On the computational complexity of action evaluations

C. J. Reynolds
MIT Media Laboratory
20 Ames St,Room E15-120f
Cambridge,Massachusetts 02139
+1-617-253-8628
carsonr@media.mit.edu

Abstract

From the standpoint of computational complexity different ethical positions are analyzed, with the eventual idea of implementing them as actual systems. For each of the action evaluation strategies suggested by hedonism, consquentialsm, and deontologism a satisfaction condition is defined. The time complexity involved in meeting these satisfaction conditions is then analyzed. Hedonism is found to be less computationally complex than consequentialism and deontologism (which are of equivilent complexity classes). In support of these analyses, a literature review discussing questions of computability, ethical computability, and computational complexity is provided. In addition, it is argued that computers need not be ethical in the same manner that humans are ethical.

Keywords

Computer Ethics, Evaluative Complexity, Satisfaction Condition, Computational Complexity, Ethical Computability.

ATTRIBUTES OF COMPUTABLE ETHICS

The goal of this paper is to describe hedonism, consequentialism, and deontologism in terms of their computational complexity. Specifically, each of these stances is analyzed in terms of the computational cost in terms of time for an agent α considering a set of actions Φ and foreseeing the consequences to a horizon τ units in the future.

This paper will begin by informally discussing differences between the potential performance of ethical reasoning by computers and the actual performance by humans, using an analogy to mathematics. The main point in this aside is to sidestep the criticism that I believe human and computers will perform ethics in the same manner.

A short literature review that discusses existing work describing computers performing ethical reasoning tasks is then provided. The limitations of this work is discussed for comparison with the analysis provided below. Following upon this will be a cursory discussion of what it means for something to be computable. This will involve a discussion of the Church-Turing thesis.

Building from this, I will describe computational complexity. There are a number of different approaches and metrics for complexity. This paper will review some of the more major before adopting a loose upper bound on time-complexity as the approach for analysis.

Borrowing from Kagan's notion of evaluative focal points, I will discuss evaluative complexity. This

is defined to be the amount of time the agent must spend in using a strategy to assess an evaluative focal point. This paper will chose actions as the evaluative focal point for which complexity is computed.

Satisfaction conditions will then be described. A satisfaction condition for a given ethical philosophy is a criteria that describes the steps an agent would need to perform to decide if a action φ was ethically acceptable. Satisfaction conditions will be provided for lazy egotistical hedonism, act consequentialism, and Kantian deontologism.

The computational complexity of evaluating each of these satisfaction conditions will be considered for progressively more informative world states. This will be performed by increasing the number of actions φ in Φ, agents α in a world ω, and time units τ towards infinity.

This paper will conclude by arguing that deontology and consequentialism are of the same time-complexity class. Furthermore, some informal ideas about reasoning with finite resources are discussed.

ANALOGY TO MATHEMATICS

While humans and computers arrive at the same answers, they use different methods to perform mathematical operations like addition. It is beyond the scope of this paper to provide a description of how humans do arithmetic; but it can be argued that humans and computers come to be able to do arithmetic by different processes: humans are taught, while computers are designed. Humans are often taught about addition by repeating simple tasks like comparing amounts, counting physical objects, learning to count up or count down, and adding physical objects (Banfill, 2002). After increasing proficiency at these simple tasks, students of arithmetic are taught progressively more complex tasks like subtraction, multiplication, and division.

In contrast, computer are designed to use specialized arithmetic units to efficiently perform arithmetic operations (Zimmerman, 1997). Early computers performed addition using a combination of NOT, AND, OR digital logic elements formed into full adder units that can be chained together to do addition on binary numbers (Avizienis, 1966). More recent approaches like ripple, bypass, and carry-skip, and variable block adders perform elaborate operations in order to minimize the speed of addition in computers (Oklobdzija, 1985).

Human mathematicians also make use of different methods to prove unsolved conjectures than the efficient heuristic search and logic chaining used by computational theorem provers. Humans working on unsolved conjectures use a wide variety of techniques including common sense reasoning, relating the unsolved conjecture to familiar problems, talking with other mathematicians, building geometric models, and looking for patterns like invariants (Polya, 1968). In contrast, computational theorem provers start from a set of axioms and mechanically using rules of inference to arrive at a proof (Kalman, 2001). In short, somewhat like arithmetic, computers and humans arrive at solved proofs using somewhat different methods.

I should be careful to note that neither of these points provide argumentation that computers could not perform mathematics in the same way that people do (for instance by closely modeling the process of a human learning mathematics or proof solving). It is only argued that existing computers use different approaches to perform arithmetic and provide proofs for unsolved conjectures than human do.

By analogy then, I argue that some varieties of ethical behavior for computers could be different to existing ethical behavior in humans. Furthermore, I argue that the ways in which humans could implement ethical behavior for computers could be different than the ways we teach one another of about ethics.

ETHICS COMPUTABILITY LITERATURE REVIEW

In "Is Ethics Computable," Moor discusses the potential of ethical reasoning by computers (Moor, 1995). He begins by discussing Bentham's utilitarianism as a possible routeine for computing the good tendancy or evil tandancy of an action. He continues by discussing a thought experiment involving creating a robot capable of ethical decision making. He briefly sketches the problems involved in creating a non-harmful robot, maximizing/minimizing robot, heuristic robot, and fair robot. He describes a method for testing the ethical decision making of computers as well as some objectios to this. He concludes by suggesting that right now computers can work with humans in an arrangement dividing the labor of ethical decision making.

In "The First Law of Robotics" Weld and Etzioni discuss means of making autonomous agents "safe" by giving them the ability to plan and perform actions in the presence of possible "harm" . A variety of researchers have considered the topic including: Axelrod (1984), Russel (1991), Danielson (1992), Eichmann (1994), Allen (2000), Wallach (2002), and Floridi (2004). However, many of these researchers shied away from providing actual implementations in favor of discussing the computability of ethics in an abstract form.

Of more relevance is the Harwood's "Ethics and Artificial Life" (Harwood, 2001). This document describes various ethical positions in terms of pseudo and actual programs that operate in an extremely simple world. His agents can perform only two actions (help or hinder). However, he ran actual simulations looking at the overall outcome of sets of agents using different philosophies to choose actions.

WHAT IS COMPUTABILITY?

In mathematics, the property of computability separates real numbers that can be deduced using mechanical procedures from others that are undecidable. For instance, the value of the 51,539,599,999th digit of π is something that can be computed, using a mechanical process codified into an algorithm (Lopez-Ortiz, 1998). In contrast Chaitin's constant is non-computable but is definable (Weisstein, 2004).

Another, more rigorous, way of defining computability is to say a computable number is a that "can be computed to any number of digits desired by a Turing machine" (Copeland, 2002). A Turing machine is (according to the Church-Turing thesis) a mechanism capable of carrying out every mechanical computation. A method M is effective or "'mechanical' just in case:

1. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);

2. M will, if carried out without error, produce the desired result in a finite number of steps;

3. M can (in practice or in principle) be carried out by a human being unaided by any machinery save paper and pencil;

4. M demands no insight or ingenuity on the part of the human being carrying it out (Copeland, 2002).

A familiar example of a method M is the use of a truth table to test for tautologies. With a pencil and paper a non-ingenious student of philosophy can compute in a finite number of steps whether a given proposition is a tautology or not.

WHAT IS COMPLEXITY?

There are many different notions of complexity used in the study of complexity theory. Some common

and practical notions of complexity are time-complexity and space-complexity. The time-complexity of a task is roughly analogous to the answer of the question "how long" a method M would take given an input string of finite length. A bewildering array of classes have been proposed for different types of computing complexity (Aaronson, 2004). Some of the more famous are P (decision problems that can be solved in polynomial time) and NP (decision problems whose potential answers can be checked in polynomial time) (Wikipedia, 2004).

One more specific notion of complexity commonly used by computer scientists is Big O notation. Essentially, Big O analysis works by increasing the input string length for a given M and seeing which factor dominates the time to halt the computation. As an example, if an algorithm M take $n^3 + 5n^2 + 2$ steps to halt, where n is the length of the input string, then it is said to be of $O(n^3)$ since that factor dominates the growth of the time complexity as n increases. Big O notation provides an asymptotic upper bound for the time complexity of a method M (Black, 2004).

SATISFYING THE LAZY HEDONIST

Hedonism is an ethical position espoused by the followers Aristippus of Cyrene that is often a straw man used for comparison. In response to Socrates' "What is the good?" the hedonist replies "pleasure." It should be noted that there are a number of approaches to Hedonism. Epicurus's particular variety did not advocate pleasure without consideration of resulting pain. He sought to avoid excess, seeking pleasure in the long term .

However, the Hedonist for whom I will develop a simple satisfaction condition is a lazy hedonist. Of course, as a hedonist, the lazy hedonist pursues pleasure. But I will define lazy hedonists as individuals whose desire for satisfaction is such that they he will accept any action which brings more pleasure (Moore, 2004).

To compute the time complexity of the Lazy Hedonist, let me propose a simple world model. Let us say there is a world ω. In ω there is a single action φ with associated outcome (change in state); and a single agent α (our Lazy Hedonist). At the outset the Lazy Hedonist has a state ζ (relating to whether their circumstance is good or bad) of 0.5. The agent is unconcerned about the welfare of others.

The hedonist's satisfaction condition is = if $\varphi > \zeta$. That is, any action that brings about a better state is accepted. Since there is only one "step" to the evaluation of the satisfaction condition, we would say that the computational complexity of the Lazy Hedonist is O(1). Meaning, for any input, the agent will perform one step.

Of course this ω is ridiculously simple. Let's consider some complications: the introduction of another agent, and another action. Let N denote the number of agents (2 in this case) and M denote the number of actions available to each agent (again 2 in this case).

If N=2 and M=2, what is the computational complexity of a hedonist deciding if they're satisfied? Since a hedonist is not concerned with the welfare of others, the addition of the second agent makes no difference. However, the existence of another action does make a difference. If the lazy hedonist will consider all actions then they must evaluate M potential actions. This leads to a time complexity of O (M), since there are M steps involved in deciding if the lazy hedonist is satisfied. However, one could allow that the lazy hedonist will be satisfied by the first action that meets the satisfaction condition. Presuming a uniform distribution of good and bad outcomes between 0 and 1 inclusive and a starting state of 0.5, the average case would be M/2 evaluations, which is still dominated by M, leaving the time complexity O(M).

SATISFYING THE CONSEQUENTIALIST

Consequentialism is a family of ethical positions that share the common attribute of weighing the good or bad of an action depending on the consequences of the action. Utilitarianism, as espoused by

Bentham and Mill, considered the consequences in terms of the pleasure and pain brought about by the action. A utilitarian consequentialist seeks "the greatest happiness of the greatest number" (Everett, 2002).

Mill states utilitarianism is the "creed which accepts as the foundation of moral 'utility' or the 'greatest happiness principle.'" Here Mill to define happiness as "intended pleasure and the absence of pain". However, diverging from Hedonism seeks happiness not just for α but for all agents in ω.

There are more or less strict notions of happiness for the greatest number. One can have limited varieties of consequentialism that seek only the happiness of a limited set of individuals. The most pure notion of happiness involves considering the consequences of an action on the happiness of all individuals who have ever lived, are living, or will ever live. I will start by developing a satisfaction condition for the limited case.

If N=2 and M=3, then α must consider the impact of φing on all the agents in ω. So for all 3 actions, the agent must consider the effects on themselves as well as the other agent. This leads to six assessments which is N * M. So we may say that the cost of a limited utilitarian examining the effect of φ is O(MN).

But what of the action's repercussions for future moments of time and future decisions? One might argue, as Lemand did, that any action in a consequentialist view could have "massive causal ramification." That it might touch, even in a diluted form on future individuals. Hitler's "great grandmother" holds some share in the pleasure and pain brought about by her actions, however minuscule (Lenman, 2000).

This call for introducing a new variable L, that denotes the limit of our causal ramification analysis. The limited consequentialism described above has an L=1, meaning the consequences of actions are only considered for the present which we might call a 1st-order limited consequentialist. Let us consider a 2nd-order limited consequentialist. In examining an action the agent must still consider the effect of each φ in M on all N agents, but now, for each action, the α must also consider how future available actions effect future N.

If N=2 and M=3 and L=2, then a second-order limited consequentialist would end up evaluating 6^1 actions in the present as well as 6^2 actions in the future after a time step τ. A proof by induction gives a computational complexity of $O((MNL)^L) + O((MN)^{L-1} + ... + O(MN)^1$. This sequence collapses into $((MN)^{L+1})/((MN-1)-1)$. Big O notation allows us to summarize this as $O(MN^L)$, since that factor dominates the growth of the time complexity.

So an Lth order limited consequentialist performs $O(MN^L)$ individual evaluations. This is exponentially more costly in L than the hedonists approach. In short, a consequentialist is more considerate than a lazy hedonist, exponentially more so, depending on how far they wish to examine the consequences of their actions.

If time goes on until ∞ (supposing time is divided into equal intervals for each evaluation of the satisfaction condition) then a pure hedonist is saddled with evaluating an infinite number of causal ramifications. The question of when a pure consequentialist would finishing determining if they were satisfied is undecidable.

The above analysis does not take into account that greater or fewer agents enter ω. It also assumes a fixed number of actions. Both the number of actions and agents could change over time. This would lead to a dynamic sequence of N_1 ... N_L and M_1 ... M_L. I will not develop this more complex case.

SATISFYING THE DEONTOLOGIST

Deontology is a tradition of ethics stemming from the agent's duty to perform or not perform certain actions. Kant's categorical imperative provides one prominent example of this ethical position.

Presumably, a deontological agent of the Kant school, when evaluating an action φ considers not just its utility, but:

1. α's intent in performing φ

2. Whether an φ is in accordance with the moral law

3. Whether performing φ can be consistent with "rules that we would be willing to have followed by all people in all circumstances" (Kant, 1785).

It is possible that the above things might be roughly equivalent, but Kant's description of the process is somewhat verbose, so I'll assume that an agent needs to perform at least the each of the above.

The satisfaction function for a Kantian deontologist would be roughly as follows. For each φ 3 separate evaluations need to be performed. Because we need to consider intent with respect to other agents, rules followed by "all people," and the moral law (which is universal). So each φ in M must be considered from the perspective of what would happen if all α in φ were to perform it.

This would be roughly analogous to all of the evaluations a limited consequentialist would consider with a few key differences. A Kantian is not impelled to find the ideal action in the same way utilitarians are. This means that once a φ that meets these criteria is found, it may be performed (an agent can work on their own non-ideal personal projects).

So if an agent were consider the case where N=1, M=1, and L=1, The agent would perform 3 actions, which collapses to O(1). With increasing numbers of actions uniformly distributed between good and bad, the agent will need to consider, on average M/2 actions as they relate to N individuals. So in the present case, deontology preforms on average NM/2 evaluations which is O(MN). If the casual ramification horizon is increased then the agent must grapple with the same exponential growth of outcomes to evaluate in ω. So using this world model we find that deontologism is $O(MN^L)$. If L is large (namely the agent considers rules of behavior that would be valid for all time) then this becomes intractable and the decision of whether the agent is satisfied never ends.

COMPARISONS

Using ω as a very far-from-realistic model for the world, we can see some interesting things about the asymptotic time-complexity of various ethical positions. Informally we can say that a lazy hedonist does less work and is less considerate than a Consequentialist or Deontologist. It appears that consequentialists and deontologists have ethical strategies that are roughly equivalent, namely $O(MN^L)$. This is a "computationally hard" task that an agent with limited resources will have difficulty performing. It is of the complexity task of NP or more specifically EXPTIME. Furthermore, as the horizon for casual ramifications moves towards infinity the satisfaction function for both consequentialism and deontologism become intractable.

ACKNOWLEDGMENTS

I would like to thank Caspar Hare and Rosalind Picard for providing encouragement for this approach. This work is supported in part by the MIT Media Lab Things That Think consortium.

REFERENCES

Aaronson S. (2004) The Complexity Zoo, http://www.complexityzoo.com

Allen, C., Varner, G. and Zinser, J. (2000) Prolegomena to Any Future Artificial Moral Agent. *Journal of Experimental and Theoretical Artificial Intelligence*, **12**, 251-261.

Avizeienis A. (1966) Arithmetic Microsystems for the Synthesis of Function Generators. *Proceedings of the IEEE*, **54**, 255-258.

Axelrod R.M. (1984) *The evolution of cooperation*. Basic Books, New York.

Banfill J. (2002) Kindergarten Math Lessons, http://www.aaamath.com/kindergarten.html

Black P.E. (2004) big-O notation, http://www.nist.gov/dads/HTML/bigOnotation.html

Copeland B.J., (2002) The Church-Turing Thesis, http://plato.stanford.edu/entries/church-turing/

Danielson P. (1992) *Artificial Morality: Virtuous Robots for Virtual Games*. Routledge, London.

Eichmann D. (1994) Ethical web agents. *Computer Networks and ISDN Systems* , **3**, 3-13.

Everett G. (2002) Utilitarianism , http://www.victorianweb.org/philosophy/phil1.html

Floridi, L. and Sanders, J.W. (2004) Mapping the Foundationalist Debate. *Minds and Machines* , **14 (3)**, 349-379.

Harwood, W. (2001) Ethics and Artificial Life. MSc Thesis, University of Oxford.

Kalman, J.A. (2001) *Automated Reasoning with OTTER*. Rinton Press, Paramus, NJ.

Kant, I. and Sher, G. (1984) Morality and Rationality. *Moral Philosophy*. 385-405. Wadsworth , Belmont, CA.

Lenman, J. (2000) Consequentialism and Cluelessness. *Philosophy and Public Affairs*, **29(2)**, 134-171.

Lopez-Ortiz, A. (1998) How to compute digits of À, http://db.uwaterloo.ca/~alopez-o/math-faq/node38.html

Moor, J.H. (1995) Is Ethics Computable? *Metaphilosophy*, **26(1)**, 1-21.

Moore, A. (2004) Hedonism, http://plato.stanford.edu/entries/hedonism/

Oklobdzija, V.G. and Barnes E.R. (1985) Some Optimal Schemes For ALU Implementation In VLSI Technology. *Proceedings of the 7th Symposium on Computer Arithmetic (ARITH-7)*. 2-8.

Polya, G. (1986) *Mathematics and Plausible Reasoning.* Princeton University Press, Princeton, NJ.

Russell, S.J. and Wefald, E. (1991) Principles of Metareasoning. *Proceedings of KR'89: Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, San Mateo, CA, USA.

Wallach, W. (2002) Robot Morals: Creating an Artificial Moral Agent (AMA) , http://www.transhumanism.org/tv/2003usa/panelaiethics.htm

Weisstein, E.W. (2004) Chaitin's Constant, http://mathworld.wolfram.com/ChaitinsConstant.html

Weld, D. and Etzioni, O. (1994) The first law of robotics (a call to arms). *Proceedings of The 12th National Conference on Artificial Intelligence*. 1042-1047.

Wikipedia (2004) Complexity classes P and NP, http://en.wikipedia.org/wiki/Complexity_classes_P_and_NP

Zimmerman, R. (1997) Computer Arithmetic: Principles, Architectures, and VLSI Design , http://www.iis.ee.ethz.ch/~zimmi/arith_lib.html

BIOGRAPHY

Carson Reynolds is a doctoral candidate in the Affective Computing group at the MIT Media Laboratory. He holds a Master of Science from the Massachusetts Institute of Technology and a Bachelor of Science in Technical Communication with a Minor in Philosophy from the University of Washington at Seattle. His work has been discussed in the Washington Times, the German news weekly Focus, and online in Wired, Engadget, Smart Mobs, and Slashdot. Currently Carson is researching applied ethics, affective sensors, and physiological interfaces for games.