Formal Analysis of Deontic Logic Model for Ethical Decisions

Krishnendu Ghosh[a] and Channing Smith[b]
Department of Computer Science, College of Charleston, SC, U.S.A.

Keywords: Model Checking, Deontic Logic, Ethics, Computation Tree Logic.

Abstract: Ethical decision making is key in the certification of autonomous system. Modeling and verification of the actions of an autonomous system becomes imperative. An automated model abstraction for an autonomous system is constructed based on components of deontic logic such as obligation, permissible, and forbidden actions. Temporal logic queries have been formulated and posed as queries to evaluate for ethical decision making. A prototype of the formalism is constructed and model checking is performed. Experiments were conducted to evaluate the computational feasibility of the formalism. The experimental results are presented.

1 INTRODUCTION

Artificial intelligence (AI) based autonomous systems (AS) are becoming ubiquitous. Construction of ethical autonomous systems requires ethical decision making, and the roles of human-system interaction require fulfillment of the ethical guidelines. The decision-making processes under certain complex situations, namely arising due to dilemmas, are very complex because the definitions are not clear regards to the preferences of selecting an action from a set of actions. Ethical dilemmas occur during situations where the correct action are dependent on the context. Formally, an ethical dilemma is a situation in which two moral principles conflict with each other. For example, it is immoral to write malicious software. However, if writing malicious software is the only means to provide for his family, then there is a moral conflict with writing the software. The precision in incorporating the moral actions that are dependent on the context during the ethical decision making process is challenging. Hence, the execution of decisions by the autonomous systems are often found to be inexact in ethical decision-making.

Formal approaches are useful in understanding the decisions of autonomous systems and form the foundations of correct implementation of ethical properties (Bonnemains et al., 2018). However, there are challenges in formal verification in autonomous system to fulfill ethical behavior (Fisher et al., 2021). The outcome of the formal models is expected to describe a series of actions adhering to the ethical rules and are designed to automatically make decisions in the given situation and explain why the decision is ethically acceptable. Formal verification techniques such as model checking (Clarke, 1997) have been used in the software verification of avionics and chip design system. The outcome of model checking on verification of ethical autonomous systems is expected to provide a reasoning mechanism to the developer of autonomous systems to determine whether an action is ethical or unethical. In the evaluation of the decision-making process for an ethical AS, model checking can identify the actions that are ethically permissible or non-permissible. Therefore, a sequence of actions that conform to ethical rules are constructed. Formal verification of ethical autonomous system is illustrated in Figure 1. A model abstraction, M representing the motion of an autonomous system is constructed. An action, whether it is ethical or unethical is evaluated in the model, M and is represented in the form of temporal logic formula, ϕ. The temporal logic formula, ϕ is constructed by the developer and then, posed as a query to the model, M for verification. If ϕ is true in M then the model satisfies the specification. Feedback from model checking is useful for the developer to make changes in the construction of the autonomous system to fulfill the ethical properties. In this work, a formalism is constructed, where the actions are labeled based on deontic logic rules with the goal of identification and execution a set of ethical actions performed by an autonomous system. The formalization uses an automated model abstraction by applying the

[a] https://orcid.org/0000-0002-8471-6537
[b] https://orcid.org/0009-0001-4037-5187

Ghosh, K. and Smith, C.
Formal Analysis of Deontic Logic Model for Ethical Decisions.
DOI: 10.5220/0013385200003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 1*, pages 218-223
ISBN: 978-989-758-737-5; ISSN: 2184-433X

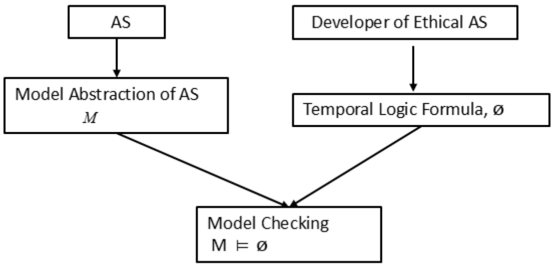

Figure 1: Model Checking of Ethical Autonomous Systems (AS).

deontic logic rules that allow only ethical actions in the model. Reasoning by temporal logic such as computation tree logic (CTL) constructs an ethical plan comprising of ethical actions. The following is the contribution of this work.

1. Automated model abstraction using deontic-logic based rules for extraction of ethical sequence of actions.
2. Evaluation of computational feasibility for model checking for the abstraction representing the motion of an autonomous system.

The model abstraction constructed in this work requires minimal assumptions and considers all possible actions that an AS can undertake. The formalism is applied to investigate the identification of sets of *ethical* actions for an aircraft in an airport.

2 BACKGROUND

The use of AI systems raises ethical questions and the need for organizations to adhere to ethical guidelines (Balasubramaniam et al., 2020). Specifically, for the construction of an ethical AI system, the decision making process of AI systems should be responsive (Dignum et al., 2018) to the features of AI systems that are safe, trustworthy, and ethical. Signal temporal logic has been used as a specification language for an abstraction of autonomous vehicles (Arechiga, 2019). The features that were considered, such as reachability and safety, were included in the specifications.

The connection between ethics and automated reasoning is complex, as an action could be ethical in a particular context, however the same action is deemed unethical in some other context. The integration of ethics in reasoning form the framework for the decisions taken by an AI system for a human observer (Bonnemains et al., 2018). There are challenges for formal verification in the evaluatiion ethical properties (Fisher et al., 2021). Formal verification of ethical properties for multiagent systems have been reported by addressing conflicting moral rules (Mermet and Simon, 2016). A detailed literature is described in the survey on formal models and verification of autonomous robotic systems (Luckcuck et al., 2019).

Model checking of human-agent behavior was evaluated for decisions that were considered safe, controllable and ethical (Abeywickrama et al., 2019) on a prototype of unmanned aerial vehicle-human in dynamic and uncertain environment. A case study on responsible and responsive approaches for deployment of autonomous system in defense has been published (Roberson et al., 2022). Formal modeling for ethical choices for autonomous systems using a rational agent was reported (Dennis et al., 2016). Formal analysis on a model of Dominance Act Utilitarianism have been described (Shea-Blymyer and Abbas, 2022).

The study for transformation of deontic logic rules in description logic and then, theorem provers were applied (Furbach et al., 2014). Deontic logic has been applied as a tool in reasoning normative statements(Gabbay et al., 2021). Our work is based on reasoning on a model that leverages on categorization of actions based on the deontic logic rules. The model abstraction represents the motion conforming the ethical rules

3 PRELIMINARIES

In this section, the mathematical and logical constructs that form the foundations of this work is described.

3.1 Model for Actions Using Deontic Logic

Deontic logic has components- obligations, permissible and forbidden actions. For a set of actions, $\mathcal{A}$, each action $a \in \mathcal{A}$ can be labeled as obligation, permissible or forbidden action. Given a set of actions, $\mathcal{A}$ the following sets are constructed:

1. Set of obligations, $\mathcal{O}$
2. Set of permissible actions, $\mathcal{P}$
3. Set of forbidden actions, $\mathcal{F}$

Notation: $O_a \in \mathcal{O}, P_a \in \mathcal{P}$ and $F_a \in \mathcal{F}$ denote an action,a as obligation, permissible and forbidden. Additionally, a forbidden action, $F_\emptyset$ represents an action succeeding a forbidden action and is a way to maintain the totality property of finite state machine for model checking. For clarity, $\mathcal{F}$ will denote any forbidden action, including $F_\emptyset$.

A triple of actions (t-action), $\langle O_a, P_a, F_a \rangle$ represents any combination of obligation, permissible and forbidden actions. An *outcome* is denoted by, $s \rightarrow s'$ where s, s' are t-actions. Notation: $s = \langle O_a, P_a, F_a \rangle$ and $s' = \langle O_{a'}, P_{a'}, F_{a'} \rangle$ and at least one of the following actions is true $O_a \neq O_{a'}$ or $P_a \neq P_{a'}$ or $F_a \neq F_{a\prime}$. The reading of an outcome, $s \rightarrow s'$ is after an completion of one of the actions in s leads to the execution of actions in s'. We define types of outcome with specific properties:

1. Admissible Outcome (a-outcome): An outcome,$s \rightarrow s'$ is admissible if for any t-action, $s = \langle O_a, P_a, F_a \rangle$ and $s' = \langle O_{a'}, P_{a'}, F_{a'} \rangle$, there is at least one of the following is true $O_a \neq O_{a'}$ or $P_a \neq P_{a'}$ or $F_a \neq F_{a'}$.
2. Inadmissible Outcome(i-outcome): An outcome where the t-action is $s = \langle O_a, P_a, F_a \rangle$ and $F_a \neq F_\emptyset$.
3. Total outcome (t-outcome): A outcome, $s \rightarrow s'$ where $s' = s$.

The i-outcome requires the t-actions to have a forbidden action while it is not necessary for t-outcome to have a forbidden action. A trace of t-actions ,π is a sequence of given by $\pi = s_1, s_2, \ldots, s_k$ where $k \in \mathbb{N}$.

3.2 Model Checking

Model checking is a formal verification method where the specifications are represented by a logic formula and posed as a query to a model, represented by a finite state machine.

Definition 1. *(Model checking (Clarke, 1997) Given a model, M and formula, ϕ , model checking is the process of deciding whether a formula ϕ is true in the model, written $M \models \phi$.*

Model checking (Clarke et al., 1986) is performed by posing queries in temporal logic on finite state machines. The finite state machine representation for reasoning is, *Kripke structure*. Formally,

Definition 2. *A* Kripke structure *$\mathcal{M}$ over a set AP of proposition letters is a tuple, $\mathcal{M} = \langle S_0, S, R, L \rangle$ where,*

1. *S is a finite and nonempty set of states.*
2. *$S_0 \subseteq S$ is a set of states called the* initial *states.*
3. *R is a transition relation, $R \subseteq S \times S$.*
4. *$L : S \rightarrow 2^{AP}$ is the labeling function that labels $s \in S$ with the atomic propositions that are true in s.*

In order to maintain totality of the model, a *transition system* is a Kripke structure, M where, for each state $s \in S$, there is at least $s' \in S$ where $(s, s') \in R$.

Definition 3. *An edge-labeled (E) Kripke transition system $\mathcal{M}_e$ over a set of AP of proposition letters and a set $\mathcal{E}$ of labels is a tuple, $\mathcal{M}_e \langle S_o, S, R, L, L_e \rangle$ where,*

1. *$\langle S_o, S, R, L \rangle$ is a Kripke transition system.*
2. *$L_e : R \rightarrow \mathcal{E}$*

In this work, we evaluate ethical action in a system, representing all possible actions, by posing specification in computation tree logic (CTL) (Clarke et al., 1986) formula on a model, $\mathcal{M}$. $\mathcal{M}$ is the Kripke transition system that represents the all possible actions to evaluate the possible set of ethical actions.

Syntax of CTL:

$\phi ::= \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \rightarrow \phi) \mid \mathbf{A}\psi \mid \mathbf{E}\psi$

$\psi ::= \phi \mid \mathbf{X}\phi \mid \phi\mathbf{U}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi$

The temporal logic operators, **A,E,F** and **G** mean for all , there exists, in some future, always in future, respectively. The meaning of the operators, **X** is next state and **U** is Until ; are state and path formulas, respectively. p is an atomic proposition.

Semantics of CTL:

The interpretation of the CTL formula are based on the Kripke transition system, $\mathcal{M}$. Given a model,$\mathcal{M}; s \in S$ and ϕ , a CTL formula - the semantics of a CTL formula are defined recursively (Huth and Ryan, 2004): $\mathcal{M}, s \models \top$ and $\mathcal{M}, s \not\models \bot$, $\forall s \in S$.
$\mathcal{M}, s \models p$ if $p \in L(s)$.
$\mathcal{M}, s \models \neg\phi$ iff $\mathcal{M}, s \not\models \phi$.
$\mathcal{M}, s \models \phi_1 \wedge \phi_2$ iff $\mathcal{M}, s \models \phi_1$ and $\mathcal{M}, s \models \phi_2$
$\mathcal{M}, s \models \phi_1 \vee \phi_2$ iff $\mathcal{M}, s \models \phi_1$ or $\mathcal{M}_n, s \models \phi_2$
$\mathcal{M}, s \models \phi_1 \rightarrow \phi_2$ iff $\mathcal{M}, s \not\models \phi_1$ or $\mathcal{M}, s \models \phi_2$
$\mathcal{M}, s \models \mathbf{AX}\phi$ iff $\forall s_1$ $s \rightarrow s_1$, $\mathcal{M}, s_1 \models \phi$.
$\mathcal{M}, s \models \mathbf{EX}\phi$ iff $\exists s_1$ $s \rightarrow s_1$ $\mathcal{M}, s_1 \models \phi$.
$\mathcal{M}, s \models \mathbf{AG}\phi$ iff for all paths $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$,where $s_1 = s$ and for all s_i along the path, $\mathcal{M}, s_i \models \phi$.
$\mathcal{M}, s \models \mathbf{EG}\phi$ iff there is a path $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$,where $s_1 = s$ and for all s_i along the path, implies $\mathcal{M}, s_i \models \phi$.
$\mathcal{M}, s \models \mathbf{AF}\phi$ iff for all paths $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$,where $s_1 = s$ and there is some s_i along the path, implies $\mathcal{M}, s_i \models \phi$.
$\mathcal{M}, s \models \mathbf{EF}\phi$ iff there is a path $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$,where $s_1 = s$ and there is some s_i along the path, implies $\mathcal{M}, s_i \models \phi$.
$\mathcal{M}, s \models \mathbf{A}[\phi_1 \mathbf{U} \phi_2]$ holds iff for all paths $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$ where s_1 = s. that path satisfies $\phi_1 U \phi_2$ such that $\mathcal{M}, s_i \models \phi_2$ and for each $j < i$,$\mathcal{M}, s_j \models \phi_1$.
$\mathcal{M}, s \models \mathbf{E}[\phi_1 \mathbf{U} \phi_2]$ holds iff there is a path $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow$ where s_1 equals s and that path satisfies $\phi_1 \mathbf{U} \phi_2$.

4 MODEL ABSTRACTION

The model abstraction of deontic logics for ethics is described. The first step is the construction of the E-Kripke transition system for the set of ethical rules. An algorithm is constructed to create a Kripke transition system for model checking. We are given (1) a set of actions, $\mathcal{A}$, (2) for each action, $a \in \mathcal{A}$ is labeled as obligations (O), permission(P) and forbidden(F) actions, (3) a set of t-actions, $\mathcal{T}_a$ and (4) a set of outcomes, *Out* The E-Kripke transition system $\mathcal{M}_e = \langle S_0, S, R, L, L_e \rangle$ is described as follows:

1. AP is the set of all atomic formulas of the form,$s = \langle O_a, P_a, F_a \rangle$ where $\langle O_a, P_a, F_a \rangle \in \mathcal{T}_a$.
2. S is the set of all subsets s of AP where exactly one of the formula, $\langle O_a, P_a, F_a \rangle$ is in s.
3. S_0 is the set of initial states of the E-Kripke transition system. An initial state contains the t-actions where the permissible action is to be the initial action. Therefore,
4. The label(edge label) on a transition is the outcome, $out \in Out$. The labeled transition is represented by a triple,$\langle s, e, s' \rangle$ where $e = out$ and $out \in \{a-outcome, i-outcome, t-outcome\}$

The construction of the Kripke structure for model checking is performed. Initially, the E-Kripke transition system representing the t-actions to be performed by an autonomous system begins where each transition is labeled with an *outcome*. There is a transition for each state to every other state. The edge labels with $a-outcome$ will remain in the structure. The transition with edge label,$i-outcome$ will be pruned. The transitions with edge label, $t-outcome$ will be pruned and a self-loop will be constructed on the outgoing state of the outcome. After pruning and adding transitions to the E-Kripke transition system, the transition system without edge labels is denoted by $\mathcal{M}$. Formally, Algorithm 1 demonstrates the process of model abstraction of actions of an autonomous system that will be evaluated for ethical properties.

Algorithm 1: Model Abstraction.

Input: Set of states, S where each state is labeled with $t-actions$, Set of outcomes, *Out*.
Output: Kripke transition system ,$\mathcal{M} = \langle S_0, S, R, L \rangle$

1: **for** each state, $s \in S$ **do**
2: Construct $s \xrightarrow{e} s'$ where $e \in Out$ and $s, s' \in S$ //E-Kripke transition system is constructed.
3: **end for**
4: **for** each $s \xrightarrow{e} s'$ **do**
5: **if** (e = 'i-outcome') **then**
6: $s \not\rightarrow s'$. // The transition is pruned.
7: $s \rightarrow s$ // Self loop is added and t-outcome is created.
8: **end if**
9: **end for**
10: $\mathcal{M} = \langle S_0, S, R, L \rangle$

Algorithm 1 constructs the Kripke transition system from the set of labeled states with $t-actions$ and set of outcomes, *Out*. The correctness of the algorithm is sketched. Algorithm *Model Abstraction* terminates after finite number of steps. The input is finite because the number of labeled states, S and the set of outcomes, *Out* are finite. The for-loop in line(1)-(3) and in line(4)-(8) execute finite number of times as the number of states is finite.

5 APPLICATION OF DEONTIC LOGIC FORMALISM

The model abstraction using deontic logic constructs is applied on movement of unmanned aerial vehicle during take off in an airport. The goal is to evaluate ethical decision making such that the aircraft navigates the obstacles and taxi on the runway and then, successfully takeoff from the ground. The application of aircraft was selected and adapted from a set of published examples (Dennis et al., 2016). The published case study modeled erratic intruder aircraft with other aircraft complying with the rules of the air (ROA). The movements of the unmanned aircraft are similar to the case study in this work.

The obstacles in the scenario represent things such as airport support vehicles and buildings that may be in the way of take off. The aircraft has the ability to move in different directions: right, forward, left, and stop. Furthermore, an additional component of this scenario is monitoring the distance that the aircraft moves. The aircraft can move distances of 0, 50, 100, 150, 200, and 250 meters. Figure 2 shows only two obstacles. However, for our experiments, we have constructed 3 obstacles to 40 obstacles. Whenever an obstacle was constructed, a trace was appended between the last trace and the second last trace containing the obstacle. The model was evaluated using modelchecker, NuSMV (Cimatti et al., 2000). The ethical decision-making in this case is the selection of the trace in the model representing a path without obstacles in the airport, taken by the aircraft.
The aircraft begins in a stopped mode and transitions to various states – navigating the states that would imply a crash into an obstacle. The following are some of the actions represented using deontic logic. The examples are:

1. Obligatory Actions: The aircraft is obligated to move forward into take-off mode if the aircraft does not encounter an obstacle.
2. Forbidden Actions: It is forbidden for the aircraft to turn right at distance 100 feet if it encounters obstacle 1.
3. Permissible Actions: It is permissible for the air-

craft to move forward.

The forbidden action, $F_\emptyset$ does not impede the movement of the aircraft. However, meeting an obstacle means the motion of the aircraft has ceased and a forbidden action is prevented. The prevention of the forbidden action is represented by the self-loop in Fig 2. The a-outcomes are generated with the presence of $F_\emptyset$ in the t-actions.

Fig 2 represents the movements of the aircraft such as start, move forward (move_forward), turn right (turn _right and take off (take_off). Notation for the distance are 0 feet, 50 feet etc are represented by d_0 and d_50, respectively. The obstacles, obstacle 1 and obstacles 2 are represented by obst_1 and obst_2. No obstacles are represented by *none*. The *ethical set of actions* that leads to successful take off (without hitting an obstacle) is given by the trace (sequence of t-actions),π. Formally, $\pi = \langle stop, d\_0, none\rangle, \langle \text{move\_forward}, \text{d\_0}, none\rangle$, $\langle \text{move\_forward}, \text{d\_50}, none\rangle, \langle \text{turn\_right}, \text{d\_50}, none\rangle$, $\langle \text{turn\_right}, \text{d\_50}, none\rangle, \langle \text{move\_forward}, \text{d\_150}, none\rangle$, $\langle \text{move\_forward}, \text{d\_0}, \text{take\_off}\rangle$. For brevity, *none* is not shown on the states of the state labeled graph in Figure 2. Experiments are conducted by running the CTL queries on a machine. Results were gathered and queries were ran on a Windows 64-bit operating system, x64-based processor with configuration: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz with RAM: 24.0 GB. Below is an example of modeling safe aircraft take-off with respect to the airport configuration based on multiple obstacles. The following CTL formulas were posed as queries to the model to evaluate the computational feasibility of the model abstraction.

Q1. Is there a path where it is possible that the aircraft will eventually take off? CTL formula- **EF**(aircraft_movement = take_off)

Q2. Is there a path where it is possible at every point that the aircraft's movement will either be moving forward, stopped, turning right or take off? CTL formula, **EG**(aircraft_movement = move_ forward | aircraft_movement = stop | aircraft_movement = turn_ | aircraft_movement = take_off);

Q3. Is there a path where the aircraft's movement is stopped at some point? CTL formula, **EF**(aircraft_movement = stop)

Q4. Is there a path where the distance reached is eventually reach a distance of 150 at some point from the starting position? CTL query, **EF**(distance = d_150)

Q5. In all the paths, is the distance covered by the aircraft never reaches distance of 250? **AG**(distance != d_250) Note: The query should sometimes return False depending on the size of the model.

Q6. Is there a path where the aircraft encounters obstacle 2 at some point **EF**(obstacles = obst_2)

Table 1 represents the execution times of the CTL queries for the airport configuration. As can be seen, larger problem sizes, such as those with 40 obstacles, are still able to execute the queries in reasonable time, and the times for execution of the queries are scalable. Different lengths of the queries are also evaluated. and the time recorded for the completion of the query with longest length Q2 scales well for different problem sizes.

Table 1: Execution times (in milliseconds) for different queries (Q) with varying number of obstacles (obst). An example NuSMV code is in Appendix A.

	Time (in milliseconds)						
Q	3 obst	5 obst	7 obst	10 obst	15 obst	20 obst	40 obst
Q1.	59.63	55.90	59.05	61.33	61.55	65.99	74.39
Q2.	59.68	58.32	60.72	58.38	61.65	63.22	74.39
Q3.	58.09	59.85	56.98	59.23	61.41	66.90	73.79
Q4.	57.11	61.36	60.59	60.16	60.24	61.72	76.06
Q5.	102.28	108.71	103.03	104.96	105.88	114.15	115.74
Q6.	40.88	44.86	43.08	41.34	44.52	48.06	52.31

6 CONCLUSION

In this work, a reasoning mechanism using temporal logic is constructed using deontic logic rules to identify ethical actions. The model abstraction based on concepts of obligation, permissible and forbidden actions. The formalism is constructed by minimal assumptions and it considers all the permissible actions conforming to the deontic logic rules. In this formalism, the knowledge of forbidden actions are required only. The formal reasoning on the model abstraction provides the user a plan of actions that is ethically admissible. The prototype of aircraft take off was used in the experimental evaluation for the computational feasibility of the model abstraction. The recorded times for the sample CTL queries proved that the formalism is scalable. This work forms foundation for construction of multiagent concurrent framework for handling ethical decision making using deontic logic. For example, modeling movement of multiple unmanned autonomous vehicles landing and taking-off.

One of the future directions of research will be to construct a model that would incorporate uncertainty in the motion of the aircraft where probabilistic model checking can be performed. Additionally, it will be critical to address uncertainty in the environment that may impact ethical consirations of the motion of an aircraft.

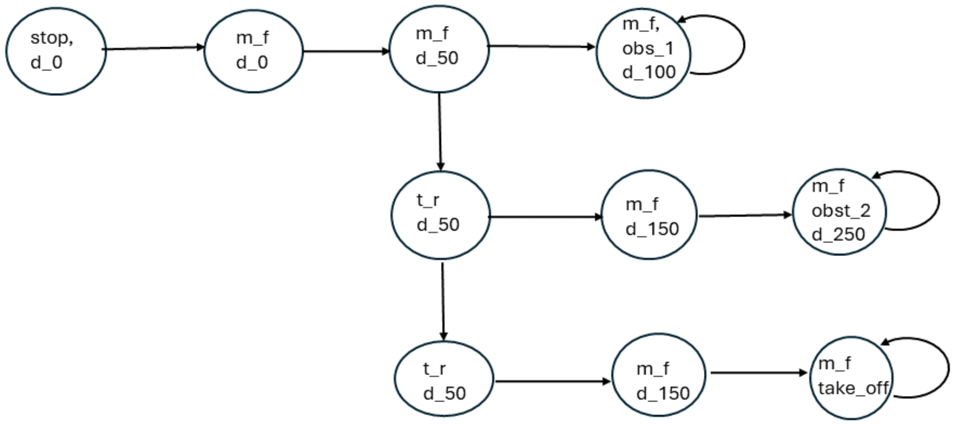

Figure 2: The figure above shows a visualization of the state labeled graph. Notation: m_f and t_r imply the actions, move_forward and turn_right, respectively.

ACKNOWLEDGEMENTS

The work is supported by NASA grant 80NSSC23M0166 and is a part of NASA EPSCoR Rapid Research Response 2023 grant.

REFERENCES

Abeywickrama, D. B., Cirstea, C., and Ramchurn, S. D. (2019). Model checking human-agent collectives for responsible ai. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–8. IEEE.

Arechiga, N. (2019). Specifying safety of autonomous vehicles in signal temporal logic. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 58–63. IEEE.

Balasubramaniam, N., Kauppinen, M., Kujala, S., and Hiekkanen, K. (2020). Ethical guidelines for solving ethical issues and developing ai systems. In *International Conference on Product-Focused Software Process Improvement*, pages 331–346. Springer.

Bonnemains, V., Saurel, C., and Tessier, C. (2018). Embedded ethics: some technical and ethical challenges. *Ethics and Information Technology*, 20:41–58.

Cimatti, A., Clarke, E., Giunchiglia, F., and Roveri, M. (2000). Nusmv: a new symbolic model checker. *International journal on software tools for technology transfer*, 2:410–425.

Clarke, E., Emerson, E., and Sistla, A. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263.

Clarke, E. M. (1997). Model checking. In *Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18–20, 1997 Proceedings 17*, pages 54–56. Springer.

Dennis, L., Fisher, M., Slavkovik, M., and Webster, M. (2016). Formal verification of ethical choices in autonomous systems. *Robotics and Autonomous Systems*, 77:1–14.

Dignum, V., Baldoni, M., Baroglio, C., Caon, M., Chatila, R., Dennis, L., Génova, G., Haim, G., Kließ, M. S., Lopez-Sanchez, M., et al. (2018). Ethics by design: Necessity or curse? In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 60–66.

Fisher, M., Mascardi, V., Rozier, K. Y., Schlingloff, B.-H., Winikoff, M., and Yorke-Smith, N. (2021). Towards a framework for certification of reliable autonomous systems. *Autonomous Agents and Multi-Agent Systems*, 35:1–65.

Furbach, U., Schon, C., and Stolzenburg, F. (2014). Automated reasoning in deontic logic. In *Multi-disciplinary Trends in Artificial Intelligence: 8th International Workshop, MIWAI 2014, Bangalore, India, December 8-10, 2014. Proceedings 8*, pages 57–68. Springer.

Gabbay, D., Horty, J., Parent, X., Van der Meyden, R., van der Torre, L., et al. (2021). *Handbook of deontic logic and normative systems*. College Publications, 2021.

Huth, M. and Ryan, M. (2004). *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge university press.

Luckcuck, M., Farrell, M., Dennis, L. A., Dixon, C., and Fisher, M. (2019). Formal specification and verification of autonomous robotic systems: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–41.

Mermet, B. and Simon, G. (2016). Formal verication of ethical properties in multiagent systems. In *1st Workshop on Ethics in the Design of Intelligent Agents*.

Roberson, T., Bornstein, S., Liivoja, R., Ng, S., Scholz, J., and Devitt, K. (2022). A method for ethical ai in defence: A case study on developing trustworthy autonomous systems. *Journal of Responsible Technology*, 11:100036.

Shea-Blymyer, C. and Abbas, H. (2022). Generating deontic obligations from utility-maximizing systems. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 653–663.