A Legal Logic Programming Framework for Autonomous Vehicles

Galileo Sartor[1,*,†], Adam Wyner[1,†]

[1]Swansea University, Swansea, UK

Abstract

In this paper, we present a framework for representing and reasoning with traffic rules in Autonomous Vehicles (AVs). We base the work on a fragment of the United Kingdom's Highway Code (HC), which outlines legal requirements and good practices to fulfil reasonable duty of care. Humans and AVs will have to interact on shared roads, so we propose that a unitary, high-level computational model to be used by both types of users, which would represent shared knowledge and practice of road use, road users, and legal rules such as they appear in the HC. The road use of AVs should be consistent with the expectations of human actors. We abstract from the specifics of data acquisition, actuator use, and vehicle control to focus on reasoning with the state of the world visible to the vehicle, it's intended action, and the reason (i.e., justification) for that action. To provide such a model, we represent portions of the HC in Logical English (LE), a controlled natural language that translates into Prolog, that is then used by the vehicle; it also provides a human readable interface. The system is composed of multiple agents that have different goals: the vehicle, violation detectors, and a validator that evaluates the violations, taking into consideration mitigating factors. These systems cooperate to obtain an environment where vehicles can reason with rules in a complex, defeasible way, while maintaining safety and responsibility when determining the validity of an action.

Keywords

logic programming, automated vehicle, highway code, controlled natural language, legal reasoning

1. Introduction

Autonomous vehicles (AVs) are expected to be a transformative technology, with the potential to improve road safety, reduce traffic jams, and in general enhance mobility. We argue that for this to become reality, AVs should be able to operate in a shared space with human drivers (HVs) and other road users, making decisions on the basis of the same set of legal (and behavioural) rules that govern human driving. This is a challenging task, as AVs must be able to interpret and apply these rules in real-time, while also taking into account the actions of other road users. In this paper, we propose a framework for the legal reasoning of AVs that is based on the principles of legal reasoning and the use of formal models.

The inclusion of a strict legal framework may render AVs less flexible and adaptable to the dynamic nature of real-world driving. This is particularly important in circumstances where the law may not provide clear guidance or where strict adherence to the law may lead to dangerous situations. These circumstances are often easy for human drivers to understand, such as mounting the pavement, or moving past a red traffic light to leave room for an emergency vehicle.

In the application of traffic rules there is a distinction between hard constraints, explicit exceptions (e.g., special rules for safety vehicles), and reasons for violating a rule. While the HC has explicit constraints, not every scenario is encoded, thus a more adaptive reasoning process is needed. To address this, our framework allows for reasoning also with rule breaking (violations). In this context we deal specifically with those violations that may be seen as mitigated by the driver (in this case the AV).

Section 2 describes the current state of the art with a short selection of previous works, focusing on the representation of traffic rules with different formalisations and the issue of rule breaking in driving

LPLR'25: Workshop on Logic Programming for Legal Reasoning at ICLP, September 12-19, 2025, Rende, Italy

[*] Corresponding author.

[†] These authors contributed equally.

galileo.sartor@swansea.ac.uk (G. Sartor); a.z.wyner@swansea.ac.uk (A. Wyner)

scenarios. In section 3 the general framework is presented, identifying the different actors involved and their responsibilities and roles. This section also discusses the issue of rule breaking and dealing with violations. Section 4 delves in the technical details of the logic representation, describing how the Knowledge base is structured, and how traffic rules are combined within the violation reasoning. Section 5 will present the final element in the framework, the Simulations, and describe how it may be used to validate and analyse the running rulebase. Finally in section 6 we will summarise the current state of this ongoing development, and plot a direction for future work.

2. Related Work

Autonomous vehicles (AVs) have been a topic of interest for decades, and recent advancements in technology have made the development of fully autonomous vehicles a realistic goal. There is still a limit in the capability of AVs, especially when interacting with other agents. Most examples of “self-driving cars” (not vehicles with advanced driving assistance capabilities such as lane changing) still require human intervention to navigate more complex situations, especially when dealing with roads where the vehicles were not specifically trained, or shared roads.

The focus in development has been mainly on the implementation and improvement of automated sensors (the world perception) and actuators (the implementation of actions by the AV), and in validating vehicle behaviour and reasoning in specific complex scenarios such as lane changing. There is also a separate research direction that aims to more generally understand the complex decision-making that is required of driving, and how it can be implemented for AVs to navigate real-world traffic scenarios safely and efficiently.

One of the major hurdles remaining in the development of autonomous vehicles is enabling these systems to make informed decisions in complex scenarios where traffic regulations may be context-dependent. As discussed in [1], many traffic rules function more as situational guidelines than rigid mandates. For this reason the adaptation of the existing rulebase (e.g., the UK Highway Code) for AVs is a complex task, but one that is necessary to ensure the behaviour of AVs is aligned with the expectations of the other (human) road users.

The representation of traffic rules, as encoded in documents such as the HC can have two main approaches: (i) a limited set of rules that deal with specific verifiable situations, or (ii) a more comprehensive set of rules that include all the provisions relevant to the vehicle.

Instances of the first approach can be related to intersections [2, 3] as well as overtaking and safe distance calculations [4], which often use languages for formal verification such as Isabelle/HOL [3]. This approach is used in trajectory monitoring and planning to ensure the mathematical reasoning is safe and consistent.

The second approach involves a broader set of rules and may introduce more issues related to exceptions, vagueness, and ambiguity in the source text. Relevant instances of this approach use defeasible deontic logic (DDL)[5] to handle rule exceptions and the resolution of vague terms in rules. Furthermore, we could consider Prolog representations of intentions and actions[6], focusing on modelling the rules as combinations of beliefs, intentions, and context.

Navigating traffic rules often requires drawing on common sense and situational understanding. For AVs, enhancing their ability to reason in nuanced contexts may require encoding additional background knowledge or commonsense reasoning, as suggested in [7]. This enhancement of the reasoning in complex situations has been addressed in [8] by using case based reasoning, with the knowledge presented to the vehicle as a set of situations, the execution by the AV, and the assessment (e.g., violation, accident, ...).

In [9], the authors note that human drivers may bend traffic rules without incurring in penalties, adjusting their behaviour according to social expectations and contextual cues. They propose a “Moral ATA” which incorporates a multi-tiered rule system allowing ethical reasoning to guide or prioritize possible actions.

Many systems investigate the possibility of applying the rules in a BDI (Belief-Desire-Intention) agent [10], in order to trace the behaviour of the autonomous vehicle and validate the rules [11].

Ethical dimensions of AV behaviour have been explored in works like [12], where the presence of an “ethical knob” allows users to set the vehicle’s moral priorities. This effectively transfers liability from the manufacturer to the user by empowering passengers to select ethical parameters with regard to extreme driving scenarios.

The use of dilemmas like the trolley problem in AV ethics remains contentious. Critics argue that such binary decision frameworks fail to represent the complexity and variability of real-world driving [13]. More realistic solutions may emerge from commonsense reasoning approaches, which better reflect the kinds of trade-off humans make regularly [14].

Adaptive frameworks, such as the one discussed in [15], propose integrating reinforcement learning with monitoring systems that assess in advance whether a legal breach might be justifiable. These systems attempt to proactively manage rule violations while balancing safety and legality.

The presented work is influenced by the previously mentioned ones, with some differences in the focus or implementation. With regard to the two main models of representation this work fits in the second group, i.e., the more comprehensive rulebases, similarly to [6]. Our goal is to have a version of the HC that can be shared and understood by both human and machine agents, and for this we strive to have a higher level of isomorphism. We did not consider the issue of enhancing the detection of traffic cues (lights, signs, ...) as in [7], as we are focusing on the reasoning after the detection has occurred. The work done in perception would happen before our component enters the process. Regarding the issue of rule breaking/bending expressed in [9], we focus less on conflicting rules, rather on a generic determination for rule violation that can be traced and reasoned upon after the fact. The idea of multiple layers of reasoning is similar to the division of labour and the distinction we keep between traffic rules and behavioural rules (as will be described in the following sections). Furthermore, the issue of liability or responsibility highlighted in [12] is left for future determination, focusing at the moment on the possibility of rule breaking and considering the decisions to come from the AV (not the passenger). While the proposed system could be used to train a RL model, through feedback coming from the decision making in long term scenarios, we propose that the evaluation happen at runtime, with the “Logic” component being integrated in the AV itself.

3. The Framework

The proposed framework is based on three main components: (i) a formal representation of the legal rules that govern driving, (ii) a set of reasoning mechanisms that allow AVs to reason with these rules in real-time, and (iii) a set of mechanisms for handling rule violations, detecting them and checking the potentially mitigating circumstances.

The framework also includes three main agents: (i) the AV, (ii) the violation detectors (e.g., cameras), and (iii) the validator, which is responsible for assessing whether the detected violations merit a penalty (to one degree or another) in given a circumstance.

3.1. Division of labour

We structure the system as a multi agent simulation, where the autonomous agents are driven in part by Prolog rules. This makes it possible to have a clear division of labour between the different agents and the tools used. In particular, the Prolog rules only reflect the traffic rules (Section 4.1) and some behavioural reasoning with regard to violations (Section 3.3). The physical constraints of the world (not part of the legal rules) are left to the simulators. Examples of this are the physicality of agents, namely two agents cannot occupy the same space, or they would cause an accident. This means that we can have the different vehicles abiding (or not) by the HC rules and even vehicles acting on the basis of different rulebases, while still maintaining the same physical constraints.

This structure enables us to abstract the view that the different components have at runtime, making it possible to swap components. An example of this is the Simulator. At the current state of development there are two Simulators, as described in Section 5, but the logic rules can remain the same.

3.2. System Components

As previously discussed, the system is made of different components and agents. In this section we will briefly describe these components and their purpose. The model is designed to emulate how a similar system works currently with human drivers, so two agents (vehicles and detectors) only act with the information they can perceive of the world, while the validator reasons with information it receives from the other two.

3.2.1. Vehicle

The vehicle is designed in an abstract way, as the specifics of the vehicle's design are not relevant for the legal reasoning. We assume the vehicle will have machine learning (ML) components, both for the detection of its surroundings (through image recognition) and for the act of driving itself. The proposed Logic component is an addition that validates the actions proposed by the ML model. The Logic component is responsible for ensuring that the vehicle's actions are in compliance with the legal rules and for providing explanations for its decisions. The proposed structure is built to be modular, so that each component can be replaced, e.g., to allow vehicles to switch between different legal rules depending on the country they are in. The vehicle is designed with a set of properties that are used to determine its state, the actions it can take, and the bearing on the validation mechanism. These properties include:

- The type of vehicle (car, bus, truck, emergency vehicle, etc.): This is important for determining the rules that apply to the vehicle, as different types of vehicles may be subject to different rules.
- The status of the vehicle (normal, emergency, etc.): This is used to distinguish vehicles that are in a states that may allow rule breaking (e.g., emergency vehicles heading to an accident, normal vehicles with potentially mitigating circumstances).
- The Risk status (also called "Behaviour"): This is used to determine the risk level of the vehicle and to adjust its behaviour accordingly. This is the property that determines a vehicle's likelihood of breaking the rules.

While driving, the AVs act according to three main components:

- Belief: The state of the world surrounding the vehicle it can perceive with sensors (signs, lights, other road users).
- Justification: The abstract goals the vehicle has (time constraints, preferred plan, etc).
- Intention: The action (or set of actions) the vehicle wants to perform in a given time.

The first part, the "Belief", contains all the information the vehicle can detect about its surroundings. This information will be used when querying the system on the validity of an action. The vehicle in question will continuously collect the information from its sensors, which could also be used by an ML component (though not developed here). In this research we are not interested in how the information is collected, i.e., about the specifics of the sensors or what type they are. The first assumption we make about the system is that the output of the sensors can be trusted and that by collecting all the information it is possible to abstract high level statements about the environment, such as "there is a traffic light and its colour is red", or "there is a vehicle oncoming from the right", regardless of the specific technical aspects. The implementation or validation of this part is out of the scope for the logic component, as it relies on the abstract, "human accessible" representation.

The main difference with existing models is the "Justification" component, which is used to represent the overarching goal of the vehicle in a given situation. The justification is the high level goal that the vehicle has (time to reach its destination, emergency needs, etc.) and it will be used to determine the risk the vehicle is willing to take in safe situations with regard to breaking traffic rules. As an example the vehicle might be willing to speed if the other vehicles are going at a similar speed, thus its action would help it reach its goal without causing accidents. The Justification and how the vehicle reasons with it is going to be discussed more in detail in Section 4.2

Finally, the third part, the “Intention”, in our case is the action the vehicle intends to take in the current situation “S”. This is usually a single action, such as “entering the junction”, “crossing the traffic light”, etc. The reasoning component of the vehicle will use this as the query, to determine if this action is logically permissible in the given context.

As a simple example consider the following scenario:

```
the vehicle is at a junction.
the vehicle sees a traffic light.
the traffic light is red.
```

In this case, if the vehicle asks whether it can enter the junction, it will pose the query “ego can enter the junction”, and the predicted response should be false. This is a very simple scenario, but the same reasoning should be applicable to more complex situations, e.g., with other vehicles, traffic signs, etc. Note that the listing above represents the scenario in natural language, and the same structure is used when modelling the rules in Logical English in Section 4.3.

3.3. Violation Detection

In the proposed system, the violation detection is a static, reactive process, with no real difference from the current systems in use for human drivers. The detectors are not part of the vehicle, but are external devices that monitor the road and detect violations, such as speed cameras, red light cameras, and other types of sensors. The detectors are not able to detect the intention of the vehicle nor auxiliary information in the circumstances, but only the action that the vehicle is taking. This is a very important distinction, as it means that the detectors are not able to reason about the violation, but only to detect it. The “detected violation” is passed on to the validator for further reasoning.

The violation detectors can be made to detect different types of violations, such as speeding, running a red light, or entering a junction without stopping, just as current systems do. In this case we are considering detectors as cameras for speeding and red light detection. We are not considering here what would happen if the violation is not detected. In this case, the vehicle would still reason with the rule violation, but the violation would not be detected by the system. It may be that the vehicle is not aware of the presence of the detectors, so it cannot take them into account when reasoning about the violation.

3.4. Validator

The “Validator” is the component responsible for assessing the detected violations, admitting any mitigating information, and determining whether to apply a penalty. The analogue in the current environment would be for example a police officer or judge. This agent also has the same set of rules as the vehicle and is able to reason about the same scenario. The validator receives from the detector the detected violation and from the vehicle the details of the situation in which the violation occurred. In this case the scenario is composed of the vehicle’s type and status, the view of it’s surrounding, and the reasoning (i.e., the trace of the program execution) in which it happened. Thus each action analysed by the validator is assigned a penalty or a mitigated status. In the current implementation there is no effect of the action on further ones by the validator. While the vehicle may alter its behaviour on the basis of the previous penalties and violations, the validator determination is specific to that action. It would be possible to construct a more complex validator to keep track of the behaviour over time, and to reason with multiple concurrent violations. The validator can then use this information to determine if the violation is exempted or not. Specifically there might be valid exceptions contained in the HC or other legal provisions (e.g., emergency vehicles, etc.) that may have allowed the vehicle in question to break the rule. There might also be “Mitigating Circumstances” that may have justified the violation, such as a sudden change in the environment (e.g., a pedestrian crossing the road), a safety reason (e.g., the need to let an emergency vehicle pass), or others. While it is possible to model a list of the explicit exceptions or abnormal situations, it is not possible to have an exhaustive list. To address this we can introduce a specific term to express abnormal situations that can be satistied by either the explicit exceptions or implicit/runtime ones. This can be leveraged by the vehicle to include its own justification in the execution process, and pass this information to the Validator.

The validator would potentially be able to reason also with case base scenarios to determine if a mitigating factor is present or not. This is a very complex task, and it is not the focus of this paper, but it is a possible future direction for the research.

If the vehicle is caught violating a rule, and if the validator determines that a penalty should be applied, the vehicle will be notified of this; and it will consider the penalty as a factor in its future violation decisions. This will be discussed more in section 4.2.

While an in depth discussion of the penalty system is beyond the scope of this paper, we can mention that the penalty system is based on a points system, similar to the one used for human drivers. Vehicles start with a certain number of points, and each confirmed violation will result in a deduction of points. At each potential violation the vehicle will check its remaining points, and will thus be less willing to break rules the less points it has.

4. Knowledge Representation

In this section, we discuss the knowledge representation used in the system, which is based on the UK Highway Code (HC) and modelled in Logical English (LE). We will then discuss how the knowledge is used by the vehicle to reason about its actions and how it deals with the possibility of rule violations.

4.1. Knowledge base

The Knowledge base is derived from the UK Highway Code (HC), which is a set of rules and guidelines for road users in the UK. In the HC there are different types of rules. Some describe obligations (e.g., you must stop at a red light) and may be linked to a legal provision (e.g., the Road Traffic Act) as well as a penalty. Others are indications for good driving behaviour, usually using the term “should”. Most rules in the HC are not legally binding on their own, but they are considered best practices for road users, and they may be used as a reference by the police and the courts in case of accidents or disputes so as to allocate liability or severity of penalty (aggravating circumstances). The HC is also used as a reference by the DVLA (Driver and Vehicle Licensing Agency) for the driving test, which implies the knowledge that human drivers bring to the road. For this paper, we are focusing on a subset of the HC, namely the rules used when navigating junctions. Most rules in this section are expressed as recommendations with a few explicit obligations. This gives us a good starting point to test the system, as it allows us to reason with the different rules, and how to integrate them in the system.

4.2. Behaviour

The previously mentioned “Justification” of the vehicle is its high level goal, representing the reason the vehicle has for violating rules, e.g., rushing a patient to the hospital. The vehicle can use this to determine the best action to take in a given situation, i.e., the “Intention” in that situation. Putting them together, if the vehicle is in a hurry (thus wishes to arrive at his destination in as little time as possible), the vehicle may choose to speed up and take more risks and potentially incur more violations. The system uses this higher level goal to attempt to accomplish each action in as little time as possible if it is safe to do so, i.e., the immediate action would not cause an accident. In the current state, the relation between the Justification and Intention is not formally defined, and the implementation is a proof of concept. But the relation is still a key part of the system that will be developed further. The determination of how this high level goal is done is out of the scope for this research, as is the issue of liability in this determination . We assume that a decision has been made and that the vehicle will try to alter its behaviour accordingly.

In the running system, the justification is used to set a risk behaviour for the vehicle, which is a measure of how likely the vehicle is to break the rules. This is a simple measure done by the vehicle, which tries to combine the information about the environment with its specific justification. What this means is that the vehicle will determine that it is more likely to break the rules if it is in a hurry. To maintain a high degree of safety, the vehicle is aware about the different rules (explicit obligations and suggestions) and identifies which rules are “safety critical” and which might allow for more variation depending on context. This distinction is made as simple as possible and is based on the following criteria: If the vehicle sees another road user that it may collide with in case of a rule violation, it will stick to the rules that would prevent such accident. If, conversely, the vehicle does not perceive any other road user, it will be more flexible in its decision making, for instance deciding to speed or not stop at a junction. This is a very simple Proof of Concept approach, but it is a first step towards a more complex model.

In our framework we distinguish between explicit exception states mentioned in the norms, e.g., the different behaviour for emergency vehicles, and the violations that may be caused by a normal vehicle. Furthermore, we use the distinction in the HC between different kinds of norms:

- legal requirements, where disobeying such rules means committing a criminal offence. You may be fined, given penalty points on your licence or be disqualified from driving;
- advisory norms, that while not binding may be used in evidence in any court proceedings under the Traffic Acts.

In this context the Validator may determine that a penalty should be set or that there were mitigating circumstances that made the vehicle action admissible such as avoiding an accident.

4.3. Logic

The traffic rules are encoded in Logical English (LE) [16], a controlled natural language built on Prolog. LE enables us to write rules and interact with the program in natural language. The LE rules are automatically converted into Prolog code that is evaluated by a Prolog interpreter, and we can use this Prolog code in the autonomous vehicle[1].

The vehicles can query the Prolog code to determine if an action they intend to take is permitted or if there is a specific obligation/prohibition in that case. The system will provide a simple response, log the scenario, query, and result for future reference. This may be used in the validation process, when evaluating violations as mentioned in 3.4.

As an example, we can consider the following rules from the HC:

- Rule 170: You should [...] give way to pedestrians crossing or waiting to cross a road into which or from which you are turning. If they have started to cross they have priority, so give way (see Rule H2) [...]
- Rule 171: You MUST stop behind the line at a junction with a ‘Stop’ sign and a solid white line across the road. Wait for a safe gap in the traffic before you move off.

These rules can be seen as examples of the two types of rules we mentioned before. The first one is a recommendation , while the second one is an obligation[2]. The second rule is a good example of a strict rule in the HC, which are generally more straight forward to represent in a logic form, whereas the recommendations are often more open textured. The recommendations may be violated without a direct penalty, but if the violation causes an accident or injury to a pedestrian, it will be taken into account due to failure of duty of care in legal proceedings.

A possible representation of the two rules in Logical English can be seen in Listing 1.

To this representation we can add a further condition that the vehicle is not in an abnormal situation and further qualify this as follows:

Listing 2 shows an example of the dynamic inclusion of the Justification (i.e., an abnormal situation) in the knowledge base. When the vehicle is created it will incorporate a fact asserting its Justification,

[1] An in depth description of Logical English or its use in this context is beyond the scope of this paper, but we will provide a few examples of how it can be used to represent the rules of the road [17]

[2] In this case the rule is in fact linked to Sec. 36 of the Road Traffic Act 1988 (Drivers to comply with traffic signs).

```
% Rule 170
a vehicle should give way to a pedestrian
if the vehicle is at a junction
and the pedestrian _is crossing
    or the pedestrian _is waiting to cross.

% Rule 171
a vehicle must stop behind the line at a junction
if the vehicle is at the junction
and the vehicle sees stop sign
and the vehicle sees solid white line across the road.
```

Listing 1: Example of Logical English code

```
% Rule 171
a vehicle must stop behind the line at a junction
if the vehicle is at the junction
and the vehicle sees stop sign
and the vehicle sees solid white line across the road
and it is not the case that
  there is an exception for the vehicle.

there is an exception for a vehicle
if the vehicle is of type emergency vehicle.

% Runtime exception
there is an exception for a vehicle
if the vehicle has the justification that a justification.
```

Listing 2: Example of Logical English code with abnormal situation

```
1  should(A, 'give way', B) :-
2      is_at(A, 'the junction'),
3      (   '_is'(B, crossing)
4      ;   '_is'(B, 'waiting to cross')
5      ).
6  must(A, 'stop behind the line at a junction') :-
7      is_at(A, 'the junction'),
8      sees(A, 'stop sign'),
9      sees(A, 'solid white line across the road').
```

Listing 3: Example of Prolog code generated from Logical English

of the form “ego has the justification that ...”. This information would be used by the reasoning system to determine that the rule has an exception, thus it can be broken. The same information would be passed by the vehicle to the Validator when reasoning on the validity of the violation and the need for a penalty. Currently the representation of the violation is a work in progress, as it existed in the above mentioned form in a previous implementation, and it is being incorporated in the new system.

In Listing 3 we can see an example of the Prolog code generated from Logical English and used by the simulated vehicle. Lines 6-9 are a translation of Rule 171.

The implementation of the rule-breaking decision system is shown in Listing 4. The first two lines are only needed in the simulator to determine if the vehicle will attempt to break the rule. At runtime, this is where the vehicle would determine its own abnormal situation (as described in Section 4.2), determining if it has a justification for breaking the rule. Lines 4-5 check that vehicle A has obligation B, rule B has a penalty for violations, and the vehicle has enough tokens to (eventually) pay the fine. Lines 6-7 check that the vehicle has enough tokens to pay the penalty. Lines 8-11 determine the new “Behaviour”, and the new set of tokens, after paying the penalty. This will impact the future decision

```
can_break_rule(A, B) :-
    % has_behaviour(A, Behaviour),
    % maybe(Behaviour),
    must(A, B),
    rule_penalty(B, Penalty),
    has_tokens(A, Tokens),
    Tokens >= Penalty,
    NewTokens is Tokens - Penalty,
    set_tokens(A, NewTokens),
    NewBehaviour is Behaviour - (Penalty/Tokens) * Behaviour,
    set_behaviour(A, NewBehaviour).
```

Listing 4: Example of Prolog code for rule breaking

making of the vehicle, as it will have less tokens to pay for future penalties, and thus will be less likely to break the rules in the future.

5. Simulation

The behaviour of the system can be visualised in two main modes: (i) the scenario sequence, and (ii) the simulation. The scenario sequence is a simple list of independent scenarios the simulated vehicle is going to encounter. For each of these, the vehicle is give the relevant facts and the intended action. The vehicle will then query the system, act accordingly, log the result, then move to the next scenario.

The simulation is a more complex environment, where the vehicle is placed in a simulated environment (NetLogo or CARLA), and it has to navigate through it, encountering different situations and other simulated agents. The vehicle is able to detect its surroundings using the simulated sensors, and reasons only with the high level facts.

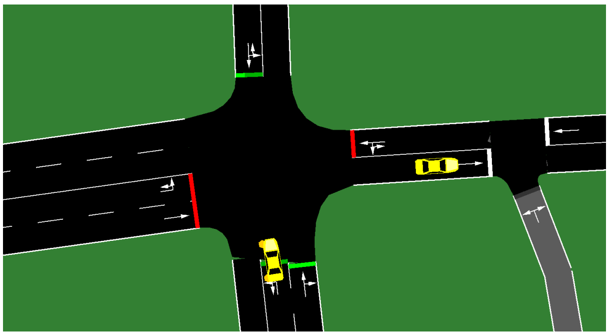

Figure 1: Example of simulation environment

In both cases the different scenarios are mostly independent, with the only link being the variation in behaviour of the vehicle.

6. Conclusion

In this paper we presented a framework to enable legal reasoning with traffic rules in autonomous vehicles. The framework is designed to allow AVs to interpret and apply traffic rules in real-time in a way that is understandable to other road users. The goal of this work is to model how AVs should behave in complex shared environments (e.g., in busy city streets). In this scenario, AVs must be able to reason and behave in a way that is consistent with the expectations of human drivers and other road users, while also being able to adapt to the dynamic nature of real-world driving without increasing the burden on the other road users. The framework also includes a set of mechanisms for handling rule violations, and a set of reasoning mechanisms that allow AVs to interpret and apply these rules in real-time.

The proposed framework will be expanded in future work in a twofold way. First, the rulebase will be expanded, increasing the number of rules and violation conditions. In addition, it will improve the tractability of the reasoning process. Further, we could consider how rule breaking can be integrated in constructing plans by simulating the potential scenarios encountered. Secondly, the integration with simulation systems will be further developed with the goal of extracting metrics on the system while running, to enable more in depth comparisons of the various approaches and changes to the model. This could help in investigating the interaction of rules that express obligations (those marked with "must" or "shall", and linked with penalties) and the other rules. Finally the validator system could be made more complex, analysing also the historic behaviour of a vehicle over time, and determining if the feedback loop onboard the vehicle requires analysing also undetected violations.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] H. Prakken, On the problem of making autonomous vehicles conform to traffic law, Artif. Intell. Law 25 (2017) 341–363.

[2] S. Maierhofer, P. Moosbrugger, M. Althoff, Formalization of intersection traffic rules in temporal logic, in: 2022 IEEE Intelligent Vehicles Symposium (IV), 2022, pp. 1135–1144. URL: https://ieeexplore.ieee.org/document/9827153. doi:10.1109/IV51971.2022.9827153.

[3] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, T. Nipkow, Formalising and monitoring traffic rules for autonomous vehicles in isabelle/HOL, in: N. Polikarpova, S. Schneider (Eds.), Integrated Formal Methods, Cham, 2017, pp. 50–66.

[4] D. M. Costescu, Autonomous vehicles' safety in mixed traffic: Accounting for incoming vehicles when overtaking, ????, pp. 1–5. URL: https://ieeexplore.ieee.org/document/8893110. doi:10.1109/EV.2019.8893110.

[5] H. Bhuiyan, G. Governatori, A. Rakotonirainy, M. W. Wong, A. Mahajan, Driving decision making of autonomous vehicle according to queensland overtaking traffic rules, The Review of Socionetwork Strategies 17 (2023) 233–254. URL: https://doi.org/10.1007/s12626-023-00147-x. doi:10.1007/s12626-023-00147-x.

[6] J. Collenette, L. A. Dennis, M. Fisher, Advising autonomous cars about the rules of the road, Electronic Proceedings in Theoretical Computer Science 371 (2022) 62–76. URL: http://arxiv.org/abs/2209.14035. doi:10.4204/EPTCS.371.5. arXiv:2209.14035 [cs].

[7] S. Kothawade, V. Khandelwal, K. Basu, H. Wang, G. Gupta, AUTO-DISCERN: Autonomous driving using common sense reasoning, in: J. Arias, F. A. D'Asaro, A. Dyoub, G. Gupta, M. Hecher, E. LeBlanc, R. Peñaloza, E. Salazar, A. Saptawijaya, F. Weitkämper, J. Zangari (Eds.), International Conference on Logic Programming 2021 Workshops, volume 2970 of *CEUR Workshop Proceedings*, CEUR, 2021. URL: https://ceur-ws.org/Vol-2970/#gdepaper7.

[8] S. Vacek, T. Gindele, J. M. Zollner, R. Dillmann, Using case-based reasoning for autonomous vehicle guidance, ????, pp. 4271–4276. URL: https://ieeexplore.ieee.org/document/4399298. doi:10.1109/IROS.2007.4399298, ISSN: 2153-0866.

[9] A. Rakow, M. Schwammberger, Brake or Drive: On the Relation Between Morality and Traffic Rules when Driving Autonomously, in: Software Engineering 2023 Workshops, ????, p. 104. URL: https://publikationen.bibliothek.kit.edu/1000169851. doi:10.18420/se2023-ws-12.

[10] M. Bratman, Intention, Plans, and Practical Reason, Cambridge, MA: Harvard University Press, Cambridge, 1987.

[11] G. V. Alves, L. Dennis, M. Fisher, A Double-Level Model Checking Approach for an Agent-Based Autonomous Vehicle and Road Junction Regulations, Journal of Sensor and Actuator Networks 10 (2021) 41.

[12] G. Contissa, F. Lagioia, G. Sartor, The ethical knob: ethically-customisable automated vehicles and the law, Artificial Intelligence and Law 25 (2017) 365–378.

[13] D. Cecchini, S. Brantley, V. Dubljević, Moral judgment in realistic traffic scenarios: moving beyond the trolley paradigm for ethics of autonomous vehicles 40 (????) 1037–1048. URL: https://doi.org/10.1007/s00146-023-01813-y. doi:10.1007/s00146-023-01813-y.

[14] J. De Freitas, A. Censi, B. Walker Smith, L. Di Lillo, S. E. Anthony, E. Frazzoli, From driverless dilemmas to more practical commonsense tests for automated vehicles 118 (????) e2010202118. URL: https://www.pnas.org/doi/10.1073/pnas.2010202118. doi:10.1073/pnas.2010202118.

[15] J. Liu, W. Zhou, H. Wang, Z. Cao, W. Yu, C. Zhao, D. Zhao, D. Yang, J. Li, Road Traffic Law Adaptive Decision-making for Self-Driving Vehicles, in: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), ????, pp. 2034–2041. URL: http://arxiv.org/abs/2204.11411. doi:10.1109/ITSC55140.2022.9922208, arXiv:2204.11411 [cs].

[16] R. Kowalski, J. Dávila, M. Calejo, Logical english for legal applications, Conference: XAIF, Virtual Workshop on XAI in Finance (2021).

[17] G. Sartor, A. Wyner, G. Contissa, Mind the gaps: Logical english, prolog, and multi-agent systems for autonomous vehicles, in: P. Cabalar, F. Fabiano, M. Gebser, G. Gupta, T. Swift (Eds.), Proceedings 40th International Conference on Logic Programming, ICLP 2024, University of Texas at Dallas, Dallas Texas, USA, October 14-17 2024, volume 416 of *EPTCS*, 2024, pp. 111–124. URL: https://doi.org/10.4204/EPTCS.416.9. doi:10.4204/EPTCS.416.9.