Spatial Reasoning about Motorway Traffic Safety with Isabelle/HOL*

Sven Linker

Department of Computer Science, University of Liverpool, UK
s.linker@liverpool.ac.uk

Abstract. Formal verification of autonomous vehicles on motorways is a challenging problem, due to the complex interactions between dynamical behaviours and controller choices of the vehicles. In previous work, we showed how an abstraction of motorway traffic, with an emphasis on spatial properties, can be beneficial. In this paper, we present a semantic embedding of a spatio-temporal multi-modal logic, specifically defined to reason about motorway traffic, into Isabelle/HOL. The semantic model is an abstraction of a motorway, emphasising local spatial properties, and parameterised by the types of sensors deployed in the vehicles. We use the logic to define controller constraints to ensure safety, i.e., the absence of collisions on the motorway. After proving safety with a restrictive definition of sensors, we relax these assumptions and show how to amend the controller constraints to still guarantee safety.
Keywords. Spatial logic, Isabelle, interactive theorem proving, motorway traffic, verification, safety.

1 Introduction

Due to the current and ongoing interest in autonomous vehicles, proving that such vehicles will behave correctly is of growing importance. Since vehicles are complex, dynamical systems, proving properties about them often involves solving differential equations, where spatial elements, e.g., position and braking distance, are functions of time. However, safety is fundamentally a spatial property: the absence of collisions, i.e., no two vehicles occupy the same space.

To overcome the complexities of proving safety properties, we proposed to separate the dynamical behaviour from the concrete changes in space [1]. To that end, we defined *Multi-Lane Spatial Logic* (MLSL), which was used to express guards and invariants of controller automata defining a protocol for safe lane-change manoeuvres. Under the assumption that all vehicles adhere to this protocol, we proved that collisions were avoided. Subsequently, we presented an extension of MLSL to reason about changes in space over time, a system of natural deduction, and formally proved a safety theorem [2,3]. This proof was carried out manually and dependent on strong assumptions about the vehicles' sensors.

* This work was supported by the EPSRC Research Programme EP/N007565/1 *Science of Sensor System Software (S4)*.

In this paper, we define a semantic embedding of a further extension of MLSL into the theorem prover Isabelle/HOL [4]. That is, we present the first tool to mechanically assist reasoning with MLSL. Subsequently, we show how the safety theorem can be proved within this embedding. Finally, we alter this formal embedding by relaxing the assumptions on the sensors. We show that the previously proven safety theorem does *not* ensure safety in this case, and how the controller constraints can be strengthened to guarantee safety.[1]

Recently, many approaches to verify traffic safety have been published. A main distinction between them is the way they abstract properties of traffic. Loos et al. used the theorem prover KeYmaera [5] to verify safety of motorway scenarios [6]. The underlying logic of KeYmaera is *Differential Dynamic Logic* [7], where the dynamical behaviour of systems is explicitly encoded within hybrid programs. This contrasts with our approach, where the main focus is on spatial aspects of traffic. However, they abstract away from the way real vehicles change lanes, i.e., vehicles may change to any lane, not only adjacent ones, in one step. We restrict the possibilities of lane changes to exactly the adjacent lanes.

Rizaldi and Althoff presented a formal implementation of traffic rules [8]. Similar to our work, they choose Isabelle/HOL to analyse several laws from the Vienna Convention on Road Traffic. However, they focus on whether the behaviour of vehicles is compliant with these laws. Our formalisation does not take legal issues into account, and concentrates only on the absence of collisions.

The distinction between dynamical behaviour and a higher-level is not unique to our work. Kamali et al. [9] used a combination of the *Belief-Desire-Intention* approach to model agents, and Timed Automata [10]. They distinguish between the planning component of a vehicle and its underlying dynamics. The planning component creates the new intentions of a vehicle according to its current belief about the situation on the road, and its general desires. The underlying dynamics then implement the plan constructed by the planning component. Both components can be verified on their own, the planning component with the model checker AJPF [11], and the dynamics with Uppaal [12]. Our spatial abstraction could serve as a middle tier between their planning component and the dynamics, by abstracting concrete values (e.g., distances) to spatial properties.

In a similar fashion, Campbell et al. used π-calculus processes to define and reason about the communication structure of vehicle networks [13]. The lower level dynamics are implemented as Hybrid Automata [14], and the connection between both levels is given by connecting the messages in the higher level with input and output messages of the automata. Our results imply that the amount of necessary communication between vehicles depends on sensor capabilities of each vehicle. Hence our results could inform the instantiations of their models.

The structure of our paper is as follows. Section 2 presents the semantic embedding of MLSL into Isabelle/HOL. In Sect. 3 and Sect. 4 we discuss the proofs for safety with different sensor capabilities. Section 5 concludes the paper.

[1] The code of the formalisation can be found at `www.github.com/svenlinker/HMLSL`. It is compatible with Isabelle2016-1.

2 Embedding MLSL into Isabelle/HOL

In this section, we present our abstraction of motorway traffic, as well as *Hybrid Multi-Lane Spatial Logic* (HMLSL), an extension of *Multi-Lane Spatial Logic* (MLSL), by introducing concepts from *Hybrid Logic* [15] and universal modalities. In the majority of the paper, we will only present the formalisation within Isabelle, but explain the relation to previous work [1,2,3].

Notations. Isabelle/HOL is based on type theory, hence every term t has a specific type τ, denoted by $t :: \tau$. The type of a function from τ to τ' is written as $\tau \Rightarrow \tau'$. Within Isabelle, we have to distinguish between the meta-logic and the object logic. In the case of Isabelle/HOL, both are instantiations of Higher-Order logic. Implication and equivalence of the meta-logic is denoted by $\Longrightarrow$ and $\equiv$, respectively. They are generally used to define terms. The object level implication is $\longrightarrow$, which is used within lemmas and theorems. In this paper, conjunction, disjunction and existential quantification will generally be used within the object logic, denoted by the operators $\wedge$, $\vee$, and $\exists$. Finally, function application will typically be denoted without parentheses, i.e., instead of $f(t)$, we will write $f\ t$.

2.1 Semantic Model

The semantics of HMLSL reflects situations as depicted in Fig. 1. That is, we consider vehicles driving on a motorway with possibly several lanes. All vehicles are assumed to drive in one direction (to the right in the figure). The *safety envelope* comprises the physical size of c as well as the distance needed for an emergency braking. Within the model, we distinguish between two spatial properties of vehicles. The *reservation* of a vehicle c is the part of the motorway that c currently drives on, defined by the lanes c uses and its safety envelope. Reservations may occupy space on up to two adjacent lanes, which indicates that the vehicle is currently performing a lane-change manoeuvre, see, e.g., vehicle a in Fig. 1. A *claim*, depicted by the dotted lines in the figure, is a formalisation of setting the turn signal, i.e., it is an indicator that c wants to change its lane. Vehicles may only hold claims while not engaged in a lane change, i.e., as long as the reservation only contains space on a single lane. A claim of a vehicle is always adjacent to its reservation and of the same length.

The semantic model we use is twofold. We use *traffic snapshots* to formalise the current situation on the whole motorway. The motorway is of infinite length, modelled by the real numbers, and consists of an arbitrary, but fixed number of discrete lanes. Furthermore, we assume an infinite number of vehicles, each of which has a position and dynamic behaviour, e.g., its velocity and current acceleration. On top of the snapshots, *views* denote a finite part of the motorway perceived by a vehicle. To that end, they consist of a closed real-valued interval, and a finite discrete interval of lanes. Each view is associated with a distinct vehicle, its *owner*. In Fig. 1, the traffic snapshot contains three lanes. A possible view v of the vehicle e is depicted as a dashed rectangle, and contains the two lower lanes. While both vehicle a and e are fully contained in v, only the safety

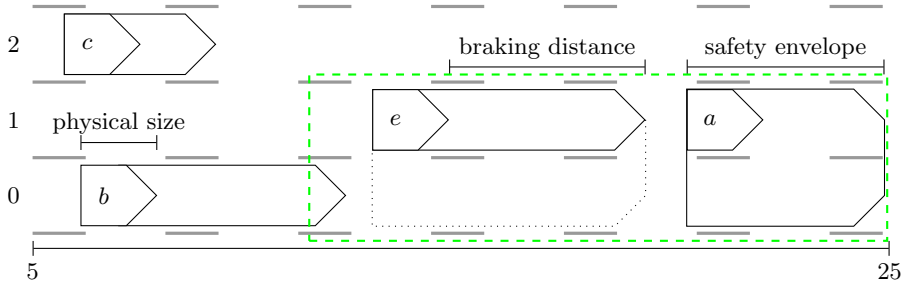

Fig. 1. Situation on a Motorway at a Single Point in Time

envelope of vehicle b is within this view. If we assume an idealised world, where each vehicle can perceive the full safety envelope of other vehicles, i.e., both their physical size and braking distance, then e can sense the presence of b. We call this type of information *perfect*. Of course, this assumption is rather strong. If we assume that vehicles know about their own safety envelope, but only about the physical size and position of other vehicles, e cannot perceive b. We will refer to this situation as *regular information* [1].

2.2 Preliminary Definitions

Formally, we introduce two new types, one for real-valued intervals and another for discrete intervals. For real valued intervals, we use the type *real_int*, which is a tuple of two real values (x, y), with the condition $x \leq y$. The discrete intervals use a definition within the *Main* library of Isabelle to define a consecutive sequence of numbers between m and n. If $m > n$, this will result in the empty set.

$$\textbf{typedef}\ real\_int = \{r :: (real * real).fst\ r \leq snd\ r\}$$
$$\textbf{typedef}\ nat\_int = \{i.(\exists (m :: nat) n.\{m..n\} = i)\}$$

For both of these types, we define several auxiliary functions and predicates. The function *right* (*left*) returns the right (left, resp.) end point of a real-valued interval. We define a partial order on *real_int* to denote subintervals, i.e., $r \leq s$ if, and only if, $left\ r \geq left\ s$ and $right\ r \leq right\ s$. Within Isabelle, we define this relation and show that *real_int* instantiates the abstract class *order*, i.e., we show reflexivity, transitivity and anti-symmetry. For *nat_int*, we prove more structure. We define the infimum $i \sqcap j$ of two intervals i and j by lifting set intersection to *nat_int*. Similarly, we can lift the subset relation on sets to *nat_int*, to constitute a partial order $\sqsubseteq$ with a least element, the empty set. Since discrete intervals are not closed under arbitrary unions, we introduce a new predicate $consec\ i\ j$, to denote that two intervals are non-empty and $\max(i) + 1 = \min(j)$. We can then define $i \sqcup j$ as the union of i and j. Furthermore, we need *measures* for both types of intervals. For discrete intervals, the measure is its cardinality lifted from sets, while the measure for real valued intervals is the difference between the left and right end points, i.e., $\|r\| \equiv right\ r - left\ r$.

Furthermore, we introduce the notion of *chopping* an interval into two subintervals. The predicate $R\_Chop(r, s, t)$ is similar to the chopping operation of Interval Temporal Logic [16]. For discrete intervals, we implemented a ternary predicate $N\_Chop(i, j, k)$, which was taken from previous work [2,3].

$$R\_Chop(r, s, t) \equiv left\ r = left\ s \wedge right\ s = left\ t \wedge right\ r = right\ t$$
$$N\_Chop(i, j, k) \equiv i = j \sqcup k \wedge (j = \emptyset \vee k = \emptyset \vee consec\ j\ k)$$

Finally, we get a countably infinite type *cars* by a bijection on natural numbers.

2.3 Views

Using these definitions, we construct a type *view* as a record of three elements: a real-valued interval modelling the *extension* along the lanes, a discrete interval denoting the perceived *set of lanes*, and an identifier for the *owner* of the view.

$$\textbf{record}\ view = ext :: real\_int \quad lan :: nat\_int \quad own :: cars$$

We lift the chopping on intervals to views. For example, *horizontal chopping*, i.e., dividing the extension of the view while keeping the set of visible lanes and the owner, is defined as follows.

$$v = u \parallel w \equiv R\_Chop(ext\, v, ext\, u, ext\, w) \wedge lan\, v = lan\, u \wedge lan\, v = lan\, w \wedge own\, v = own\, u \wedge own\, v = own\, w$$

The functions *lan* , *ext* and *own* are automatically generated by Isabelle, to refer to the respective parts of the views. The predicate $v = u\text{--}w$ denotes *vertical chopping*. Furthermore, we introduce a relation $v = c > u$ to change the owner of the view v to c, while keeping the spatial borders.

$$(v = c > u) \equiv ext\, v = ext\, u \wedge lan\, v = lan\, u \wedge own\, u = c$$

We can prove several lemmas about views and their relationships. For example, if we can chop a view v vertically into u and w and can switch v to a view v' with the owner c, we can chop v' into counterparts to u and w.

$$\textbf{lemma}\ v = u\text{--}w \wedge v = c > v' \longrightarrow (\exists u'\ w'.u = c > u' \wedge w = c > w' \wedge v' = u'\text{--}w')$$

2.4 Traffic Snapshots

The formalisations of the underlying traffic situations, called *traffic snapshots*, have to capture the intuitions given in Sect. 2.1, i.e., reservations, claims, positions, physical sizes, braking distances and the dynamical behaviour of vehicles. For all of these, we use functions whose domain is the type *cars*. Since the definitions for traffic snapshots are long, but straightforward, we mostly refrain from providing the Isabelle code, but describe the formal concepts. Reservations and claims are given by the functions $res, clm\colon cars \Rightarrow nat\_int$, positions, physical sizes and braking distances are given by $pos, ps, bd\colon cars \Rightarrow real$. The dynamic behaviours over time, i.e., the increases in the cars' positions, are given by a real-valued function for each vehicle: $dyn\colon cars \Rightarrow (real \Rightarrow real)$. Traffic snapshots are tuples $ts = (pos, res, clm, dyn, ps, bd)$, with the following conditions:

1. $res\ c \cap clm\ c = \emptyset$,
2. $|res\ c| \geq 1$,
3. $|res\ c| \leq 2$,
4. $|clm\ c| \leq 1$,
5. $|res\ c| + |clm\ c| \leq 2$,
6. $clm\ c \neq \emptyset \longrightarrow$
   $\exists n\colon res\ c \cup clm\ c = \{n, n+1\}$,
7. $ps\ c > 0$,
8. $bd\ c > 0$.

Conditions 1-6 are the *sanity conditions* from previous work [2,3], that vehicles have to respect to be spatially well-defined. For example, we require reservations and claims to be adjacent, that vehicles have at most one claim, and so forth. Condition 7 denotes that vehicles have to be physically present (even though they may be arbitrarily small), while 8 ensures that a vehicle needs some leading safe space. Subsequently, we will refer to the reservation function of a traffic snapshot ts by $res\ ts$, and also respectively notate the other functions.

Example 1. The traffic situation ts in Fig. 1 can be formalised as follows.

$$\begin{array}{llll} pos\ ts\ a = 22 & pos\ ts\ b = 7 & pos\ ts\ c = 6 & pos\ ts\ e = 17 \\ res\ ts\ a = \{0,1\} & res\ ts\ b = \{0\} & res\ ts\ c = \{2\} & res\ ts\ e = \{1\} \\ clm\ ts\ a = \emptyset & clm\ ts\ b = \emptyset & clm\ ts\ c = \emptyset & clm\ ts\ e = \{0\} \\ bd\ ts\ a = 3 & bd\ ts\ b = 6 & bd\ ts\ c = 2 & bd\ ts\ e = 6 \end{array}$$

As an example, we further set $ps\ ts\ d = 1$ and $dyn\ ts\ d\ x = \frac{1}{2} \cdot a_d \cdot x^2$ for all vehicles d. That is, we assume that each vehicle has its own acceleration a_d. The view v indicated by the dashed rectangle is given by $ext\ v = (13, 25)$, $lan\ v = \{0, 1\}$ and $own\ v = e$. Observe that the concrete values of the functions are less important than the relations between them. In particular, we do not instantiate dyn in any proofs in this paper, and use it as an abstraction of the cars' dynamics.

Between two traffic snapshots ts and ts', different *global* and *local* transitions are possible. The only type of global transition is the passing of time, i.e., ts' is the result of purely dynamical behaviour of vehicles, starting at ts. The passing of x time units is denoted by $ts{-}x{\rightarrow}ts'$, during which only the vehicles' position is updated according to their dynamic behaviour, with the precondition that $dyn\ ts\ c\ y \geq 0$ for all c and $0 \leq y \leq x$. This ensures that cars only drive forward. Furthermore, single vehicles can perform *local* transitions. A vehicle c can

1. *create a new claim*, residing on a lane adjacent to its current reservation, which may only consist of a single lane, denoted by $ts{-}c(c,n){\rightarrow}ts'$,
2. *create a new reservation*, i.e., it has to currently possess a claim and mutates this claim to a reservation, denoted by $ts{-}r(c){\rightarrow}ts'$,
3. *withdraw its claim*, i.e., remove a currently existing claim from the road, denoted by $ts{-}wdc(c){\rightarrow}ts'$,
4. *withdraw a reservation*, i.e., if its current reservation comprises two lanes, c shrinks its reservation to a single lane, denoted by $ts{-}wdr(c,n){\rightarrow}ts'$, or
5. *adjust its dynamics*, i.e., change the function responsible for its dynamic behaviour to a given function $f\colon real \to real$, denoted by $ts{-}dyn(c,f){\rightarrow}ts'$.

All of these relations can be straightforwardly defined using the notion of traffic snapshots. For example, we define creation of a claim as follows.[2]

$$\begin{aligned} ts{-}c(c,n){\rightarrow}ts' \equiv\ & (pos\ ts') = (pos\ ts) \wedge (res\ ts') = (res\ ts) \\ & \wedge (dyn\ ts') = (dyn\ ts) \wedge (ps\ ts') = (ps\ ts) \wedge (bd\ ts') = (bd\ ts) \\ & \wedge |clm\ ts\ c| = 0 \wedge |res\ ts\ c| = 1 \\ & \wedge ((n+1 \in res\ ts\ c) \vee (n-1 \in res\ ts\ c)) \\ & \wedge (clm\ ts') = (clm\ ts)(c := Abs\_nat\_int\{n\}) \end{aligned}$$

The definition ensures that except for the claim of c, all parts of ts are equal to their counterparts in ts'. Furthermore, it requires that within ts, the vehicle c may only possess a single reservation, and no claim at all. The claim on lane n may only be created, if the reservation consists of a lane adjacent to n. Finally, the claims in ts' are the claims in ts, except for the newly created claim of c.

With these relations, we create two additional types of transition. *Evolutions* consists of arbitrary sequences of time passing and dynamic adjustments. We denote the evolution from ts to ts' by $ts \rightsquigarrow ts'$. Within Isabelle, we use an inductive definition to enable reasoning about evolutions. An *abstract transition* is an arbitrary transition sequence between ts and ts'. We denote such sequences by $ts \Rightarrow ts'$. Similarly to evolutions, we can define abstract transitions inductively.

Example 2. Consider again the traffic snapshot ts depicted in Fig. 1. The vehicle b can create a claim on lane 1, since its reservation contains only the lane 0. That is, there is a ts', such that $ts{-}c(b,1){\rightarrow}ts'$. However, there is no possibility for b to create a claim on lane 2.

Since views are intended to be relative to their owner, we have to consider the position of a view if the owner moves. Let v be a view with owner e. If time passes between snapshots ts and ts', we have to compute the difference between the position of e in ts and ts' and add it to the borders of the extension of v. Within Isabelle, we define a suitable function $move\ ts\ ts'\ v$.

2.5 Sensors

The preceding definitions are independent of the types of sensor the vehicles possess. The sensors, however, define the information each vehicle may use to decide, whether manoeuvres on the road can be safely performed, e.g., a lane change manoeuvre. We parameterise our model with a function representing the distances obtained from the sensors, i.e., a function returning the perceived length of a vehicle c by a vehicle e at the current traffic snapshot ts.

$$sensors :: cars \Rightarrow traffic \Rightarrow cars \Rightarrow real$$

[2] The function *Abs_nat_int* takes a set of natural numbers as its input, and returns an element of type *nat_int*. It is automatically created by Isabelle as a result of the type definition in Sect. 2.2. Subsequently, we will silently omit these functions.

We require *sensors* to return a non-zero length for each vehicle. That is, for all vehicles e, c and all traffic snapshots ts, we have $sensors\ e\ ts\ c > 0$. Using the sensor definition as a parameter implies that all vehicles use the same definition of the sensor function. In general, however, this function can be as complicated as necessary. We then define the space used by a vehicle c as observed by e.

$$space\ ts\ v\ c\ \equiv (pos\ ts\ c,\ pos\ ts\ c + sensors\ (own\, v)\ ts\ c)$$

2.6 Restriction to Views

Our intention when using views together with traffic snapshots is to limit the space a vehicle can perceive at any time, since it can only take a limited amount of information into account. We need to restrict the perceived length of a vehicle to the view, as well as the lanes used for claims and reservations.

We denote the perceived length of a vehicle c by the owner of a given view v by $len\ v\ ts\ c$. Consider Fig. 1, and the indicated view v owned by the vehicle e. For the vehicles e and a, we intend that *space* and *len* coincide on v. However, for c, we have to ensure that *len* is empty, since it drives outside of v. The size of *len* for b depends on the type of information we assume: with perfect information, we want that *len* is not empty and describes the small part of the safety envelope of b within in v, while with regular information, we intend that *len* returns an empty interval. We therefore define the perceived length as follows.

$$\begin{aligned} len\ v\ ts\ c \equiv\ & \textbf{if}\ (left\ (space\ ts\ v\ c)) > right\ (ext\, v)) \\ & \textbf{then}\ (right\ (ext\, v), right\ (ext\, v)) \\ & \textbf{else if}\ (right\ (space\ ts\ v\ c) < left\ (ext\, v)) \\ & \quad \textbf{then}\ (left\ (ext\, v), left\ (ext\, v)) \\ & \quad \textbf{else}\ (\max\ (left\ (ext\, v))\ (left\ (space\ ts\ v\ c)), \\ & \qquad \min\ (right\ (ext\, v))\ (right\ (space\ ts\ v\ c))) \end{aligned}$$

The first two cases ensure that vehicles not visible in the view v (either to the left or to the right) will be represented by an empty interval. The last case is defined such that *len* is always a sub-interval of the extension of the view.

We have proved several properties about *len* needed in the safety proofs. For example, if the perceived length of a vehicle fills the extension of a given view, then it does the same for the horizontal sub-views.

$$\textbf{lemma}\ len\ v\ ts\ c = ext\, v \wedge v = u \parallel w \longrightarrow len\ u\ ts\ c = ext\, u$$
$$\textbf{lemma}\ len\ v\ ts\ c = ext\, v \wedge v = u \parallel w \longrightarrow len\ w\ ts\ c = ext\, w$$

The restriction of the claims and reservations to a view is the intersection with the lanes visible in the view. Within Isabelle, we use the following definition.

$$restrict\ v\ f\ c \equiv (f\ c) \sqcap lan\, v$$

To use this function we partially evaluate one of the functions *res* or *clm*. For example, the restriction of reservations contains at most two lanes at any time.

$$\textbf{lemma } |\mathit{restrict}\ v\ (\mathit{res}\ ts)\ c| \leq 2$$

However, most properties of *restrict* hold for any possible function from *cars* to *lanes*. E.g., if a view v can be vertically chopped into sub-views u and w, the restriction of a function f to v is the union of the restriction of f on u and w.

$$\textbf{lemma } v = u \text{--} w \longrightarrow \mathit{restrict}\ v\ f\ c = \mathit{restrict}\ u\ f\ c \sqcup \mathit{restrict}\ w\ f\ c$$

2.7 Hybrid Multi-Lane Spatial Logic

The logic HMLSL is a modal extension of first-order logic. In addition to first-order operators, HMLSL contains two spatial predicates $re(c)$ and $cl(c)$, which are true, if, and only if, the current view consists of a single lane that is completely filled with the reservation of the vehicle denoted by c (or its claim, respectively). To reason about views with more lanes, and different topological relations between vehicles, we can *chop* views either horizontally with the binary modality $\frown$, or vertically using $\smile$. Intuitively, $\phi \frown \psi$ splits the extension of a view into two disjoint sub-views, where ϕ holds on the left interval and ψ on the right, while the set of lanes and the owner is kept. For each type of spatial transition $*(c)$, we use a family of modalities $\Box*(c)$. I.e., the modalities are parameterised and this parameter will be evaluated like other variables in the formulas. Furthermore, we use a single modality to refer to evolutions between snapshots, i.e., the passing of time and changes in the dynamical behaviour of the vehicles. The universal modality $\mathbf{G}$ is defined with respect to abstract transitions, i.e., it can be used to define invariance properties. Finally, we employ a modality $@c$ in the fashion of Hybrid Logic (HL) [15]. In HL, $@c$ is used to switch to the world c, regardless of the accessibility relation of the logic. Within MLSL, we use $@c$ to exchange the owner of the current view, which allows to reason about different perspectives on parts of the motorway. The information we have at our disposal may change for different perspectives, depending of the type of sensors in the vehicles. For a given view v, while we evaluate the formula $@c\ \varphi$, we switch to a view v' with the same extension and lanes as v, but whose owner is c.

Definition 1 (Syntax of HMLSL). *The syntax of formulas of the* hybrid multi-lane spatial logic *is given as follows, where c, d are variables of type cars:*

$$\phi ::= \bot \mid c = d \mid re(c) \mid cl(c) \mid \phi_1 \rightarrow \phi_2 \mid \forall c \bullet \phi_1 \mid \phi_1 \frown \phi_2 \mid \phi_1 \smile \phi_2 \mid M\phi$$

where $M \in \{\Box\mathsf{r}(c), \Box\mathsf{c}(c), \Box\mathsf{wd\ c}(c), \Box\mathsf{wd\ r}(c), \Box\tau, \mathbf{G}, @c\}$*, and c, d are variables.*

To define HMLSL within Isabelle, we follow an approach of Benzmüller and Paulson to embed quantified multi-modal logics into HOL [17]. Essentially, we encode formulas as functions from the set of worlds to truth values. Since our semantic model consists of both views and traffic snapshots, we define the formulas

of HMLSL to be functions taking both of these entities as parameters, i.e., we translate them directly into HOL. This allows for a natural definition and notation of the operators, while still enabling us to use the automatic proof methods of Isabelle. For brevity, we define a type synonym $\sigma = \mathit{traffic} \Rightarrow \mathit{view} \Rightarrow \mathit{bool}$.

Most operators combine several terms of type σ, and return a new term of type σ. For example, negation is of type $\sigma \Rightarrow \sigma$. Conjunction and the chopping modalities have the type $\sigma \Rightarrow \sigma \Rightarrow \sigma$, since they are just binary connectives. The box modalities, however, also take a vehicle as a parameter, i.e., their type is $\mathit{cars} \Rightarrow \sigma \Rightarrow \sigma$. Due to space limitations, we only provide some examples.

$$\begin{aligned}
\neg\varphi &\equiv \lambda\ ts\ v.\neg\varphi(ts)(v)\\
\varphi \frown \psi &\equiv \lambda\ ts\ v.\exists u\ w.(v = u \parallel w) \wedge \varphi(ts)(u) \wedge \psi(ts)(w)\\
\Box c(c)\ \varphi &\equiv \lambda\ ts\ v.\forall ts'\ n.(ts - c(c,n) \rightarrow ts') \longrightarrow \varphi(ts')(v)\\
\mathbf{G}\,\varphi &\equiv \lambda\ ts\ v.\forall ts'.(ts \Rrightarrow ts') \longrightarrow \varphi(ts')(\mathit{move}\ ts\ ts'\ v)\\
@c\ \varphi &\equiv \lambda\ ts\ v.\forall u.(v = c > u) \longrightarrow \varphi(ts)(u)
\end{aligned}$$

To avoid confusion with the object logic of Isabelle/HOL, we use bold symbols for the operators of HMLSL. While the Boolean operators are just translations to operators of HOL, the operators specific to HMLSL refer to the elements of the models given in the previous section. E.g., the semantics of the chop modalities refer to the chopping operations of Sect. 2.3. The behavioural modalities use the transition relations of Sect. 2.4, e.g., the modality **G** is defined with respect to all abstract transitions leaving the current traffic snapshot. The semantics of atomic formulas refers to the measures of intervals and restrictions to views.

$$\begin{aligned}
\mathit{re}(c) \equiv \lambda\ ts\ v.\ &\mathit{len}\ v\ ts\ c = \mathit{ext}\,v \wedge \mathit{restrict}\ v\ (\mathit{res}\ ts)\ c = \mathit{lan}\,v\\
&\wedge \|\mathit{ext}\,v\| > 0 \wedge |\mathit{lan}\,v| = 1
\end{aligned}$$

These abbreviations correspond directly to the original definitions of MLSL [1,2]. Furthermore, we can define the *somewhere* modality as an abbreviation.

$$\langle\varphi\rangle \equiv \top \frown (\top \smile \varphi \smile \top) \frown \top$$

Finally, we also introduce notions for validity and satisfaction, which allow us to state lemmas comfortably, but can also be used within proofs of these lemmas.

$$\models \varphi \equiv \forall ts.\forall v.\varphi(ts)(v) \qquad\qquad ts, v \models \varphi \equiv \varphi(ts)(v)$$

We prove several lemmas to show that the definitions work as intended. For example, somewhere distributes over disjunction, which can be proven by a single application of the *blast* proof method. Furthermore for each vehicle, there can be at most two reservations visible anywhere on the motorway. Finally, we show how the transitions to create reservations are connected to the claims and reservations on the road. The proof of these lemmas need manual intervention, but mainly to guide the automatic methods.

$$\begin{aligned}
&\textbf{lemma}\ \models \langle\varphi \vee \psi\rangle \leftrightarrow \langle\varphi\rangle \vee \langle\psi\rangle\\
&\textbf{lemma}\ \models \neg\langle \mathit{re}(c)\rangle \smile \langle \mathit{re}(c)\rangle \smile \langle \mathit{re}(c)\rangle\\
&\textbf{lemma}\ \mathit{reservation} : \models (\Box r(c)\ \mathit{re}(c)) \leftrightarrow (\mathit{re}(c) \vee \mathit{cl}(c))
\end{aligned}$$

3 Safety with Perfect Information

In this section, we instantiate the sensor function of the semantic model such that each vehicle possesses ideal and unrestricted sensors and can thus obtain *perfect information* of the space visible in its view. Formally, the sensor function consists of the sum of the physical size of a vehicle and its safety distance.

$$sensors\ e\ ts\ c \equiv ps\ ts\ c + bd\ ts\ c$$

Observe that the sensors do not distinguish between the owner of the view and any other vehicle. That is, they always return the full safety envelope of a vehicle.

Safety in our model is modelled by the absence of overlapping reservations. That is, our safety predicate can be defined as follows.

$$safe\ e \equiv \forall c.\neg(c = e) \rightarrow \neg\langle re(c) \wedge re(e)\rangle\text{`}$$

To restrict the allowed behaviour of vehicles on the road, we require them to adhere to certain protocol specifications. Vehicles have to respect reservations as long as they only drive on the road without changing lanes, i.e., during evolutions. This is ensured by the *distance controller* DC.

$$DC \equiv \mathbf{G}\,(\forall c\ d.\neg(c = d) \rightarrow \neg\langle re(c) \wedge re(d)\rangle \rightarrow \Box\tau\ \neg\langle re(c) \wedge re(d)\rangle)$$

Intuitively, DC ensures that two different vehicles c and d, whose reservations do not overlap initially, will keep their distances so that no overlap occurs, as long as only time passes and dynamics are adjusted.

The only transition after which new reservations appear on the road is the creation of reservations. Observe that a unsafe situation can only occur, if there was already a claim overlapping with a reservation before the transition happened. Hence we have to forbid the creation of reservations in this case. To that end, we define the *potential collision check*.

$$pcc\ c\ d \equiv \neg(c = d) \wedge \langle cl(d) \wedge (re(c) \vee cl(c))\rangle$$

Finally, the *lane change controller* restricts the vehicles such that if a vehicle holding a claim created a reservation, while a potential collision exists, we would get a contradiction. Hence, such a transition cannot occur.

$$LC \equiv \mathbf{G}\,(\forall d.(\exists c.pcc\ c\ d) \rightarrow \Box r(d)\ \bot)$$

Observe that this formula is slightly more restrictive than necessary. The potential collision check is already satisfied, if two claims overlap, which does not immediately lead to overlapping reservations, if only one of the vehicles changes the claim to a reservation. That is, in a model with interleaving semantics, as we defined in Sect. 2.4, we could reduce this check to only be satisfied, if the claim overlaps with a reservation. However, the given formula even ensures safety, if we allowed for synchronous creation of reservations [18].

Our safety theorem is as follows. If the initial situation is safe, and all vehicles adhere to DC and LC, safety is an invariant along all possible transitions.

$$\textbf{theorem}\ safety : \models (\forall e.safe\ e) \wedge DC \wedge LC \rightarrow \mathbf{G}\,(\forall e.safe\ e)$$

Proof. We only present a proof sketch, since the proof itself consists of roughly 200 lines of Isar proof script. We fix an arbitrary traffic snapshot ts and view v, and proceed by induction on transition sequences $ts \Rightarrow ts'$. The base case follows by the assumption $\forall e.safe\ e$. The induction step consists of a case distinction for the different transition types, where we assume that $ts \Rightarrow ts'$ holds for some ts' and $ts', v \models \forall e.safe\ e$. In all cases, we prove the theorem by contradiction.

For evolutions, fix a ts'' with $ts' \rightsquigarrow ts''$ and $ts'', move\ ts\ ts''\ v \models \neg\forall e.safe\ e$. That is, there are c and e, such that $ts'', move\ ts'\ ts''\ v \models \langle re(c) \wedge re(e)\rangle$. By the induction hypothesis and DC, we get $ts', move\ ts\ ts'\ v \models \Box_\tau\ \neg\langle re(c) \wedge re(e)\rangle$, and thus $ts'', move\ ts\ ts''\ v \models \neg\langle re(c) \wedge re(e)\rangle$. This yields the contradiction.

For creations of reservations, fix d and ts'', such that both $ts{-}r(d){\rightarrow}ts''$ and $ts'', move\ ts\ ts''\ v \models \neg\forall e.safe\ e$. That is, there are c and e, such that $ts'', move\ ts\ ts''\ v \models \langle re(c) \wedge re(e)\rangle$. Subsequently, we have to distinguish the cases whether $d = c$ or $d = e$, or neither. In the latter case, we have that the overlap exists on ts' as well and get a contradiction. The other two cases are similar, and we only discuss the case $d = e$. In this case, we get that on ts', a claim or a reservation of e was overlapping with the reservation of c, i.e., $ts', move\ ts\ ts'\ v \models (\langle re(c){\wedge}re(e)\rangle {\vee} \langle re(c){\wedge}cl(e)\rangle)$. The first case contradicts the induction hypothesis. The latter case implies $ts', move\ ts\ ts'\ v \models \langle re(c){\wedge}(re(e){\vee}\ cl(e))\rangle$ This is exactly the potential collision check $pcc\ e\ c$. With LC, we get the contradiction. The other cases are all proved in similar ways, by concluding that the overlap existed on ts', contradicting the induction hypothesis. □

The safety theorem states that our controllers ensure safety, from the perspective of a single vehicle, since we never employ the hybrid modality $@c$. However, with our assumption of perfect knowledge, we can prove the following theorem, which shows that switching to a different owner does not impact safety.

$$\textbf{lemma}\ \models (\forall e.safe\ e) \rightarrow @c\ (\forall e.safe\ e)$$

Hence, no vehicle perceives a collision, which implies that safety is an invariant along all transitions for all vehicles.

4 Safety with Regular Information

In this section, we discuss how the proof given previously is affected, if we assume regular sensors. That is, while a vehicle can compute its own braking distance, it is not able to refer to the braking distance of other vehicles. However, we assume that the sensors can identify the physical size of other vehicles.

$$sensors\ e\ ts\ c\ \equiv \textbf{if}\ (e = c)\ \textbf{then}\ ps\ ts\ c + bd\ ts\ c\ \textbf{else}\ ps\ ts\ c$$

Hence, each vehicle e has complete information about its own safety envelope (the sum of its physical size and braking distance), but does not know anything about the braking distance of other vehicles. Note that the sensor function is a global parameter of HMLSL, i.e., all vehicles use the same function. With this sensor definition, we can still proceed to prove the safety theorem given in

Sect. 3. However, since we neither refer to views with different owners in the safety property, nor in the theorem itself, we cannot prove the invariance of safety if we switch owners. Instead, we can prove the following lemma.

$$\textbf{lemma}\ \exists ts\ v.\ ts, v \models \forall e.safe\ e \wedge (\exists c.@c\ \neg(\forall e.safe\ e))$$

The proof consists of a straightforward, but tedious, construction of a suited traffic snapshot ts and view v. The essential parts of ts and v are shown in Fig. 2. Vehicle e is currently engaged in a lane change, while the vehicle c drives behind e on one lane. The view v indicated by the dashed rectangle is owned by e, hence e can only perceive the physical size of c, and not its full safety envelope, denoted by the dashed lines in front of c. For e, the situation seems perfectly safe, since the part of c visible to e is disjoint from e's reservation. In particular, we get $ts, v \models \forall e.safe\ e$. However, if we switch the view to be owned by c, we get overlapping reservations, i.e., we also have $ts, v \models \exists c.@c\ \neg(\forall e.safe\ e)$.

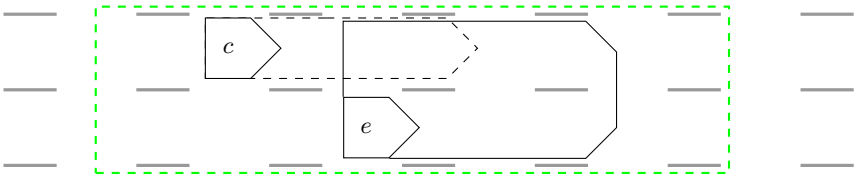

Fig. 2. Unsafe Situation with Regular Information

We can amend our controller specification, however, to also take the perspective of other vehicles into account.

$$DC' \equiv \mathbf{G}\ (\forall c\ d.\neg(c = d) \rightarrow @d\ \neg\langle re(c) \wedge re(d)\rangle \rightarrow \Box\tau\ @d\ \neg\langle re(c) \wedge re(d)\rangle)$$
$$LC' \equiv \mathbf{G}\ (\forall d.(\exists c.@c\ (pcc\ c\ d) \vee @d\ (pcc\ c\ d)) \rightarrow \Box r(d)\ \bot)$$

Note that withinthe distance controller, we still only refer to the perspective of a single vehicle, i.e., this specification can be implemented without coordinating with other vehicles. In the lane change controller, however, we specifically refer to views with different owners to restrict the possible transitions of one vehicle. For implementations, this implies that information has to be passed between vehicles. This is in line with our previous automata based specification of the lane change controller for regular sensors [1].

With these definitions, we can prove a slightly refined safety theorem. We not only require that *safe* e is satisfied for all vehicles e, but that *safe* e is satisfied, *after we switch to the view owned by* e. This addition is sufficient, since for each e, the views it owns contain the maximum amount of information about e.

$$\textbf{theorem}\ safety : \models (\forall e.@e\ (safe\ e)) \wedge DC' \wedge LC' \rightarrow \mathbf{G}\ (\forall e.@e\ (safe\ e))$$

The proof of this theorem is then similar to the safety proof in Sect. 3, insofar that we start by induction on the length of transition sequences, and then proceed

by contradiction. We need to distinguish several more cases, but these cases themselves are proven analogously to the original proof.

5 Conclusion

We presented a semantical embedding of the spatio-temporal logic HMLSL, specifically designed to reason about motorway traffic, into Isabelle/HOL, and thus implemented the first computer-based assistance for reasoning with HMLSL. Isabelle/HOL as a framework enabled us to use its highly sophisticated automatic proof methods. Within this embedding, we proved the absence of collisions, if the controllers of all vehicles adhere to a certain set of constraints. The constraints needed for proving safety differ, if we reduce the capability of the sensors deployed in the vehicles. Parameterising our embedding with the types of sensors allowed us to prove general theorems and rules of MLSL, which could subsequently be used by all instantiations of HMLSL.

Of course, our level of abstraction is high, since we focus on the spatial aspects of the motorway. However, our safety theorems show which capabilities vehicles have to possess, to ensure safety on a motorway. E.g., for perfect information, the controllers only have to adhere to the constraints implied by the reservations. For regular information, the vehicles need more capabilities, in particular, the ability to pass information between them. Olderog et al. examined ways to link a formal model very similar to ours (i.e., based on similar notions of traffic snapshots and views) with concrete controller implementations [19]. They specify high-level controllers, where MLSL formulas may be used as guards and invariants. To link our presentation to their work, the semantics of these controllers, as well as the *linking predicates* that specify the connection between the dynamics and the high-level controllers would have to be formalised within Isabelle/HOL. Then, proving safety amounts to proving that the controllers satisfy our requirements. Since Olderog et al. assumed perfect information for the controllers, their general approach has to be refined to take less idealistic information into account.

Our current proofs show safety of motorway traffic, which can be achieved, if the vehicles do not drive at all. Hence, proving liveness properties would be an interesting extension of our current approach. Both sensor definitions we presented are very idealistic. For example, we did not take imprecision or probabilistic failures into account. However, such properties could be encoded into more complex sensor functions, e.g., by using probability measures in Isabelle/HOL as defined by Hölzl [20]. Since our definition of HMLSL is parametric in the sensor definition, the main properties of the logic can be reused, and only the new implications of the sensor definition have to be proven.

Furthermore, the embedding is designed for motorway traffic, i.e., vehicles driving into one direction on a multi-lane highway. A natural extension would be to take oncoming traffic into account and could be done along lines of previous work [21]. In this case, the model would probably just need slight adjustments, e.g., to distinguish vehicles driving in different directions. Extensions to model urban traffic could be defined following, e.g., Hilscher and Schwammberger [22]

or Xu and Li [23]. However, the models in both of these approaches differ strongly from the model for motorway traffic.

References

1. Hilscher, M., Linker, S., Olderog, E., Ravn, A.: An abstract model for proving safety of multi-lane traffic manoeuvres. In: ICFEM. Volume 6991 of LNCS., Springer (2011) 404–419
2. Linker, S., Hilscher, M.: Proof theory of a multi-lane spatial logic. LMCS **11** (2015)
3. Linker, S.: Proofs for Traffic Safety: Combining Diagrams and Logic. PhD thesis, University of Oldenburg (2015) `http://oops.uni-oldenburg.de/2337/`.
4. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL — A Proof Assistant for Higher-Order Logic. Volume 2283 of LNCS. Springer (2002)
5. Platzer, A., Quesel, J.D.: KeYmaera: A hybrid theorem prover for hybrid systems. In: IJCAR. Volume 5195 of LNAI., Springer (2008) 171–178
6. Loos, S.M., Platzer, A., Nistor, L.: Adaptive cruise control: Hybrid, distributed, and now formally verified. In: FM. Volume 6664 of LNCS., Springer (2011) 42–56
7. Platzer, A.: The complete proof theory of hybrid systems. In: LICS, IEEE (2012) 541–550
8. Rizaldi, A., Althoff, M.: Formalising traffic rules for accountability of autonomous vehicles. In: ITSC, IEEE (2015) 1658–1665
9. Kamali, M., Dennis, L.A., McAree, O., Fisher, M., Veres, S.M.: Formal verification of autonomous vehicle platooning. arXiv preprint arXiv:1602.01718 (2016)
10. Alur, R., Dill, D.L.: A theory of timed automata. TCS **126** (1994) 183 – 235
11. Dennis, L.A., Fisher, M., Webster, M.P., Bordini, R.H.: Model checking agent programming languages. ASE **19** (2012) 5–63
12. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a Nutshell. STTT **1** (1997) 134–152
13. Campbell, J., Tuncali, C.E., Liu, P., Pavlic, T.P., Ozguner, U., Fainekos, G.: Modeling concurrency and reconfiguration in vehicular systems: A π-calculus approach. In: CASE, IEEE (2016) 523–530
14. Alur, R.: Principles of Cyber-Physical Systems. MIT Press (2015)
15. Braüner, T.: Hybrid logic and its proof-theory. Springer (2010)
16. Moszkowski, B.C.: A temporal logic for multilevel reasoning about hardware. Computer **18** (1985) 10–19
17. Benzmüller, C., Paulson, L.: Quantified multimodal logics in simple type theory. Logica Universalis **7** (2013) 7–20
18. Bochmann, G.V., Hilscher, M., Linker, S., Olderog, E.R.: Synthesizing and verifying controllers for multi-lane traffic maneuvers. FAC (2017) 1–18
19. Olderog, E.R., Ravn, A.P., Wisniewski, R.: Linking discrete and continuous models, applied to traffic manoeuvrers. In: Provably Correct Systems, Springer (2017) 95–120
20. Hölzl, J.: Markov processes in Isabelle/HOL. In: CPP 2017, ACM (2017) 100–111
21. Hilscher, M., Linker, S., Olderog, E.R.: Proving safety of traffic manoeuvres on country roads. In: Theories of Programming and Formal Methods. Volume 8051 of LNCS., Springer (2013) 196–212
22. Hilscher, M., Schwammberger, M.: An abstract model for proving safety of autonomous urban traffic. In: ICTAC. Volume 9965 of LNCS., Springer (2016) 274–292
23. Xu, B., Li, Q.: A Spatial Logic for Modeling and Verification of Collision-Free Control of Vehicles. In: ICECCS, IEEE (2016) 33–42