of q_0 and let $Z_0 \subseteq Z$ be those histories consistent with h_0 that terminate at time $|h_0|$. For z' extending some $z_0 \in Z_0$ and for human agent $d \in D_0$ let $h_d(z')$ be the history of d's interactions with its environment, as modeled in z', and let $u_d(z')(.)$ be the values of d expressed as a utility function, as modeled in z'. The observations and (surrogate) actions of π_6 include natural language communication with each human, and π_m can use the same interface via A and O to the model q_0 for conversing in natural language with each model human $d \in D_0$. In order to evaluate $u_d(z')(h_d(z'))$, π_m can ask model human d to express a utility value between 0 and 1 for $h_d(z')$ (i.e., d's recent experience). The model q_0 is stochastic so define Z'' as the set of histories extending z' with this question and terminating within a reasonable time limit with a response $w(z'')$ (for $z'' \in Z''$) from model human d expressing a utility value for $h_d(z')$. Define $P(z''|z')$ as the probability that q_0 computes z'' from z'. Then $u_d(z')(h_d(z'))$ can be estimated by:

$$u_d(z')(h_d(z')) = \frac{\sum_{z'' \in Z''} P(z''|z')\, w(z'')}{\sum_{z'' \in Z''} P(z''|z')}. \tag{9}$$

This is different than asking human d to write down a description of his or her values, since here the system is asking the model of d to individually evaluate large numbers of histories that d may not consider in writing down a values description.

An average of $u_d(z')(h_d(z'))$ over all humans can be used to define $u_0(h', z')$ and then (8) can be applied to $u_0(h', z')$ to define a model-based utility function $u(h')$ for π_m. However, this utility function has a problem similar to the unintended behavior of reinforcement learning described by Dewey (2011): π_m will be motivated to modify the utility functions u_d of each human d so that they can be more easily maximized.

This problem can be avoided by replacing $u_d(z')(h_d(z'))$ by $u_d(z_0)(h_d(z'))$ where $z_0 \in Z_0$. By removing the future value of u_d from the definition of $u(h')$, π_m cannot increase $u(h')$ by modifying u_d. Computing $u_d(z_0)(h_d(z'))$ is more complex than asking model human d to evaluate its experience as in (9). The history h_0 includes observations by π_6 of physical objects and humans, and π_m can use the same interface via O to the model q_0 for observing physical objects and humans at the end of state history z'. And surrogate actions for π_6 define an interface via A and O to the model q_0 that π_m can use for communicating visually and aurally with model human d after state history z_0. These interfaces can be used to create a detailed interactive visualization and hearing of the environment over a short time interval at the end of state history z', to be explored by model human d at the end of state history z_0 (i.e., two instances of the model q_0, at state histories z' and z_0, are connected via their interfaces A and O using visualization logic). Define Z'' as a set of histories extending z_0 with a request to model human d to express a utility value between 0 and 1 for $h_d(z')$, followed by an interactive exploration of the world of z' by model human d, and finally terminating within a reasonable time limit with a response $w(z'')$ (for $z'' \in Z''$) from model human d expressing a utility value for the world of z'. Define $P(z''|z_0)$ as the probability of that q_0 computes z'' from z_0. Then $u_d(z_0)(h_d(z'))$ can be estimated by:

$$u_d(z_0)(h_d(z')) = \frac{\sum_{z'' \in Z''} P(z''|z_0)\, w(z'')}{\sum_{z'' \in Z''} P(z''|z_0)}. \tag{10}$$

The utility function should be uniform over all histories $h_d(z')$ but $u_d(z_0)(.)$ varies over different $z_0 \in Z_0$. However (10) does not assume that z' extends z_0 so use the probability $P(z_0|h_0, q_0)$ that q_0 computes z_0 given h_0 (as in Section 3.1) to define:

$$u_d(h_0)(h_d(z')) := \sum_{z_0 \in Z_0} P(z_0|h_0, q_0)\, u_d(z_0)(h_d(z')). \tag{11}$$

Now define a utility function for agent π_m as a function of z':

$$u_0(h', z') := \frac{\sum_{d \in D_0} f(u_d(h_0)(h_d(z')))}{|D_0|}. \tag{12}$$

Here $f(.)$ is a twice differentiable function over $[0, 1]$ with positive derivative and negative second derivative so that low $u_d(h_0)(h_d(z'))$ values have a steeper weighting slope than high $u_d(h_0)(h_d(z'))$ values. This gives π_m greater utility for raising lower human utilities, helping those who need it most. For any h' extending h_0 a model-based utility function $u(h')$ for agent π_m can be defined by the sum in (8) of $u_0(h', z')$ values from (12).

In the absence of an unambiguous way to normalize utility functions between agents, we assume that the constraint of utility values to the range $[0, 1]$ provides normalization. In order to account for humans' evaluations of the long term consequences of π_m's actions, π_m should use a temporal discount γ close to 1.

The set D_0 of humans in (12) is the set at time $|h_0|$ rather than at the future time of z'. This avoids motivating π_m to create new humans whose utility functions are more easily maximized, similar to the use of $u_d(z_0)(h_d(z'))$ instead of $u_d(z')(h_d(z'))$.

The agent π_m will include (6) and should periodically (perhaps at every time step) set h_0 to the current history and learn a new model q_0. Should it also update D_0 (to those judged to be human by consensus of members of D_0 at the previous time step), define a new set Z_0, relearn the evolving values of humans via (10) and (11), and redefine $u(h')$ via (12) and (8)? To stay consistent with the values of evolving humans and the birth of new humans, π_m should redefine its utility function periodically. But there could also be risks in allowing the utility function of π_m to evolve. The proofs that agents will not modify their utility functions (Schmidhuber 2009; Hibbard 2012) do not apply here since those proofs assumed that redefining the utility function is an action of the agent to be evaluated according to the current utility function using (1)–(3). Here the definition of π_m could simply include periodic redefinition of its utility function without regard to its optimality according to the current utility function.

I cannot offer a proof that π_m avoids all unintended behaviors. And there are problems with the estimate of human values in (10): the model human is visualizing rather than experiencing first person, and human values do not conform to the preconditions for utility functions. But every sane human assigns nearly minimal value to human extinction so the utility function $u(h')$ for agent π_m will assign nearly minimal value to human extinction. Actions motivated by this utility function must increase its value, so no unintended instrumental action will cause human extinction. Similarly π_m will not make any unintended instrumental actions abhorred by a large majority of humans.

4. Discussion

This paper has addressed several sources of unintended AI behavior and discussed ways to avoid them. It has proposed a two-stage agent architecture for safe AI. The first stage agent, π_6, learns a model of the environment that can be used to define a utility function for the second stage agent, π_m. This paper shows that π_6 can learn an environment model without unintended behavior. And the design of π_m avoids some forms of unintended behavior. However, this paper does not prove that π_m will avoid all unintended behaviors. It would be useful to find computationally feasible implementations for the definitions in this paper.

While the proposed two-stage agent architecture is intrusive and manipulative, that seems likely in any scenario of super-human AI. The key point is whether the AI's utility function is democratic or serves the interests of just a few humans. An appealing goal is to find an AI architecture that gives humans the option to minimize their interaction with the AI while protecting their interests.

This paper addresses unintended AI behaviors. However, I believe that the greater danger comes from the fact that above-human-level AI is likely to be a tool in military and economic competition between humans and thus have motives that are competitive toward some humans.

Acknowledgments

I would like to thank Luke Muehlhauser for helpful discussions.

References

Anderson, Michael, Susan Leigh Anderson, and Chris Armen, eds. 2005. *Machine Ethics: Papers from the 2005 AAAI Fall Symposium.* Technical Report, FS-05-06. AAAI Press, Menlo Park, CA. http://www.aaai.org/Library/Symposia/Fall/fs05-06.

Asimov, Isaac. 1942. "Runaround." *Astounding Science-Fiction,* March, 94–103.

Bostrom, Nick. 2003. "Ethical Issues in Advanced Artificial Intelligence." In *Cognitive, Emotive and Ethical Aspects of Decision Making in Humans and in Artificial Intelligence,* edited by Iva Smit and George E. Lasker, 12–17. Vol. 2. Windsor, ON: International Institute for Advanced Studies in Systems Research / Cybernetics.

———. 2012. "The Superintelligent Will: Motivation and Instrumental Rationality in Advanced Artificial Agents." In "Theory and Philosophy of AI," edited by Vincent C. Müller. Special issue, *Minds and Machines* 22 (2): 71–85. doi:10.1007/s11023-012-9281-3.

Dewey, Daniel. 2011. "Learning What to Value." In Schmidhuber, Thórisson, and Looks 2011, 309–314.

Goertzel, Ben. 2004. "Universal Ethics: The Foundations of Compassion in Pattern Dynamics." Working paper, October 25. Accessed January 16, 2013. http://www.goertzel.org/papers/UniversalEthics.htm.

Hay, Nicholas James. 2005. "Optimal Agents." B.Sc thesis, University of Auckland. http://www.cs.auckland.ac.nz/~nickjhay/honours.revamped.pdf.

Hibbard, Bill. 2001. "Super-Intelligent Machines." *ACM SIGGRAPH Computer Graphics* 35 (1): 13–15. http://www.siggraph.org/publications/newsletter/issues/v35/v35n1.pdf.

———. 2008. "The Technology of Mind and a New Social Contract." *Journal of Evolution and Technology* 17 (1): 13–22. http://jetpress.org/v17/hibbard.htm.

———. 2012. "Model-Based Utility Functions." *Journal of Artificial General Intelligence* 3 (1): 1–24. doi:10.2478/v10229-011-0013-5.

Hutter, Marcus. 2005. *Universal Artificial Intelligence: Sequential Decisions Based On Algorithmic Probability.* Texts in Theoretical Computer Science. Berlin: Springer. doi:10.1007/b138233.

———. 2009a. "Feature Dynamic Bayesian Networks." In *Proceedings of the Second Conference on Artificial General Intelligence 2009,* edited by Ben Goertzel, Pascal Hitzler, and Marcus Hutter, 67–72. Amsterdam: Atlantis.

———. 2009b. "Feature Reinforcement Learning: Part I. Unstructured MDPs." *Journal of Artificial General Intelligence* 1 (1): 3–24. doi:10.2478/v10229-011-0002-8.

Kurzweil, Ray. 2005. *The Singularity Is Near: When Humans Transcend Biology.* New York: Viking.

Li, Ming, and Paul Vitányi. 1997. *An Introduction to Kolmogorov Complexity and Its Applications.* 2nd ed. Graduate Texts in Computer Science. New York: Springer.

Lloyd, Seth. 2002. "Computational Capacity of the Universe." *Physical Review Letters* 88 (23): 237901. doi:10.1103/PhysRevLett.88.237901.

Muehlhauser, Luke, and Louie Helm. 2012. “The Singularity and Machine Ethics.” In *Singularity Hypotheses: A Scientific and Philosophical Assessment,* edited by Amnon Eden, Johnny Søraker, James H. Moor, and Eric Steinhart. The Frontiers Collection. Berlin: Springer.

Omohundro, Stephen M. 2008. “The Basic AI Drives.” In *Artificial General Intelligence 2008: Proceedings of the First AGI Conference,* edited by Pei Wang, Ben Goertzel, and Stan Franklin, 483–492. Frontiers in Artificial Intelligence and Applications 171. Amsterdam: IOS.

Orseau, Laurent, and Mark Ring. 2011. “Self-Modification and Mortality in Artificial Agents.” In Schmidhuber, Thórisson, and Looks 2011, 1–10.

Ring, Mark, and Laurent Orseau. 2011. “Delusion, Survival, and Intelligent Agents.” In Schmidhuber, Thórisson, and Looks 2011, 11–20.

Russell, Stuart J., and Peter Norvig. 2010. *Artificial Intelligence: A Modern Approach.* 3rd ed. Upper Saddle River, NJ: Prentice-Hall.

Schmidhuber, Jürgen. 2009. “Ultimate Cognition *à la* Gödel.” *Cognitive Computation* 1 (2): 177–193. doi:10.1007/s12559-009-9014-y.

Schmidhuber, Jürgen, Kristinn R. Thórisson, and Moshe Looks, eds. 2011. *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3–6, 2011. Proceedings.* Lecture Notes in Computer Science 6830. Berlin: Springer. doi:10.1007/978-3-642-22887-2.

Sutton, Richard S., and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction.* Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press.

Von Neumann, John, and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior.* 1st ed. Princeton, NJ: Princeton University Press.

Waser, Mark R. 2010. “Designing a Safe Motivational System for Intelligent Machines.” In *Artificial General Intelligence: Proceedings of the Third Conference on Artificial General Intelligence, AGI 2010, Lugano, Switzerland, March 5–8, 2010,* edited by Eric B. Baum, Marcus Hutter, and Emanuel Kitzelmann, 170–175. Advances in Intelligent Systems Research 10. Amsterdam: Atlantis. doi:10.2991/agi.2010.21.

———. 2011. “Rational Universal Benevolence: Simpler, Safer, and Wiser than ‘Friendly AI.’” In Schmidhuber, Thórisson, and Looks 2011, 153–162.

Yudkowsky, Eliezer. 2004. *Coherent Extrapolated Volition.* The Singularity Institute, San Francisco, CA, May. http://intelligence.org/files/CEV.pdf.