Full presentation: Reasoning about machine ethics

Swarat Chaudhuri and Moshe Y. Vardi

We are entering an era where "intelligent" machines have unprecedented power over humans, including, sometimes, the literal power of life and death. Machine learning algorithms are increasingly used to classify humans as suspected terrorists or deserving recipients of financial aid. While military drones are currently flown by human operators, discussions about fully autonomous robotic soldiers are afoot. Once released on our streets, self-driving cars will almost certainly face circumstances where they just cannot avoid harming a human.

It is thus imperative that the new generations of intelligent machines be not just efficient or functional but also *ethical*. If we can prove that a machine has behaved unethically, we can hold liable the operators and designers of the machine. But when can we consider a program to be ethical, and what are the design principles that we ought to follow to ensure ethical program behavior? The emerging field of *machine ethics* [2, 10, 1, 4] explores these questions.

Machine ethics has received a significant amount of interest over the last decade, primarily from ethicists and artificial intelligence experts. Efforts here range from deliberations about the extent to which machines should be given ethical autonomy [1], to classification of AI agents into ethical categories [10], to logic-based methodologies that can guide engineering [5, 11], to concrete algorithms for ethical decision making [9]. Conspicuously missing from the area, however, are contributions by software engineering and programming language researchers.

We believe that this is unfortunate for two reasons. First, the goal of machine ethics, at the end, is to guarantee that programs behave according to certain rigorous (moral and ethical) requirements. The area would thus seem to be a natural target for automated formal reasoning about programs.

Secondly, machine ethics can be a source of interesting problems for the formal methods community. Most recent work in software analysis is motivated by applications in system software (such as operating system code). Machine ethics, in contrast, involves reasoning about knowledge, choices, obligations, etc. in high-level decision-making.

Intelligent programs Reasoning about ethics is arguably only sensible for "intelligent" programs, but when is a program "intelligent"? There is no one answer to this question, but a reasonable requirement is that the program be able to adapt to new circumstances and learn new modes of behavior. For instance, a natural model for such a program is a Markov Decision Process (MDP): a dynamical system where probabilistic and nondeterministic transitions are permitted and where transitions are associated with quantitative *rewards*. Nondeterminism in an MDP can model the decision-making capacities of a program and parameters that are unknown at the time of design. The probabilistic transitions can be used to model uncertainty in the program's inputs. The rewards would associate a value on the outcome of each action. The operational semantics of the program would replace the nondeterministic choice either with determinism or with probabilistic choice, thus producing a concrete (possibly randomized) algorithm.

Ethical programs In philosophy, an ethical framework is a set of guidelines that an agent follows while making ethically significant decisions. For instance, a robot that follows a utilitarian ethics would want to maximize the aggregate well-being of every human in its environment. As computer scientists, we may view these guidelines as forming a *specification*, given either as constraints on the decision-making behavior of the program or as a reference program. A program is then ethical if it satisfies or simulates this specification.

We have some choices in defining a specification of a program's ethics:

- Our specifications can be formalizations of traditional ethical frameworks — for example utilitarian, Kantian, or Rawlsian ethics. Note that developing such a formalization is a challenging task, as traditional ethical systems are often defined rather abstractly.
- We could define the specification as "whatever an average human follows." For instance, to define the ethics of a self-driving car, we could train a statistical model using observations about how a population of humans drives their cars. The downside of this definition is that it sets the ethics of the typical human as an upper bound on machine machines. Given that humans suffer from weaknesses such as racial prejudice, short-termism, and the instinct to survive at all costs, this choice seems arbitrary.

Now we give some examples of concrete formal methods problems concerning machine ethics.

Reasoning about optimal strategies Consider a robot deployed in a battlefield that is required to take lethal actions. A utilitarian ethics demands that at any program state, the right action for the robot is the one that produces maximal strategic benefit while causing minimal harm to civilians. However, the calculation of harm and benefit cannot be based on purely local criteria, as a locally optimal decision may not be globally optimal.

Thus, the problem of enforcing utilitarian ethics amounts to solving a quantitative formal reasoning problem where the goal is to prove that the program takes an optimal sequence of actions while ensuring certain basic boolean criteria [6] (this definition of optimality presumes a notion of rewards associated with program actions). Given that the environment will have to be modeled probabilistically in most realistic settings, the reasoning problem would presumably also have to be stochastic.

Reasoning about fairness Many research efforts in the formal methods community study reasoning about security and data privacy. But what about the *fairness* of decisions made by a program? Specifically, let us consider programs in charge of allocating a set of resources (for example, financial aid). Each principal has a set of fields that are morally relevant to the resource allocation problem (for example, academic ability), and a set of fields that are not (for example, membership in a political party). Rawlsian ethics [12], when applied here, would aim to guarantee that if P_1 and P_2 are principals that are approximately equivalent in morally relevant attributes, then P_1 and P_2 receive similar outcomes. Note that this requirement allows for affirmative action; it's just that the criteria for affirmative action are to be included among morally relevant fields.

Dwork et al. have recently posed [8] the question of building classifiers that are fair in this sense. They show that the algorithmic ideas needed to design fair classifiers resemble that needed for differentially private data release mechanisms. A natural question for formal methods researchers is to systematically prove the fairness of a program. We note that this question is quite similar to the question of reasoning about *robustness*, which has received attention in the verification community of late [7].

Logics for ethics How do we *compose* a set of ethical obligations for a program? This question necessitates the study of logics of ethics and obligations. One traditional choice here is *deontic logic*, or the class of modal logics permitting formulas $O(p)$, meaning "The agent is obligated to ensure p".

It is well-known [3] that logics of obligation can suffer from paradoxes — for example, what if a drone in a battlefield is obligated to ensure p, but it is impossible to ensure p? Is the drone still required to fulfill its other obligations? The answer should presumably be yes. However, note that this is a departure from classical logic, where the impossibility to meet p would amount to an inconsistency, rendering moot every other consideration. Designing logics that overcome such conceptual issues (possibly using a quantitative weighting of various obligations) can be an interesting challenge for programming language researchers.

References

[1] Colin Allen, Wendell Wallach, and Iva Smit. Why machine ethics? *Intelligent Systems, IEEE*, 21(4):12–17, 2006.

[2] Michael Anderson and Susan Leigh Anderson. Machine ethics: Creating an ethical intelligent agent. *AI Magazine*, 28(4):15, 2007.

[3] Lennart Åqvist. Deontic logic. In *Handbook of philosophical logic*, pages 605–714. Springer, 1984.

[4] Nick Bostrom and Eliezer Yudkowsky. The ethics of artificial intelligence. *Draft for Cambridge handbook of artificial intelligence*, 2011.

[5] Selmer Bringsjord, Konstantine Arkoudas, and Paul Bello. Toward a general logicist methodology for engineering ethically correct robots. *Intelligent Systems, IEEE*, 21(4):38–44, 2006.

[6] Swarat Chaudhuri, Martin Clochard, and Armando Solar-Lezama. Bridging boolean and quantitative synthesis using smoothed proof search. In *POPL*, 2014.

[7] Swarat Chaudhuri, Sumit Gulwani, Roberto Lublinerman, and Sara Navidpour. Proving programs robust. In *ESEC/FSE*, pages 102–112. ACM, 2011.

[8] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *ITCS*, pages 214–226, 2012.

[9] Bruce M McLaren. Computational models of ethical reasoning: Challenges, initial steps, and future directions. *Intelligent Systems, IEEE*, 21(4):29–37, 2006.

[10] James H Moor. The nature, importance, and difficulty of machine ethics. *Intelligent Systems, IEEE*, 21(4):18–21, 2006.

[11] Thomas M Powers. Prospects for a kantian machine. *Intelligent Systems, IEEE*, 21(4):46–51, 2006.

[12] John Rawls. *A theory of justice*. Belknap Press, 1999.