Formal Verification of Strategies for Autonomous Driving (Research Abstract)

Qais Hamarneh

Karlsruhe Institute of Technology, Karlsruhe 76131, Germany
qais.hamarneh@kit.edu

Abstract. The recent surge in the volume of autonomous driving algorithms raises the need for formal methods to compare them. In this work, we introduce a game-theoretic traffic model that allows us to verify properties of driving strategies, and to compare different strategies in terms of various driving metrics, such as energy consumption. The game model is based on urban multi-lane spatial logic, a discrete-time traffic model. The formal verification is done using strategy logic, a game-theoretic logic that allows quantifying over strategies as first-class variables.
We introduce two ways to compare strategies. The first measures the expected value of the actions suggested by the strategy in each situation independently. The second considers the expected value of the whole path from the initial state to the destination.

Keywords: Autonomous vehicles · Game theory · Urban multi-lane spatial logic · Strategy logic.

1 Motivation

Over the past decade, there has been significantly increased interest in self-driving vehicles, reflected by the growing number of published papers and industry investment. Consequently, numerous algorithms have been developed for autonomous driving. This has created a need for a formal approach that can compare different driving algorithms in terms of essential driving metrics, such as energy consumption or expected time to reach the destination.

To analyze a driving algorithm, the algorithm must be viewed within a multi-agent context. Where each agent attempts to achieve an individual goal. While the agents are not competing against each other, their actions still influence the choices available for other agents. Game theory provides an appropriate framework to model and reason about such an environment.

In this work, we make use of two types of logic: urban multi-lane spatial logic (UMLSL) [5] [7] to create a game-theoretic model and strategy logic [4] [6] to reason about and compare driving strategies.

UMLSL provides a simple discrete model of traffic and a logic, with which safety [5] [7], fairness [3], and liveness [8] of traffic maneuvers can be proven. The discrete nature of UMLSL allows us to combine it with other types of temporal logic such as strategy logic [4] [6]. Strategy logic is a game-theoretic logic that allows for quantifying over strategies as first-class variables. This feature proves particularly useful in our model for comparing different strategies and proving the existence of an optimal strategy in terms of some specific metric.

Using our UMLSL-based game model and strategy logic, we examine properties related to local actions, such as accelerating or changing lanes. We then look at global properties related to a trajectory implied by following the consecutive actions suggested by the strategy.

Initially, the traffic model will consist of a finite set of players and a finite state space. In this initial model, every player drives according to a deterministic strategy and possesses perfect knowledge of the entire map. In the second step, we augment the model by assigning each player a cognitive profile. The players' cognitive profiles influence their driving strategies. For example, more aggressive drivers tend to drive faster. Knowing the cognitive profiles of other players allows the driver to predict their next moves, and thus we assume players drive according to randomized strategies. Finally, we expand this model to accommodate an unbounded number of players, allowing cars to enter or exit the map at any step. This extension also considers an unbounded state space, and that each player's knowledge is limited to their view and the road network.

The primary focus of the formal verification aspect of this work is to express formulae for comparing strategies. This comparison is based on a given metric, such as energy consumption. We consider two types of strategy comparison.

The first type compares the sum of the expected values of the measured metric given the action suggested by the strategy over all possible states. For example, a strategy that tends to drive fast consumes more energy than a strategy that tends to drive slower.

The second type compares the expected value of the measured metric over all the possible paths from the initial state to the destination following the actions suggested by the strategy. For example, a strategy that suggests taking the highway, might result in a longer distance and consume more energy, but takes less time than a strategy that favors urban streets.

2 Preliminaries

Urban Multi-lane Spatial Logic (UMLSL) is a spatial interval logic that is based on the view of an arbitrary car [5] [7]. Figure 1 shows an example of an abstract model, where the following UMLSL formula is true in the view of car E (the shaded area):

$$\langle re(E) \frown (free^{<d} \land \neg\langle cs\rangle) \frown \langle cs\rangle\rangle$$

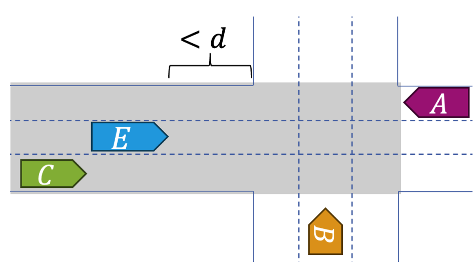

Fig. 1: Example of UMLSL model.

The formula reads: Somewhere (the outer $\langle\rangle$ is the *somewhere* operator) there exists a space reserved by the car E ($re(E)$), followed by ($\frown$) a free non-crossing ($\neg\langle cs\rangle$) space with length less than a given distance d ($free^{<d}$), followed by a crossing segment. [7] provides a controller that can tell which actions are safe for each car using UMLSL.

A **Concurrent game structure** [1] is a tuple

$$(S, Pl, Act, AP, \ell, \gamma, \tau, s_0)$$

such that S, Pl, Act, AP are non-empty sets of states, players, actions, and atomic propositions respectively. $s_0 \in S$ is the initial state. $\ell : S \to \mathcal{P}(AP)$ is a labeling function, $\gamma : S \times Pl \to \mathcal{P}(Act)$ is the enabling function. $\tau : S \times (Pl \to Act) \to S$ is a transition function. The transition function depends on the actions chosen by all players at the same time, hence the game is concurrent.

A **Strategy** [6] [2] is a function that at every step throughout the game tells the player which action to take. Strategies are defined in the game-theory logics [1] [4] as a function of the game path $f : S^* \to Act$. We also consider other variants of the strategy definition, mainly memoryless or finite-memory strategies $f : S^n \to Act$, where the action is selected based only on the last (up to) n states as well as randomized strategies $S^* \to Distr(Act)$, where $Distr(Act)$ is the set of all probability distributions on Act [2].

Strategy logic was first introduced by Chatterjee et al. [4] for two player turn-based games, then extended by Mogavero [6] for concurrent game structures. It allows for quantifying over strategies as first-class variables. For instance, $\langle\langle x\rangle\rangle [\![y]\!](\alpha, x)(\beta, y)\varphi$ reads: there exists a strategy x that can be used by player α to guarantee the property φ for any strategy y used by player β.

3 Proposed Work

Our contribution can be divided into two main parts. The first is modeling traffic as a concurrent game structure. The second is adopting strategy logic to express the essential properties of the traffic game model; in particular, expressing the two methods of strategy comparison.

3.1 The Traffic Game Model

The initial step in our work is to define a concurrent game structure based on the UMLSL traffic model. The players Pl are the traffic participants (for simplicity only cars) and the states S are composed of a set of atomic propositions which express a traffic situation using the UMLSL model. The actions allowed by the UMLSL controller include: acceleration and deceleration, turning right or left, changing lanes, and activating a turn signal. The transition function τ is also based on the transitions of the UMLSL model.

The complexity of this game model evolves over three variations. The first variation is an overly simplified model. Here we assume a finite state space and a finite set of players. Each player has full knowledge of the traffic status (even outside their view) and the final destination of all other players. Only deterministic strategies are considered in this model.

The second variation attaches a *cognitive profile* to each player. The cognitive profile of a player includes a set of parameters such as aggressiveness, and confidence. The cognitive profile helps players predict each others' next moves. For instance, an aggressive driver is more likely to go faster if given the chance. To make use of the cognitive profile, we consider randomized strategies.

In the third expansion, we consider an unbounded number of players, such that players can enter or exit the game at every step. Hence, the number of states is also unbounded. In this variation, we also consider players to have restricted knowledge. Each player's knowledge is restricted to their own view (as defined in UMLSL), the road network, and the location of their destination.

3.2 Comparing Strategies

For each variation of the game model, we utilize strategy logic to express essential properties like safety (absence of collisions). However, the primary focus of the formal verification aspect of this work lies in formalizing properties for comparing strategies. We define two methods of comparison based on a given ordered metric, taking energy consumption as an illustrative example.

The first method of comparison evaluates the overall behavior of a strategy by considering each state individually. For a given state $s \in S$, the strategy f's expected value of energy consumption is the average consumption over all states $\tau(s, d)$, where $d : Pl \to Act$ with $d(ego) = f(s)$, where $ego \in Pl$ is a fixed player. Summing up these values for all states in S, gives us a value to compare with other strategies. This comparison can also be constrained to states that satisfy specific properties, such as only urban streets and no highways.

The second method of comparison examines the entire path from the initial state s_0 until a certain property is satisfied; for instance, reaching the destination in under 5 km. The average energy consumption of every possible path, following the actions suggested by the strategy, is measured and compared to other strategies. An illustrative example of this comparison is when two strategies propose different routes to reach the same destination. One route may be shorter but congested, while the other strategy may recommend a longer but faster route. The distinction between the two strategies only becomes evident when considering the entire route.

4 Conclusion

The complexity and diversity of algorithms for autonomous traffic maneuvers require a formal approach to compare them to each other to find out which one is better suited to which situation. In this paper, we propose a traffic game model based on UMLSL [7], which allows for reasoning about and comparing driving strategies by incorporating strategy logic [6]. We conclude that our traffic game model, in combination with using strategy logic, will offer unique possibilities for formalizing and comparing algorithms for autonomous driving.

References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM (JACM) **49**(5) (2002)
2. Baier, C., Brazdil, T., Grosser, M., Kucera, A.: Stochastic game logic. In: Fourth International Conference on the Quantitative Evaluation of Systems (QEST 2007). IEEE (2007)
3. Bischopink, C., Schwammberger, M.: Verification of fair controllers for urban traffic manoeuvres at intersections. In: Formal Methods. FM 2019 International Workshops - Porto, Portugal, October 7-11, 2019, Revised Selected Papers, Part I. Lecture Notes in Computer Science, vol. 12232. Springer (2019)
4. Chatterjee, K., Henzinger, T.A., Piterman, N.: Strategy logic. Information and Computation **208**(6) (2010)
5. Hilscher, M., Linker, S., Olderog, E.R., Ravn, A.P.: An abstract model for proving safety of multi-lane traffic manoeuvres. In: Formal Methods and Software Engineering: 13th International Conference on Formal Engineering Methods, ICFEM 2011, Durham, UK, October 26-28, 2011. Proceedings 13. Springer (2011)
6. Mogavero, F.: Reasoning about strategies. In: Logics in Computer Science: A Study on Extensions of Temporal and Strategic Logics. Springer (2013)
7. Schwammberger, M.: An abstract model for proving safety of autonomous urban traffic. Theoretical Computer Science **744** (2018)
8. Schwammberger, M.: Introducing liveness into multi-lane spatial logic lane change controllers using UPPAAL. In: Proceedings 2nd International Workshop on Safe Control of Autonomous Vehicles, SCAV@CPSWeek 2018, Porto, Portugal, 10th April 2018. EPTCS, vol. 269 (2018)