2015 IEEE 18th International Conference on Intelligent Transportation Systems

Formalising Traffic Rules for Accountability of Autonomous Vehicles

Albert Rizaldi and Matthias Althoff

***Abstract*— One significant barrier in introducing autonomous driving is the liability issue of a collision; e.g. when two autonomous vehicles collide, it is unclear which vehicle should be held accountable. To solve this issue, we view traffic rules from legal texts as requirements for autonomous vehicles. If we can prove that an autonomous vehicle always satisfies these requirements during its operation, then it cannot be held responsible in a collision. We present our approach by formalising a subset of traffic rules from the Vienna Convention on Road Traffic for highway scenarios in Isabelle/HOL.**

I. Introduction

Recently, there have been many studies aimed at increasing the safety of autonomous vehicles. However, another important but rarely studied area is the liability aspect of autonomous vehicles. That is, when a collision occurs, we want to determine who is responsible for it. The United Kingdom government, for example, intends to to review the regulation about the "clarification of liabilities" in a collision for autonomous vehicles [1].

From an engineering perspective, we address this issue by ensuring that autonomous vehicles always comply with the traffic rules so that they cannot be held liable for a collision. To check traffic rule compliance rigorously, we need to ensure that the traffic rules must be testable, i.e. we can *always* decide whether an autonomous vehicle's behaviour complies with traffic rules unambiguously. However, the traffic rules from legal text are written mostly in natural language and therefore, they are often *abstract* and *imprecise* [2]. As the first step to make the traffic rules machine checkable, we propose to *concretise* and *formalise* traffic rules.

Buchanan and Headrick [3] could be considered as the first to propose a serious effort in formalising law. Historically speaking, this is the period when expert systems was popular in the domain of Artificial Intelligence (cf. the history of AI in [4]). Therefore, it is not surprising that they suggested to formalise law with technique used in expert systems, that is, with *rules*. Additionally, there are also other works which use other computational structures such as algorithm, flow chart, and decision nets to formalise law. However, since we opt for logical formalism, we refrain from discussing these structures and suggest the interested readers to consult the work of Sergot [5] instead.

Two major milestones for formalising law are the works of Sergot et al. [6] and Bench-Capon et al. [7] in which they formalised the British Nationality Act and the Supplementary Benefit Act, respectively, with Horn fragments of First-Order Logic (Horn clauses) in PROLOG [8]. Despite their popularity, Horn clauses are not used in this work due to the limited expressiveness of a primitive concept [2] they can be represent. For example, when formalising a legal sentence such as "Driving should be on the *rightmost lane* if possible, except for *overtaking*" with Horn clauses, we are forced to assume that the primitive concepts *rightmost lane* and *overtaking* can always be determined. Unfortunately, for our purpose, we have to base our formalisation with primitive concepts such as positions, orientations, speeds, and accelerations; not only the high-level concepts such as 'overtaking' and 'rightmost lane' directly.

Apart from Propositional and First-Order Logic, there is also Deontic Logic [9] for formalising law which can express the notions of *permission* and *obligation* explicitly. These two notions can be easily recognised in legal documents by identifying the keywords 'may' and 'must', respectively. In relation to formalisation of traffic rules, this logic was extended by Royakkers [10], and he showed how that extended logic can address the issue of conflicting speed limits in Dutch Traffic Regulation 1990.

We also decided not to use Deontic Logic because our goal is to elicit a set of formal specification for autonomous vehicles from legal texts. Deontic Logic, meanwhile, is more suitable to prove that a law sentence can be deduced from a set of legal facts and concepts. This observation is inline with the study from Jones and Sergot [11] in which they discuss when to use and not to use Deontic Logic for formalising law.

Another line of work other than formalisation of law which is related to our work is the formal verification of autonomous vehicles. Formal verification for autonomous vehicles can be broadly classified into two categories: set-based and logic-based approaches. With set-based approaches, such as in [12], a requirement is formalised by specifying a set of acceptable behaviours. Meanwhile, with logic-based approaches, such as [13]–[15], a requirement is formalised by translating it into a logical sentence directly. In this work, we opt for logic-based approach because logical operators, such as disjunction, conjunction, and implication, are closer to natural languages than set operators, such as set complement, union, and intersection.

Logic-based approaches in [13]–[15] are aimed more towards verifying the detailed *design* of a system rather than specifying a complex requirement. That is, in addition to formalising the properties, we would also need to provide the system to be verified. Since our work is aimed at the early *specification* phase, we refrain from using these logics.

Albert Rizaldi and Matthias Althoff are with the Technische Universität München, Fakultät für Informatik, Lehrstuhl für Robotik und Echtzeitsysteme, Boltzmannstraße 3, 85748, Garching, Germany. {rizaldi, althoff}@in.tum.de

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.269

IEEE computer society

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

Instead, we use Higher Order Logic (HOL), a purely logical environment, with the Isabelle theorem prover [16] as our research platform.

In this work, we show that by concretising and formalising traffic rules in Higher Order Logic using the Isabelle theorem prover, it is possible to check the compliance of traffic rules unambiguously and formally. We list the contributions of our paper as follows:

- We formalise hybrid traces which could be perceived as the abstraction of autonomous vehicles' behaviours obtained from the data recorded on a black box (see e.g. [17]) in Isabelle/HOL (Sec. III);
- We formalise a subset of traffic rules from the Vienna Convention for Road Traffic which applies to highway scenarios in Isabelle/HOL (Sec. III);
- We show a pattern for deriving a procedure for each rule to verify whether a hybrid trace satisfies the rule correctly (Sec. IV)

Of course, our work is the first step in the direction of formalising traffic rules and it needs further discussion to possibly become a standard. However, to our best knowledge, there has not been any published work with the same goal.

II. Problem Formulation

Our main source of legal text is the Vienna Convention on Road Traffic from 1968 [18]. Since this treaty has been widely ratified[1], our formalisation can be applied for many countries; something that is more difficult to achieve had we chosen a legal text for a specific country only. However, our approach in principle can also be applied for national traffic rules.

Since the Vienna Convention on Road Traffic contains many traffic rules, we select a subset of traffic rules, which applies for (controlled-access) highways only. This is the most likely domain that fully autonomous vehicles will firstly be legal. In addition, it has a simpler structure compared to, for example, urban environments, which require additional considerations of pedestrians, bicyclists, oncoming traffic, etc.

Vanholme et al. [19] have previously elicited, simplified, and paraphrased the traffic rules from the Vienna Convention on Road Traffic for highways. The list of traffic rules considered in this work is the subset of rules from their work and it is shown in Tab. I. Column 'Article' in the table shows the corresponding article in the Vienna Convention on Road Traffic associated with each traffic rule. Also note that Article 34 will be formalised differently from the rest of the rules. This is because this rule is a *metarule* which alters the meaning of other rules.

TABLE I

Traffic Rules from the Vienna Convention on Road Traffic for Highways.

Identifier	Description	Article
Φ_1	Road users should avoid damage to road infrastructure or to other road users.	7
Φ_2	Driving should be on the rightmost lane if possible, except for overtaking.	10
Φ_3	Speed must be adapted to road and weather conditions (e.g. visibility and road friction), speed limit signs, and the presence of other vehicles.	13
Φ_4	The distance between vehicles must be such that a collision can be avoided if a vehicle performs an emergency brake.	13
Φ_5	Braking should only be performed for safety reasons and must be indicated with braking lights.	17
Ψ	Priority vehicles are exempt from traffic rules, except from Article 7.	34

Let us denote the set of formalised traffic rules from the Vienna Convention on Road Traffic as $\Phi_1, \Phi_2, \ldots, \Phi_n$, the behaviour of an autonomous car as a trace ρ, and the environment as $\mathcal{E}$. Then the main problem addressed in this work is to check whether the behaviour ρ, given an environment $\mathcal{E}$, satisfies all the traffic rules $\Phi_1, \Phi_2, \ldots, \Phi_n$. That is,

$$\rho, \mathcal{E} \models \bigwedge_{i=1}^{n} \Phi_i \ . \tag{1}$$

The notation $\models$ is called the satisfaction relation and this problem is usually called the model checking problem [20].

In this work, we check the satisfiability problem in Eq. (1), by deriving a procedure Φ_i-*check* which checks trace ρ and environment $\mathcal{E}$ against property Φ_i such that

$$\Phi_i\text{-}\mathit{check}(\rho, \mathcal{E}) = \mathsf{True} \iff \rho, \mathcal{E} \models \Phi_i \ . \tag{2}$$

Then, the problem of satisfiability in Eq. (1) can be divided into the subproblems of checking each property Φ_i separately by using Φ_i-*check* for $1 \le i \le n$. To ensure that our approach is rigorous, we also need to verify Eq. (2) for each procedure Φ_i-*check*.

III. Formalising Vienna Convention Traffic Rules

This section presents the formalisation of the traffic rules from the Vienna Convention on Road Traffic in Isabelle/HOL. We start this section with a brief introduction to the notation used in Isabelle/HOL. Since each formalised traffic rule Φ_i will be checked against a trace ρ (Eq. 1), we first formalise the traces ρ and continue with formalising the traffic rules in Isabelle/HOL.

A. Notations

Isabelle/HOL uses type theory as its basis to encode the proofs, therefore each term in our formalisation has a type. We denote a term t with the type τ by $t :: \tau$. For example, a lane can be identified with a natural number, that is, *lane-id* $:: \mathbb{N}$. Function type is denoted by $\Rightarrow$. Thus, a function which describes the speed of an autonomous vehicle over time is represented by *speed* $:: \mathbb{R} \Rightarrow \mathbb{R}$. Function

[1] https://treaties.un.org/pages/ViewDetailsIII.aspx?&src=TREATY&mtdsg_no=XI~B~19&chapter=11&Temp=mtdsg3&lang=en

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

applications are often written without parentheses; the speed of the vehicle at time t is written as $\mathit{speed}\ t$ instead of $\mathit{speed}\,(t)$. Likewise, a function with more than one argument is represented without commas and parentheses. For instance, the lane currently occupied by the ego vehicle given a trace ρ and time t is denoted by $\mathit{lane\text{-}occupied}\ \rho\ t$. A set of type α is denoted by $\alpha\ \mathit{set}$. For example, a set of lanes occupied by the autonomous vehicle is therefore denoted by $\mathit{occupied\text{-}lanes} :: \mathbb{N}\ \mathit{set}$. When discussing the list datatype, we use the operator $\cdot$ to append an element to the head of a list. Notation $\mathit{LEAST}\, x.\ P$ denotes the smallest element x such that predicate P is true. The infimum of a set is denoted by *Inf*, and similarly, the supremum of a set is denoted by *Sup*.

B. Hybrid Traces

In practice, the behaviour of an autonomous vehicle is obtained from a black box. A black box could include data, such as position, orientation, speed, and acceleration, which evolve continuously over time. However, it could also record data such as the lane it currently occupies or whether the braking light is turned on or not. Unlike the previous data, these data evolve only at certain time points and remain unchanged for the rest of the time. To capture these two types of data, we formalise the trace ρ as hybrid traces formalised as follows.

Definition 1 (Continuous Valuation and Activity): Suppose that we denote the type of continuous variables with V_C. A continuous valuation is a function v which assigns a real number value to a continuous variable, i.e. $v\ ::\ V_C \Rightarrow \mathit{real}$. A continuous activity meanwhile is the function a which specifies the valuation at each time, that is, $a\ ::\ \mathit{real} \Rightarrow V_C \Rightarrow \mathit{real}$.

Definition 2 (Discrete Valuation): Suppose that we have a set of discrete variables V_D and a list of their types DT. A discrete valuation is a function j which assigns each discrete variable with a value according to its type, that is, $j\ ::\ V_D \Rightarrow DT$.

Definition 3 (Event and Hybrid Trace): An event could be either:

1) a pair $(d,\ a)$ where a is a continuous activity (see Def. 1) and $d :: \mathbb{R}$ is the duration of the activity; or
2) a discrete valuation (see Def. 2).

A hybrid trace ρ is a finite list of events.

Def. 3 does not put any restriction for the duration. However, since the duration of an event cannot be negative, we always assume that the duration is nonnegative. Additionally, Def. 3 does not rule out hybrid traces containing consecutive discrete-discrete valuation or continuous-continuous activity. Consider the situation where we have two consecutive discrete valuations such that each valuation updates different discrete variables. Since discrete valuations take no time, we can combine them into single function and hence a single event[2]. A similar argument also applies for the continuous activities. Hence, we wish to eliminate this type of hybrid trace by defining the alternating property for hybrid traces as follows:

Definition 4 (Alternating Hybrid Traces): A trace ρ is an alternating hybrid trace if there are no two consecutive continuous flows nor two consecutive discrete valuations.

Note also that Def. 3 allows the duration d to be negative. We eliminate this condition by defining the non-Zeno condition as follows.

Definition 5 (Non-Zeno Traces): A trace ρ is non-Zeno if each continuous activity (d, a) in trace ρ has positive duration $0 < d$.

For each event in a trace, there is a corresponding interval of time for which the event is valid. We define a function $\mathit{time\text{-}interval}$ which takes a trace ρ as an argument and returns the time interval I to which the first event of the trace corresponds. We list the properties of these time intervals as follows [21].

Theorem 1 (Hybrid Traces): If a trace ρ is non-Zeno, and

1) the head of trace ρ is a discrete valuation, then
$\exists\, t_3{\geq}0.\ (\mathit{time\text{-}interval}\ \rho) = \{t \mid t_3 \leq t \wedge t \leq t_3\}$
2) the head of trace ρ is a continuous flow, then
$\exists\, t_1\ t_2.$
$(0 \leq t_1 \wedge 0 \leq t_2 \wedge t_1 < t_2) \wedge$
$(\mathit{time\text{-}interval}\ \rho) = \{t \mid t_1 < t \wedge t < t_2\}$
3) $\rho = e \cdot \rho_t$ where e is any event, and the interval of time associated with ρ_t and ρ is I_1 and I_2 respectively, then
$$\mathit{Inf}\ I_2 = \mathit{Sup}\ I_1$$
4) trace ρ is alternating and non-empty, then there is a maximum value $\mathit{currt}\ \rho$ such that the union of all time intervals in the trace, $\mathit{complete\text{-}time}\ \rho$, is
   - if the head of trace ρ is a discrete valuation:
   $(\mathit{complete\text{-}interval}\ \rho) = \{t \mid 0 \leq t \wedge t \leq \mathit{currt}\ \rho\}$
   - if the head of trace ρ is a continuous flow:
   $(\mathit{complete\text{-}interval}\ \rho) = \{t \mid 0 \leq t \wedge t < \mathit{currt}\ \rho\}$

C. Traffic Rules

We begin formalising traffic rules by instantiating the continuous and discrete variables as in Def. 1 and Def. 2. There are five relevant continuous variables:

$$V_C = \mathit{position\text{-}x} \mid \mathit{position\text{-}y}\ \mid \mathit{orientation} \mid \mathit{speed} \mid \mathit{acceleration}$$

We assume that the black box records these data for both the ego vehicle and the other traffic participants. As for the discrete variables, there are only two relevant discrete variables:

$$V_D = \mathit{lane\text{-}identifier} \mid \mathit{brake\text{-}light}$$

[2] If two consecutive discrete valuations update the same discrete variables, we assume that the last discrete valuation takes effect. This is because there could only be one event at one time in reality.

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

Each lane on a highway is identified by a natural number. For left-driving countries, the leftmost lane is identified by 0 and increased accordingly to the right. The reverse is true for the right-driving countries.

Since a vehicle can occupy more than one lane, i.e. when performing an overtaking manoeuvre, the type of the discrete variable *lane-identifier* is the set of natural numbers $\mathbb{N}$ *set*. Variable *brake-light*, however, represents the status of the braking lights. Since we are interested whether the braking lights are turned on or off, it is formalised by making its type boolean $\mathbb{B}$. A discrete valuation which assign the value *True* to this variable is interpreted as the situation where the braking lights are turned on; the converse is also true.

1) Free of Collision (Rule Φ_1): There are two keywords to consider if we want to formalise rule Φ_1 (Article 7): "road infrastructure" and "damage". We interpret the former keyword by assuming that there are two path-connected[3] objects: $\mathit{left\text{-}road\text{-}boundary} :: (\mathbb{R} \times \mathbb{R})\ \mathit{set}$ and $\mathit{right\text{-}road\text{-}boundary} :: (\mathbb{R} \times \mathbb{R})\ \mathit{set}$. Additionally, these two road boundaries are assumed to be disjoint.

$$\mathit{left\text{-}road\text{-}boundary} \cap \mathit{right\text{-}road\text{-}boundary} = \emptyset$$

The latter keyword is interpreted by checking the intersection of occupancies. We consider the ego vehicle to cause damage to road infrastructure if its occupancy intersects with either the left or right road boundaries. The requirement that the ego vehicle does not cause damage to road infrastructure is formalised by defining $\mathit{no\text{-}collide\text{-}boundaries}\ \rho$ as follows:

$$\begin{aligned} &\forall\, t \in (\mathit{complete\text{-}interval}\ \rho).\\ &\quad (\mathit{ego\text{-}occupancy}\ \rho\ t) \cap \mathit{left\text{-}road\text{-}boundary} = \emptyset\ \wedge\\ &\quad (\mathit{ego\text{-}occupancy}\ \rho\ t) \cap \mathit{right\text{-}road\text{-}boundary} = \emptyset \end{aligned}$$

The definition above requires the function $\mathit{ego\text{-}occupancy}\ \rho\ t$, which computes the occupancy of the ego vehicle for trace ρ and time t. If we represent that the shape of the ego vehicle by a set of points in $\mathbb{R} \times \mathbb{R}$ with its centre at $(0, 0)$, the occupancy can be obtained by translating the shape of the ego vehicle to the current position and rotating it by the current orientation.

The rule that the ego vehicle "should not cause damage to the other traffic participants" is defined such that the ego vehicle's occupancy does not intersect with any traffic participant's occupancy. It is formalised by defining $\mathit{no\text{-}collide\text{-}traffic\text{-}participant}\ \rho$ as follows:

$$\begin{aligned} &\forall\, t \in (\mathit{complete\text{-}interval}\ \rho).\\ &\quad \forall\, \mathit{tpid}.\\ &\qquad (\mathit{ego\text{-}occupancy}\ \rho\ t) \cap (\mathit{tp\text{-}occupancy}\ \rho\ \mathit{tpid}\ t) = \emptyset \end{aligned}$$

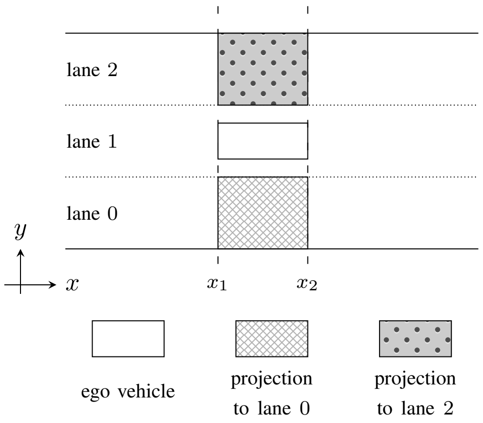

Fig. 1. An illustration to projection function.

The function $\mathit{tp\text{-}occupancy}\ \rho\ \mathit{tpid}\ t$ is similar to the function $\mathit{ego\text{-}occupancy}\ \rho\ t$, except that the former requires the identifier of the other traffic participants *tpid*. This is because each traffic participant can have different shapes, positions, and orientations. Finally, given a hybrid trace ρ, rule Φ_1 is formalised as follows:

$$\mathit{no\text{-}collide\text{-}boundaries}\ \rho \wedge \mathit{no\text{-}collide\text{-}traffic\text{-}participant}\ \rho$$

2) Rightmost Lane and Priority Vehicle (Rule Φ_2 and Ψ): In order to formalise rule Φ_2, we first assume that there is a function which maps a lane identifier with the real lane object $\mathit{lane\text{-}mapping} :: \mathbb{N} \Rightarrow (\mathbb{R} \times \mathbb{R})\ \mathit{set}$. As usual, we also assume that each set of points mapped to a lane identifier i is path-connected:

$$\forall\, i.\ \mathit{path\text{-}connected}\ (\mathit{lane\text{-}mapping}\ i)$$

and disjoint:

$$\forall\, i\ j.\ i \neq j \longrightarrow \mathit{lane\text{-}mapping}\ i \cap \mathit{lane\text{-}mapping}\ j = \emptyset$$

Two important keywords for rule Φ_2 are "rightmost lane" and "overtaking". The rightmost lane, or the leftmost lane for the left-driving countries, is interpreted as the lane with the smallest identifier such that the occupancy of the ego vehicle does not intersect with the other traffic participants' occupancy. To explain this notion clearly, we introduce the *projection* function.

Function $\mathit{projection}\ \rho\ t\ n$ finds the set of points mapped to lane identifier n whose x-elements also belong to the occupancy of the ego vehicle at time t (see Fig. 1).

$$\begin{aligned} &\mathit{projection}\ \rho\ t\ i =\\ &\{(x,\, y) \mid (x,\, y) \in (\mathit{lane\text{-}mapping}\ i)\ \wedge\\ &\qquad\qquad (\exists\, y_2.\ (x,\, y_2) \in (\mathit{ego\text{-}occupancy}\ \rho\ t))\} \end{aligned}$$

The rightmost lane is then formalised as a function as follows:

$$\begin{aligned} &\mathit{LEAST}\ n.\\ &\forall\, \mathit{tpid}.\ (\mathit{projection}\ \rho\ t\ n) \cap (\mathit{tp\text{-}occupancy}\ \rho\ \mathit{tpid}\ t) = \emptyset \end{aligned}$$

This function finds the smallest lane identifier n such that the projection of the ego vehicle's occupancy at time t, i.e. $\mathit{projection}\ \rho\ t\ n$, does not intersect with the occupancies of any traffic participant at time t, i.e. $\mathit{tp\text{-}occupancy}\ \rho\ \mathit{tpid}$

[3] Path-connectedness simply means that the objects with this property are solid objects and not just sets of scattered points in two dimensional space. Formally, it means for any two points in a path-connected object, we can always draw a line connecting those two points; and the line must always lie inside the object.

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

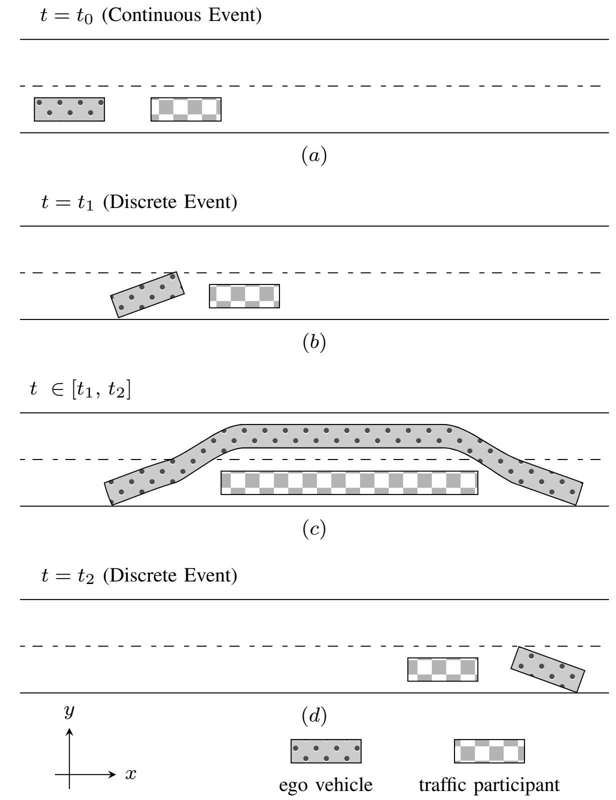
$t = t_0$ (Continuous Event)
(a)
$t = t_1$ (Discrete Event)
(b)
$t \in [t_1, t_2]$
(c)
$t = t_2$ (Discrete Event)
(d)
y
x
ego vehicle
traffic participant

Fig. 2. An illustration of overtaking scenario where $t_0 < t_1 < t_2$. The initial traffic situation is depcited in part (a). At time $t = t_1$, the tip of the ego vehicle touches the lane divider and it marks the beginning of overtaking manoeuvre (part (b)). Part (c) shows the occupancies at time $t \in [t_1, t_2]$. At time $t = t_2$, the occupancy of the ego vehicle is completely back to the initial lane (part (d)).

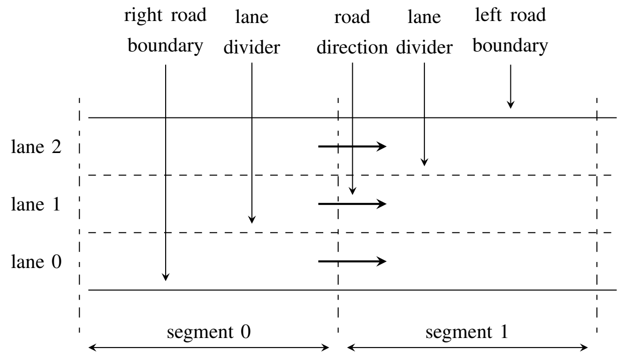

Fig. 3. An example how the road is segmentised into two segments for each lane.

t.

When we overtake another traffic participant, *1)* as soon as we surpass this traffic participant; and *2)* the projection to the lane *lane-id* is free of any traffic participant's occupancy; the definition of function *rightmost* above stipulates that we have to return to this lane immediately. However, when we overtake another vehicle, we have to leave a sufficient distance before returning to the original lane. This is where the phrase "except for overtaking" solves these two conflicting requirements. As long as we are by definition performing an overtaking manoeuvre, we do not have to return to the rightmost lane immediately.

The keyword "overtaking" in this article refers to the time interval when the ego vehicle is performing an overtaking manoeuvre. We say that time interval $[t_1, t_2]$ is the overtaking interval of the hybrid trace ρ if and only if the following conditions are satisfied (see Fig. 2).

1) At time t_1 (Fig. 2b), the ego vehicle starts the overtaking manoeuvre. The starting time of the overtaking manoeuvre is defined as the earliest time when the ego vehicle occupies the two lanes $\{\mathit{lane\text{-}id},\ \mathit{lane\text{-}id} + \mathit{1}\}$ simultaneously. This is the time when the corner of the ego vehicle's occupancy just starts to touch the lane divider.
$$(\mathit{lane\text{-}occupied}\ \rho\ t_1) = \{\mathit{lane\text{-}id},\ \mathit{lane\text{-}id} + \mathit{1}\}$$
2) There is a traffic participant identified with tpid located in front of the ego vehicle at t_1.
$$\mathit{front}\ (\mathit{tp\text{-}occupancy}\ \rho\ \mathit{tpid}\ t)\ (\mathit{ego\text{-}occupancy}\ \rho\ t)$$
3) At time t_2 (Fig. 2d) where $t_2 > t_1$, the ego vehicle ends the overtaking manoeuvre. The end of the overtaking manoeuvre is the earliest time when the ego vehicle's occupancy completely re-enters the lane occupied at the beginning of the overtaking manoeuvre.
$$(\mathit{lane\text{-}occupied}\ \rho\ t_2) = \{\mathit{lane\text{-}id}\}$$
4) The traffic participant identified with tpid as in Condition 2 is located at the behind of the ego vehicle at time t_2 (Fig. 2d).
$$\mathit{behind}\ (\mathit{tp\text{-}occupancy}\ \rho\ \mathit{tpid}\ t)\ (\mathit{ego\text{-}occupancy}\ \rho\ t)$$

By using the function *lane-occupied* which returns the set of lanes currently occupied, we formalise the rule Φ_2 and Ψ for a hybrid trace ρ with predicate *rightmost-lane* ρ as follows:

$$\begin{aligned}&\forall\, t \in \mathit{complete\text{-}interval}\ \rho.\\ &\quad \neg\ \mathit{priority\text{-}car} \vee \neg\ (\mathit{overtaking\text{-}interval}\ \rho\ t) \longrightarrow\\ &\qquad\qquad (\mathit{lane\text{-}occupied}\ \rho\ t) = \{\mathit{rightmost}\ \rho\ t\}\end{aligned}$$

In the definition above, we specify that at all times the lane occupied by the ego vehicle must be the rightmost lane, except when the ego vehicle is a priority car (Article 34) or it is performing an overtaking manoeuvre.

3) Speed Limit (Rule Φ_3): For formalising the speed limit rule in Article 13, we assume that the road can be segmented [22] as in Fig. 3. That is, there is a function which maps each point (x, y) to a segment (i, j), where i and j denote the lane and the segment identifier respectively. The speed limit signs in Rule Φ_3 are then formalised as a function which assign a speed limit to each segment identifier (i, j), i.e. $\mathit{speed\text{-}limit} :: \mathbb{N} \times \mathbb{N} \Rightarrow \mathbb{R}$. With additional functions *ego-speed* and *current-segment* which return the current

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

speed and segment the ego vehicle occupies respectively, we formalise rule Φ_3 as follows:

$$\begin{aligned}
&\textit{speed-limit-compliance}\ \rho \equiv \\
&\forall\, t \in\ (\textit{complete-interval}\ \rho). \\
&\qquad (\textit{ego-speed}\ \rho\ t) < \textit{speed-limit}\ (\textit{current-segment}\ \rho\ t)
\end{aligned}$$

4) Safe Distance (Rule Φ_4): The next property specified in Article 13 concerns the safe distance between the ego vehicle and the other traffic participant in front of it. However, the legal text does not specifically define how far a safe distance is. To determine this, we concretise this requirement by analysing the initial distance required if both the ego vehicle and the traffic participant in front of it perform an emergency brake simultaneously[4]. Both the ego vehicle and the vehicle in front of it are viewed as a point mass. More precisely, the ego vehicle is represented by its front position while the other vehicle is represented by its rear position.

The following equation determines the position of an object in a straight line over time, given an initial position s_0, an initial speed v_0 and an acceleration a.

$$s(t) = s_0 \ + \ v_0 \cdot t \ + \ \frac{1}{2} \cdot a \cdot t^2 \tag{3}$$

We use indexes 'ego' and 'tp' as the qualifier for the ego vehicle and the vehicle in front of it, respectively, except for the acceleration where we assume that we have the same maximum deceleration[5]. The difference of the distance between these two vehicles after performing emergency brake over time is

$$\begin{aligned}
\Delta s(t) &= (s_{0,tp} \ + \ v_{0,tp} \cdot t \ + \ \frac{1}{2} \cdot a \cdot t^2) - \\
&\quad (s_{0,ego} \ + \ v_{0,ego} \cdot t \ + \ \frac{1}{2} \cdot a \cdot t^2) \\
&= \Delta s_0 \ + \ \Delta v_0 \cdot t
\end{aligned} \tag{4}$$

with

$$\begin{aligned}
\Delta s_0 &= s_{0,tp} - s_{0,ego}, \\
\Delta v_0 &= v_{0,tp} - v_{0,ego}.
\end{aligned}$$

With the equation above, we are particularly interested with the difference between these two vehicles when both vehicles have completely stopped, i.e. $\Delta s(t_{stop})$. The following two equations compute the time required for both vehicles to come to a complete stop.

$$t_{stop,ego} = -\frac{v_{0,ego}}{a} \qquad\quad t_{stop,tp} = -\frac{v_{0,tp}}{a} \tag{5}$$

For this particular scenario, the value for the time t_{stop} is the maximum between these two stoppage times.

$$t_{stop} \ = \ \max\{t_{stop,ego},\, t_{stop,tp}\} \tag{6}$$

For a particular time instant, the requirement of safe distance can be formalised by checking whether the following inequality is true or not.

$$\Delta s(t_{stop}) > 0 \tag{7}$$

Before we formalise rule Φ_4, we introduce the predicate $\textit{closest-in-front-ego}\ \rho\ t\ \textit{tpid}$ which is true when the traffic participant with identifier $\textit{tpid}$ satisfies the following conditions:

1) Traffic participant $\textit{tpid}$ is located in front of the ego vehicle.

   $\textit{front}\ (\textit{tp-occupancy}\ \rho\ \textit{tpid}\ t)\ (\textit{ego-occupancy}\ \rho\ t)$

2) Traffic participant $\textit{tpid}$ is at the same lane with the ego vehicle.

   $$\begin{aligned}
   &(\textit{tp-occupancy}\ \rho\ \textit{tpid}\ t) \\
   &\qquad\qquad \subseteq \\
   &\quad (\textstyle\bigcup_{(\textit{lane} \in \textit{lane-occupied}\ \rho\ t)}\ (\textit{lane-mapping lane}))
   \end{aligned}$$

3) Each traffic participant $\textit{tpid}' \neq \textit{tpid}$ which satisfies Condition 1 and 2 above must be located in front of *tpid.*

   $\textit{front}\ (\textit{tp-occupancy}\ \rho\ \textit{tpid}'\ t)\ (\textit{tp-occupancy}\ \rho\ \textit{tpid}\ t)$

If we formalise the function which computes $\Delta(t_{stop})$ in Eq. 7 for traffic participant $\textit{tpid}$ with $\textit{dist-at-stop-time}\ \rho\ t\ \textit{tpid}$, then rule Φ_4 is formalised as follows:

$$\begin{aligned}
&\textit{safe-distance}\ \rho \equiv \\
&\forall\, t \in (\textit{complete-interval}\ \rho). \\
&\quad \forall\, \textit{tpid}.\ (\textit{closest-in-front-ego}\ \rho\ t\ \textit{tpid}) \longrightarrow \\
&\qquad\qquad\qquad\qquad 0 < (\textit{dist-at-stop-time}\ \rho\ t\ \textit{tpid})
\end{aligned}$$

Note that the formalisation above applies only when we assume equal maximum deceleration for both vehicles.

5) Braking lights: In order to formalise rule Φ_5, we check at each time point whether acceleration of the ego vehicle is negative or not. If it is, then the braking lights must be turned on. This rule is formalised as follows.

$$\begin{aligned}
&\textit{turn-brake-light}\ \rho \equiv \\
&\forall\, t \in (\textit{complete-interval}\ \rho). \\
&\qquad (\textit{ego-acceleration}\ \rho\ t) < \mathit{0} \longrightarrow (\textit{brake-lights-on}\ \rho\ t)
\end{aligned}$$

IV. Deriving Procedure to Verify Traffic Rules

All traffic rules formalised in this work belong to the category of *safety properties*. A safety property is usually characterised by the occurrence of universal quantification over time. As can be seen from all rules formalised in Sec. III-C, they have the form:

$$\forall t \in \textit{complete-interval}\ \rho\ . \quad P_i$$

where P_i is a predicate which characterises rule Φ_i. One obvious way to derive a procedure to check rule Φ_i, that is Φ_i*-check*, is to test whether P_i is true or not for every point in time in $\textit{complete-interval}\,\rho$. However, we know from Thm. 1 that the time interval is a subset of real numbers and that the number of points in time in this interval is infinite. Therefore, using this approach would be computationally infeasible.

[4] We assume this to simplify our formalisation. In reality, there must be a delay between these two brakings (reaction time).

[5] Situations where decelerations are not the same can be easily extended. See the work by Weiß [23]

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

Although the number of points in time in a time interval is infinite, the number of events in our hybrid trace is finite (see Def. 3). Therefore, we check the predicate P_i for each event in the hybrid trace ρ instead of checking it at each point in time t in the time interval $\textit{complete-interval}\ \rho$. This observation implies the following pattern for deriving procedure Φ_i-*check*.

1) $\Phi_i\textit{-check}\ \ []\ \ = \ \ \Phi_i\textit{-check-nil}$
2) $\Phi_i\textit{-check}\ \ e \cdot \rho_t = \ \ \Phi_i\textit{-check-event}\ e\ \wedge\ \Phi_i\textit{-check}\ \rho_t$

There are two cases which constitute the body of the procedure Φ_i-*check*. In the first case, we need to derive the expression Φ_i-*check-nil* which checks whether an empty hybrid trace complies with rule Φ_i or not (base case). Since an empty hybrid trace corresponds to the time $t = 0$, this case checks whether the initialisation satisfies rule Φ_i or not. For example, rule Φ_5 is not satisfied only when the initial acceleration is negative and the brake light is not turned on. This is expressed as follows:

$$\Phi_i\textit{-check-nil}\ \ = \\ (\textit{icv}\ \textit{acc}_{\textit{ego}} < 0) \longrightarrow (\textit{idv}\ \textit{brake\_light} = \textit{True})$$

Function *icv* and *idv* takes continuous and discrete variables as arguments, respectively, and return the corresponding initial value.

The second case considers the case when the hybrid trace is not empty (inductive case). A trace $\rho = e \cdot \rho_t$ complies with rule Φ_i if *i)* the single event trace $[e]$ with initialisation obtained from the values in the head of ρ_t complies with rule Φ_i; and *ii)* the tail ρ_t complies with rule Φ_i. We check the compliance of the former and the latter by using Φ_i-*check-event* and Φ_i-*check* respectively. For example, in order to comply with rule Φ_5, a single event trace $[e]$ with initialisation obtained from the head of ρ_t must satisfy the following conditions:

1) If the event e is a continuous flow (a, d),

$$\Phi_i\textit{-check-event}\ \ e\ = \\ \forall t.\ 0 < t\ \wedge\ t < d\ \wedge\ (a\ t\ \textit{acc}_{\textit{ego}} < 0) \longrightarrow \\ \textit{idv}\ \textit{brake\_light} = \textit{True})$$

2) When the event e is a discrete valuation j,

$$\Phi_i\textit{-check-event}\ \ e\ = \\ (\textit{icv}\ \textit{acc}_{\textit{ego}} < 0) \longrightarrow (j\ \textit{brake\_light} = \textit{True})$$

A continuous flow (a, d) complies with rule Φ_5 provided that braking lights are turned on if the acceleration throughout the whole duration d is negative. Note that this definition implies that the situation where the acceleration changes from a negative to a nonnegative value should be regarded as two continuous flows and there must be a discrete valuation in between them to update the braking lights accordingly. Meanwhile, if a discrete valuation turns on braking lights when the acceleration is negative, then this discrete valuation complies with rule Φ_5.

The next step after deriving this function is to prove that function Φ_i-*check* indeed satisfies Eq. (2). We have proved that Φ_5-*check* satisfies this property by using the induction principle, but we omit the proof due to space restrictions.

V. Conclusion and Future Work

Summary: In this work, we present the formalisation of a subset of the traffic rules elicited from the Vienna Convention on Road Traffic for highways in Isabelle/HOL. We started by formalising the hybrid traces which can be regarded as an abstraction of the behaviours of autonomous vehicles obtained from the data recorded in a black box. We then formalised rules Φ_1–Φ_5 based on the notion of hybrid traces. Additionally, we also show the pattern for deriving the procedure to check each traffic rule.

Discussion: As can be seen from the formalisations, there is no explicit formula when we formalise the priority vehicles rule. We formalise this rule implicitly by adding a Boolean variable which is

1) true if the autonomous vehicle is a priority vehicle; and
2) false otherwise;

into the premise of rule Φ_2. Depending on how we interpret the priority vehicle rule, we can also add this boolean variable to the premise of rule Φ_3–Φ_5, as implied by the priority vehicle rules. However, we argue that being exempt from rule Φ_3–Φ_5 can cause damage to other road users. For example, if we exempt a priority vehicle from safe distance rule Φ_4, this could potentially cause collisions with other traffic participants (conflicting with rule Φ_1). Therefore, we do not add this status to the premise of other rules, except rule Φ_2.

There are also challenges in formalising the speed adaptation rule Φ_3 and the braking rule Φ_5. For rule Φ_3, it is especially difficult to concretise the notion of visibility. However, future speed limit information infrastructure could take into account these notions of visibility and road friction, and communicate them to autonomous vehicles. As for rule Φ_5, the main difficulty lies in the notion of "safety reasons" which is very abstract. Nevertheless, safety reasons can potentially be concretised by prediction under uncertainty using reachability analysis [12].

Conclusion: This work shows that we can make the traffic rules precise and unambiguous by concretising and formalising them. By formally specifying them, we can also derive a procedure for checking their compliance. This formal specification can be useful for technology providers as the basis for developing autonomous vehicles. Our approach can also be useful in terms of path planning since the procedure for checking each rule can be used to assess the safety (rule Φ_1) or the compliance with other rules.

Future Work: We wish to perform validation on the formalised traffic rules. This involves consulting with domain experts in the transportation community and legal experts. We also would like to enlarge the number of formalised traffic rules and procedures for checking them with a vision to obtain as many precise and unambiguous traffic rules as possible.

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.

Acknowledgements

The authors thank Tobias Nipkow and Fabian Immler for reviewing our Isabelle formalisation. We also gratefully acknowledge financial support by the German Research Foundation (DFG) Graduiertenkolleg 1480 (PUMA).

References

[1] Department for Transport, "The pathway to driverless cars: A detailed review of regulation for automated vehicle technologies," Great Minster House, 33 Horseferry Road, London SW1P 4DR, February 2015.

[2] M. Sergot, "Machine intelligence 11," J. E. Hayes, D. Michie, and J. Richards, Eds. New York, NY, USA: Oxford University Press, Inc., 1988, ch. Representing Legislation As Logic Programs, pp. 209–260.

[3] B. G. Buchanan and T. E. Headrick, "Some speculation about artificial intelligence and legal reasoning. stanford law review (1970): 40-62," *Stanford Law Review*, vol. 23, no. 1, pp. 40–62, 1970.

[4] S. J. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach*. Pearson Education, 2010.

[5] M. Sergot, "The representation of law in computer programs," in *Knowledge-Based Systems and Legal Applications*, ser. APIC, T. Bench-Capon, Ed. Elsevier Science, 1991.

[6] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory, "The British Nationality Act as a logic program," *Commun. ACM*, vol. 29, no. 5, pp. 370–386, May 1986.

[7] T. J. M. Bench-Capon, G. O. Robinson, T. W. Routen, and M. J. Sergot, "Logic programming for large scale applications in law: A formalisation of Supplementary Benefit legislation," in *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, ser. ICAIL '87. New York, NY, USA: ACM, 1987, pp. 190–198.

[8] A. Colmerauer and P. Roussel, "The birth of Prolog," in *The Second ACM SIGPLAN Conference on History of Programming Languages*, ser. HOPL-II. New York, NY, USA: ACM, 1993, pp. 37–52.

[9] L. T. McCarty, "Permissions and obligations," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'83. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1983, pp. 287–294.

[10] L. Royakkers, *Extending Deontic Logic for the Formalisation of Legal Rules*, ser. Law and Philosophy Library. Springer Netherlands, 1998.

[11] A. J. Jones and M. Sergot, "Deontic logic in the representation of law: Towards a methodology," *Artificial Intelligence and Law*, vol. 1, no. 1, pp. 45–64, 1992.

[12] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[13] W. Damm, H.-J. Peter, J. Rakow, and B. Westphal, "Can we build it: formal synthesis of control strategies for cooperative driver assistance systems," *Mathematical Structures in Computer Science*, vol. 23, pp. 676–725, 8 2013.

[14] S. Linker and M. Hilscher, "Proof theory of a Multi-Lane Spatial Logic," in *Theoretical Aspects of Computing – ICTAC 2013*, ser. Lecture Notes in Computer Science, Z. Liu, J. Woodcock, and H. Zhu, Eds. Springer Berlin Heidelberg, 2013, vol. 8049, pp. 231–248.

[15] S. M. Loos, A. Platzer, and L. Nistor, "Adaptive Cruise Control: Hybrid, distributed, and now formally verified," in *FM 2011: Formal Methods*, ser. Lecture Notes in Computer Science, M. Butler and W. Schulte, Eds. Springer Berlin Heidelberg, 2011, vol. 6664, pp. 42–56.

[16] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2283.

[17] L. Jiang and C. Yu, "Design and implementation of car black box based on embedded system," in *International Conference on Electrical and Control Engineering (ICECE) 2010*, June 2010, pp. 3537–3539.

[18] United Nations Economic Commission for Europe, "Vienna convention on road traffic," United Nations, Tech. Rep., Nov. 1968.

[19] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, "Highly automated driving on highways based on legal safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 333–347, March 2013.

[20] C. Baier and J. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[21] A. Cimatti, M. Roveri, and S. Tonetta, "Requirements validation for hybrid systems," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, A. Bouajjani and O. Maler, Eds. Springer Berlin Heidelberg, 2009, vol. 5643, pp. 188–203.

[22] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 420–425.

[23] E. Weiß, "Set-based prediction of interacting road vehicles," Bachelor Thesis, Technische Universität München, January 2015.

Authorized licensed use limited to: DGIST. Downloaded on March 13,2023 at 02:17:38 UTC from IEEE Xplore. Restrictions apply.