An Abstract Model for Proving Safety of Multi-Lane Traffic Manoeuvres*

Martin Hilscher[1], Sven Linker[1], Ernst-Rüdiger Olderog[1], Anders P. Ravn[2]

[1] Department of Computing Science, University of Oldenburg, Germany
{martin.hilscher, sven.linker, olderog}@informatik.uni-oldenburg.de
[2] Department of Computer Science, Aalborg University, Denmark
apr@cs.aau.dk

Abstract. We present an approach to prove safety (collision freedom) of multi-lane motorway traffic with lane-change manoeuvres. This is ultimately a hybrid verification problem due to the continuous dynamics of the cars. We abstract from the dynamics by introducing a new spatial interval logic based on the view of each car. To guarantee safety, we present two variants of a lane-change controller, one with perfect knowledge of the safety envelopes of neighbouring cars and one which takes only the size of the neighbouring cars into account. Based on these controllers we provide a local safety proof for unboundedly many cars by showing that at any moment the reserved space of each car is disjoint from the reserved space of any other car.

Keywords. Multi-lane motorway traffic, lane-change manoeuvre, collision freedom, abstract modelling, spatial interval logic, timed automata

1 Introduction

To increase the safety of road traffic many individual driving assistant systems based on suitable sensors have been developed for cars. The next step is to utilize car to car communication to combine such individual system to build up more advanced assistance functionalities. In this paper we study one such functionality, lane-change assistance for cars driving on a multi-lane motorway. The challenge is to develop lane-change controllers based on suitable sensor and communication facilities such that the *safety* (*collision freedom*) of multi-lane motorway traffic can be demonstrated if all cars are equipped with such a controller. This is ultimately a problem of hybrid system verification, where the car dynamics, the car controllers, and suitable assumptions together should imply safety.

In the California PATH (Partners for Advanced Transit and Highways) project automated highway systems for car platoons including lane change have been designed. Lygeros et al. [1] sketch a safety proof taking car dynamics into account, but admitting *safe collisions*, i.e., collisions at a low speed. Not all scenarios of multi-lane traffic are covered in the analysis. Jula et al. [2] provide calculations

* This research was partially supported by the German Research Council (DFG) in the Transregional Collaborative Research Center SFB/TR 14 AVACS.

of safe longitudinal distances between cars based on car dynamics. Werling et al. [3] study car traffic in urban scenarios and an abstract representation of several car manoeuvres. In their analysis cars are assumed to drive with constant speed. To simplify safety proofs controller patterns are exploited in Damm et al. [4], where a proof rule for collision freedom of two traffic agents based on criticality functions is proposed. This proof rule has for instance been applied to verify a distance controller. However, it is not clear how to extend this approach to deal with arbitrarily many cars on a motorway. Our paper is inspired by approaches to controller design for hybrid systems that *separate* the dynamics from the control layer. Raisch et al. [5,6] introduce abstraction and refinement to support a hierarchical design of hybrid control systems. Van Schuppen et al. [7] introduce synthesis of control laws for piecewise-affine hybrid systems based on simplices.

Our key idea for coping with the safety of many cars on a motorway is to show that different cars occupy and reserve disjoint spaces. To this end, we introduce an abstract model of multi-lane motorway traffic based on spatial properties of local views of cars. The properties are expressed in a new dedicated *Multi-Lane Spatial Logic* (MLSL) inspired by Moskowski's interval temporal logic [8], Zhou, Hoare and Ravn's Duration Calculus [9], and Schäfer's Shape Calculus [10]. MLSL is a two-dimensional extension of interval temporal logic, where one dimension has a continuous space (the position in each lane) and the other has a discrete space (the number of the lane). In MLSL we can for instance express that a car E has reserved a certain space on its lane. However, that the size of this reservation covers the braking distance of E is not part of the spatial logic. This would come into the picture only when refining the spatial properties to the car dynamics, which is not part of this paper. By using MLSL, we separate the purely spatial reasoning from the car dynamics.

As we shall see, spatial properties needed for the safety proof can be expressed very concisely in MLSL. We shall use formulas of MLSL as guards and state invariants of abstract lane-change controllers. In a technical realisation of such controllers, the properties that may appear in the formulas stipulate suitable sensors of the cars, for instance distance sensors.

The contributions of our paper are follows:

- we introduce an abstract model of motorway traffic with lane-change manoeuvres and a suitable spatial interval logic MLSL (Sect. 2);
- we provide two variants of lane-change controllers, a simple one with perfect knowledge of the safety envelopes of neighbouring cars and an elaborated one which takes only the extension of the neighbouring cars into account, but requires communication with a helper car (Sect. 3);
- we conduct proofs of safety (collision freedom) for both controllers (Sect. 4).

Finally, in Sect. 5 we conclude and discuss more related and future work.

2 Abstract Model

Usually, road traffic is modelled as a dynamical system, where each vehicle has a trajectory in the plane defined by its position, its speed and its acceleration [1].

However, to conduct a proof of safety of many cars on a multi-lane motorway, this is a far too detailed description of traffic. Thus we introduce a more abstract model which is based on local views of cars as shown in Fig. 1.

We start from a global picture of multi-lane motorway traffic, where the road has an infinite extension with positions represented by the real numbers and where lanes are represented by natural numbers $0, 1, \ldots, n$. At each moment of time each car, with a unique identity denoted by letters A, B, ..., has its position *pos*, speed *spd*, and acceleration *acc*. We assume that all traffic proceeds in one direction, with increasing position values, in the pictures shown from left to right. The abstract model is introduced by allowing for each car only local views of this traffic. A view of a car E comprises a contiguous subset of lanes, and has a bounded extension. A view containing all lanes with an extension up to a given constant, the *horizon*, will be called *standard view*.

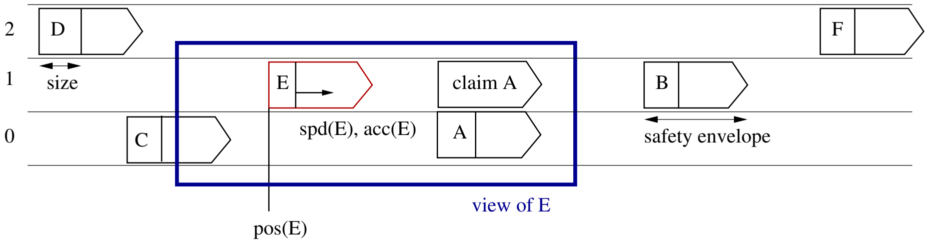


Fig. 1. View of car E comprising a bounded extension of lanes 1 and 0. Car E sees its own reservation, both the reservation and the claim of car A ahead, which is preparing for a change from lane 0 to lane 1, and part of the reservation of car C driving on a neighbouring lane behind E. It does not see the cars B, D and F because they are driving outside of its view.

What a car "knows" of its view is expressed by formulas in a dedicated multi-lane spatial logic, which extends interval temporal logic [8] to two dimensions, one with a continuous space (the position in each lane) and the other with a discrete space (the number of the lane). Such a formula consists of a finite list of lanes, where each lane is characterized by a finite sequence of segments. A segment is either occupied by a car, say E, or it is empty (*free*). For instance, in the view of car E shown in Fig. 1, the following formula ϕ holds:

$$\phi \equiv \left\langle \begin{array}{c} \mathit{free} \frown E \frown \mathit{free} \frown cl(A) \frown \mathit{free} \\ C \frown \mathit{free} \frown re(A) \frown \mathit{free} \end{array} \right\rangle$$

Here $\frown$ is the *chop operator* of interval temporal logic; it serves to separate adjacent segments in a lane. In the logic we can distinguish whether a car A has *reserved* a space in a lane ($re(A)$) or only *claimed* a space ($cl(A)$) for a planned lane change manoeuvre. We stipulate that reserved and claimed spaces have the extension of the *safety envelopes* of the cars, which include at each moment the speed dependent braking distances. The key idea of our approach is that we abstract from the exact values of these distances in our safety proof.

2.1 Traffic snapshot

We introduce a formal model $\mathcal{TS}$ of a traffic snapshot, which describes the traffic on the motorway at a given point in time. Henceforth we assume a globally unique identifier for each car and take $\mathbb{I}$ as the set of all such *car identifiers*, with typical elements $A, B, \ldots$. Furthermore, $\mathbb{L} = \{0, \ldots, N\}$, for some fixed $N \geq 1$, denotes the set of motorway lanes, with typical elements l, m, n.

Definition 1 (Traffic snapshot). *A* traffic snapshot $\mathcal{TS}$ *is a structure*

$$\mathcal{TS} = (res, clm, pos, spd, acc),$$

where res, clm, pos, spd, acc are functions

- $res : \mathbb{I} \to \mathcal{P}(\mathbb{L})$ *such that $res(C)$ is the set of lanes C reserves,*
- $clm : \mathbb{I} \to \mathcal{P}(\mathbb{L})$ *such that $clm(C)$ is the set of lanes C claims,*
- $pos : \mathbb{I} \to \mathbb{R}$ *such that $pos(C)$ is the position of car C along the lanes,*
- $spd : \mathbb{I} \to \mathbb{R}$ *such that $spd(C)$ is the current speed of the car C,*
- $acc : \mathbb{I} \to \mathbb{R}$ *such that $acc(C)$ is the current acceleration of the car C.*

We denote the set of all traffic snapshots by $\mathbb{TS}$.

Definition 2 (Transitions). *The following* transitions *describe the changes that may occur at a traffic snapshot $\mathcal{TS} = (res, clm, pos, spd, acc)$. Note that we use the overriding notation $\oplus$ of Z for function updates [11].*

$$\begin{aligned}
\mathcal{TS} \xrightarrow{t} \mathcal{TS}' \quad \Leftrightarrow \quad & \mathcal{TS}' = (res, clm, pos', spd', acc) \\
& \wedge \forall C \in \mathbb{I} \colon pos'(C) = pos(C) + spd(C) \cdot t + \tfrac{1}{2} acc(C) \cdot t^2 \\
& \wedge \forall C \in \mathbb{I} \colon spd'(C) = spd(C) + acc(C) \cdot t \qquad (1)
\end{aligned}$$

$$\begin{aligned}
\mathcal{TS} \xrightarrow{\mathsf{c}(C,n)} \mathcal{TS}' \quad \Leftrightarrow \quad & \mathcal{TS}' = (res, clm', pos, spd, acc) \\
& \wedge |clm(C)| = 0 \wedge |res(C)| = 1 \\
& \wedge \{n+1, n-1\} \cap res(C) \neq \emptyset \\
& \wedge clm' = clm \oplus \{C \mapsto \{n\}\} \qquad (2)
\end{aligned}$$

$$\begin{aligned}
\mathcal{TS} \xrightarrow{\mathsf{wd\ c}(C)} \mathcal{TS}' \quad \Leftrightarrow \quad & \mathcal{TS}' = (res, clm', pos, spd, acc) \\
& \wedge clm' = clm \oplus \{C \mapsto \emptyset\} \qquad (3)
\end{aligned}$$

$$\begin{aligned}
\mathcal{TS} \xrightarrow{\mathsf{r}(C)} \mathcal{TS}' \quad \Leftrightarrow \quad & \mathcal{TS}' = (res', clm', pos, spd, acc) \\
& \wedge clm' = clm \oplus \{C \mapsto \emptyset\} \\
& \wedge res' = res \oplus \{C \mapsto res(C) \cup clm(C)\} \qquad (4)
\end{aligned}$$

$$\begin{aligned}
\mathcal{TS} \xrightarrow{\mathsf{wd\ r}(C,n)} \mathcal{TS}' \quad \Leftrightarrow \quad & \mathcal{TS}' = (res', clm, pos, spd, acc) \\
& \wedge res' = res \oplus \{C \mapsto \{n\}\} \\
& \wedge n \in res(C) \wedge |res(C)| = 2 \qquad (5)
\end{aligned}$$

$$\begin{aligned}
\mathcal{TS} \xrightarrow{\mathsf{acc}(C,a)} \mathcal{TS}' \quad \Leftrightarrow \quad & \mathcal{TS}' = (res, clm, pos, spd, acc') \\
& \wedge acc' = acc \oplus \{C \mapsto a\} \qquad (6)
\end{aligned}$$

In (1) time can pass, which results in the cars moving along the motorway according to their respective speeds and accelerations. A car may *claim* a neighbouring lane n *iff* it currently does not already claim another lane or is in the progress of changing the lane and therefore reserves two lanes (2). Furthermore a car may *withdraw* a claim (3) or *reserve* a previously claimed lane (4) or withdraw the reservation of all but one of the lanes it is moving on (5). Finally a car may change its acceleration (6).

Example. The following trace shows a car C driving for t_1 seconds on lane 1 or 3, then claiming lane 2, driving for t_2 seconds while claiming lane 2, reserving lane 2, driving for t_{lc} seconds on both lanes (moving over) and then withdrawing all reservations but the one for lane 2.

$$\mathcal{TS}_1 \xrightarrow{t_1} \mathcal{TS}_2 \xrightarrow{\mathsf{c}(C,2)} \mathcal{TS}_3 \xrightarrow{t_2} \mathcal{TS}_4 \xrightarrow{\mathsf{r}(C)} \mathcal{TS}_5 \xrightarrow{t_{lc}} \mathcal{TS}_6 \xrightarrow{\mathsf{wd}\ \mathsf{r}(C,2)} \mathcal{TS}_7$$

2.2 View

For our safety proof we will restrict ourselves to finite parts of a traffic snapshot $\mathcal{TS}$ called views, the intuition being that the safety of manoeuvres can be shown using local information only.

Definition 3 (View). *A* view V *is defined as a structure* $V = (L, X, E)$*, where*

- $L = [l, n] \subseteq \mathbb{L}$ *is an interval of lanes that are visible in the view,*
- $X = [r, t] \subseteq \mathbb{R}$ *is the extension that is visible in the view,*
- $E \in \mathbb{I}$ *is the identifier of the car under consideration.*

A subview *of* V *is obtained by restricting the lanes and extension we observe. For this we use sub- and superscript notation:* $V^{L'} = (L', X, E)$ *and* $V_{X'} = (L, X', E)$*, where* L' *and* X' *are subintervals of* L *and* X*, respectively.*

For a car E *and a traffic snapshot* $\mathcal{TS} = (res, clm, pos, spd, acc)$ *we define the standard view of* E *as*

$$V_s(E, \mathcal{TS}) = (\mathbb{L}, [pos(E) - h, pos(E) + h], E) \ ,$$

where the horizon h *is chosen such that a car driving at maximum speed can, with lowest deceleration, come to a standstill within the horizon* h*.*

Sensor Function. Subsequently we will use a car dependent sensor function $\Omega_E : \mathbb{I} \times \mathbb{TS} \rightarrow \mathbb{R}_+$ which, given a car identifier and a traffic snapshot, provides the length of the corresponding car, as perceived by E. In Section 3 we will give safety proofs for two sensor function instantiations, one delivering the *safety envelope* of all cars (perfect knowledge) and one delivering only the actual *size* of cars. See Fig. 1 for illustration.

Abbreviations For a given view $V = (L, X, E)$ and a traffic snapshot $\mathcal{TS} = (res, clm, pos, spd, acc)$ we use the following abbreviations:

$$\begin{aligned} I_V = \{C \mid C \in \mathbb{I} \wedge (\exists l \in L : l \in res(C) \vee l \in clm(C)) \\ \wedge [pos(C), pos(C) + \Omega_E(C, \mathcal{TS})] \cap X \neq \emptyset\} \end{aligned} \tag{7}$$

$$res_V = res \cap (I_V \times \mathcal{P}(L)) \tag{8}$$

$$clm_V = clm \cap (I_V \times \mathcal{P}(L)) \tag{9}$$

$$len_V : \begin{cases} I_V \to \mathcal{P}(X) \\ C \mapsto [pos(C), pos(C) + \Omega_E(C, \mathcal{TS})] \cap X \end{cases} \tag{10}$$

The set (7) is constructed in the following way: a car C is in I_V *iff* it occupies (intends to change to) a lane considered in this view and C's occupation of the road as perceived by E intersects with the extension considered in the view. The functions (8) and (9) are restrictions of their counterparts in $\mathcal{TS}$ to the sets of lanes and identifiers considered in this view. The function (10) gives us the part of the motorway car E perceives occupied by a car cut on the edges of the view's extension.

2.3 A Multi-Lane Spatial Logic

In this section we will define the syntax and semantics of the spatial logic used in the definition of the lane change controller. Since we are interested in the safety of manoeuvres on a motorway with multiple lanes, we call this logic *multi-lane spatial logic* (MLSL). We employ five different atoms, boolean connectors and first-order quantification. Furthermore we use two *chop* operations. The first chop is denoted by $\frown$ like for interval logics, while the second chop operation is given only by the vertical arrangement of formulae.

Their intuitions are as follows. A formula $\phi_1 \frown \phi_2$ is satisfied by a view V with the extension $[r, t]$, if V can be divided at a point s into two subviews V_1 and V_2, where V_1 has the extension $[r, s]$ and satisfies ϕ_1 and V_2 has the extension $[s, t]$ and satisfies ϕ_2, respectively. A formula $\substack{\phi_2 \\ \phi_1}$ is satisfied by V with the lanes l to n, if V can be split along a lane m into two subviews, V_1 with the lanes l to m and V_2 with the lanes $m + 1$ to n, where V_i satisfies ϕ_i for $i = 1, 2$.

The set of variables ranging over car identifiers is denoted by Var, with typical elements c, d, u and v. To refer to the car owning the current view, we use a special variable ego $\in$ Var.

Definition 4 (Syntax). *The syntax of the* multi-lane spatial logic MLSL *is given by the following formulae:*

$$\phi ::= \mathit{true} \mid u = v \mid \mathit{free} \mid re(\gamma) \mid cl(\gamma) \mid \phi_1 \wedge \phi_2 \mid \neg \phi_1 \mid \exists v \colon \phi_1 \mid \phi_1 \frown \phi_2 \mid \begin{matrix} \phi_2 \\ \phi_1 \end{matrix}$$

where γ is a variable or a car identifier, and u and v are variables. We denote the set of all MLSL formulae by Φ.

Definition 5 (Valuation and Modification). *A* valuation *is a function* $\nu\colon \mathrm{Var} \to \mathbb{I}$*. For a valuation* ν *we use the overriding notation* $\nu \oplus \{v \mapsto \alpha\}$ *to denote the* modified valuation*, where the value of* v *is modified to* α*.*

Since the semantics is defined with respect to both views and valuations, we will only consider valuations ν which are *consistent* with the current view $V = (L, X, E)$, which means that we require $\nu(\mathrm{ego}) = E$. In the following definition, observe that we require that the spatial atoms may only hold on a view with exactly one lane and an extension greater than zero. In the semantics of *free*, we abstract from cars visible only at the endpoints of the view.

Definition 6 (Semantics). *In the following, let* u *and* v *be variables and* γ *a variable or a car identifier. The* satisfaction *of formulae with respect to a traffic snapshot* $\mathcal{TS}$*, a view* $V = (L, X, E)$ *with* $L = [l, n]$ *and* $X = [r, t]$*, and a valuation* ν *consistent with* V *is defined inductively as follows:*

$$
\begin{array}{lll}
\mathcal{TS}, V, \nu \models \mathit{true} & & \mathit{for\ all}\ \mathcal{TS}, V, \nu \\
\mathcal{TS}, V, \nu \models u = v & \Leftrightarrow & \nu(u) = \nu(v) \\
\mathcal{TS}, V, \nu \models \mathit{free} & \Leftrightarrow & |L| = 1\ \mathit{and}\ |X| > 0\ \mathit{and} \\
& & \forall i \in I_V : \mathit{len}_V(i) \cap (r, t) = \emptyset \\
\mathcal{TS}, V, \nu \models \mathit{re}(\gamma) & \Leftrightarrow & |L| = 1\ \mathit{and}\ |X| > 0\ \mathit{and}\ \nu(\gamma) \in I_V\ \mathit{and} \\
& & \mathit{res}_V(\nu(\gamma)) = L\ \mathit{and}\ X = \mathit{len}_V(\nu(\gamma)) \\
\mathcal{TS}, V, \nu \models \mathit{cl}(\gamma) & \Leftrightarrow & |L| = 1\ \mathit{and}\ |X| > 0\ \mathit{and}\ \nu(\gamma) \in I_V\ \mathit{and} \\
& & \mathit{clm}_V(\nu(\gamma)) = L\ \mathit{and}\ X = \mathit{len}_V(\nu(\gamma)) \\
\mathcal{TS}, V, \nu \models \phi_1 \wedge \phi_2 & \Leftrightarrow & \mathcal{TS}, V, \nu \models \phi_1\ \mathit{and}\ \mathcal{TS}, V, \nu \models \phi_2 \\
\mathcal{TS}, V, \nu \models \neg\phi & \Leftrightarrow & \mathit{not}\ \mathcal{TS}, V, \nu \models \phi \\
\mathcal{TS}, V, \nu \models \exists v\colon \phi & \Leftrightarrow & \exists\, \alpha \in I_V : \mathcal{TS}, V, \nu \oplus \{v \mapsto \alpha\} \models \phi \\
\mathcal{TS}, V, \nu \models \phi_1 \frown \phi_2 & \Leftrightarrow & \exists\, s\colon r \leq s \leq t\ \mathit{and} \\
& & \mathcal{TS}, V_{[r,s]}, \nu \models \phi_1\ \mathit{and}\ \mathcal{TS}, V_{[s,t]}, \nu \models \phi_2 \\
\mathcal{TS}, V, \nu \models \begin{matrix}\phi_2 \\ \phi_1\end{matrix} & \Leftrightarrow & \exists\, m\colon l - 1 \leq m \leq n + 1\ \mathit{and} \\
& & \mathcal{TS}, V^{[l,m]}, \nu \models \phi_1\ \mathit{and}\ \mathcal{TS}, V^{[m+1,n]}, \nu \models \phi_2
\end{array}
$$

We write $\mathcal{TS} \models \phi$ *if* $\mathcal{TS}, V, \nu \models \phi$ *for all views* V *and consistent valuations* ν*.*

For the semantics of the vertical chop, we set the interval $[l, m] = \emptyset$ if $l > m$. A view V with an empty set of lanes may only satisfy *true* or an equality formula. We remark that both chop modalities are associative. For the definition of the controller we employ some abbreviations. In addition to the usual definitions of $\vee, \rightarrow, \leftrightarrow$ and $\forall$, we use a single variable or car identifier γ as an abbreviation for $\mathit{re}(\gamma) \vee \mathit{cl}(\gamma)$. Furthermore, we use the notation $\langle\phi\rangle$ for the two-dimensional modality *somewhere* ϕ, defined in terms of both chop operations:

$$\langle \phi \rangle \equiv \mathit{true} \frown \begin{pmatrix} \mathit{true} \\ \phi \\ \mathit{true} \end{pmatrix} \frown \mathit{true}.$$

In the following, the main application of the somewhere modality is to abstract the exact positions on the road from formulae, e.g., to identify overlaps of claims and safety envelopes. If a view V satisfies the formula $\exists c\colon\ \langle \mathit{cl}(\mathrm{ego}) \land \mathit{re}(c) \rangle$, then there is a part on some lane in V occupied by both the claim of the car under consideration and the safety envelope of some car c.

In the safety proof we exploit that somewhere distributes over disjunction:

$$\langle \phi_1 \lor \phi_2 \rangle \equiv \langle \phi_1 \rangle \lor \langle \phi_2 \rangle\,. \tag{11}$$

This equivalence is an immediate consequence of the semantics.

3 Controllers

We now present two lane-change controllers, one with perfect knowledge of the safety envelopes (covering the necessary braking distances) of neighbouring cars and one which takes only the physical size of the neighbouring cars into account.

The controllers are specified as *timed automata* [12] with clocks ranging over $\mathbb{R}$ and data variables ranging over $\mathbb{L}$ and $\mathbb{I}$. The semantics is a transition system, where a configuration $\mathcal{C}$ consists of a traffic snapshot $\mathcal{TS}$, the standard view V of a car, a valuation ν (also of clocks and data variables), and the current state q of the controller, i.e. $\mathcal{C} = (\mathcal{TS}, V, \nu, q)$. To restrict the transitions which are allowed in a lane-change manoeuvre, like the creation of new claims and the extension and shrinking of reservations, suitable MLSL formulae will appear in transition guards and state invariants. We take care that none of our controllers introduces a timelock, which would prevent time from progressing unboundedly.

Timed automata working in parallel can communicate with each other via *broadcast channels* as in UPPAAL [13]. Using a CSP-style notation [14], sending a value val over a channel p is denoted by $p!\mathit{val}$; receiving a value over p and binding it to a variable c appearing free in a guard ϕ is denoted by $p?c\colon \phi$. Formally, $\mathcal{TS}, V, \nu \models p?c\colon \phi$ *iff* $\mathcal{TS}, V, \nu \oplus \{c \mapsto \mathit{val}\} \models \phi$, where val is the value simultaneously sent via $p!\mathit{val}$ by another automaton. A message sent by a car C is broadcast to all cars within the extension of the standard view of C.

3.1 Changing Lanes with Perfect Knowledge

Let us first assume that every car can perceive the full extension of claims and reservations of all cars within its view. In other words, every car has *perfect knowledge* of the status of the road within its view. This assumption is formalised through the sensor function Ω_E, which defines the extension of the cars seen by the owner E of a view. Putting $\Omega_E(I, \mathcal{TS}) = \mathit{se}(I, \mathcal{TS})$ models that the sensors return the whole safety envelope for all cars. This implies that a car E perceives a car C as soon as C's safety envelope enters the view of E.

Intuitively, a car C on lane n, in the following called the *actor*, can claim a space on a target lane m next to n to start the manoeuvre. This does not yet imply that C actually changes the lane. It corresponds to setting the direction indicator to prepare for a lane change. The goal of the actor is to safely convert its claim into a reservation of m. If the space claimed by the actor is already occupied or claimed by another car (*potential collision check*), C removes its claim and continues driving on its current lane. Even though we assume instantaneous transitions, we allow time to pass up to a certain time bound *to* between claiming and reserving a lane. If no potential collision occurs, the actor communicates its new reservation and starts its manoeuvre. Since we abstract from the exact form of changing the lane, we just assume that the manoeuvre takes at most t_{lc} time to finish. Finally, the actor shrinks its reservation to solely m.

This intuition is formalised by the lane-change controller LCP in Fig. 2. At the initial state q_0, we assume that the car has reserved exactly one lane, which is saved in the variable n. Furthermore, we employ an auxiliary variable l to store the lane the actor wants to change to. The *collision check cc* expresses the disjointness of the actor's reservation and the reservations of all other cars:

$$cc \equiv \neg\exists c\colon c \neq \text{ego} \wedge \langle re(\text{ego}) \wedge re(c)\rangle\,,$$

the *potential collision check* $pc(c)$ for a car c expresses the overlapping of the actor's claim with (the reservation or claim of) c:

$$pc(c) \equiv c \neq \text{ego} \wedge \langle cl(\text{ego}) \wedge c\rangle\,.$$

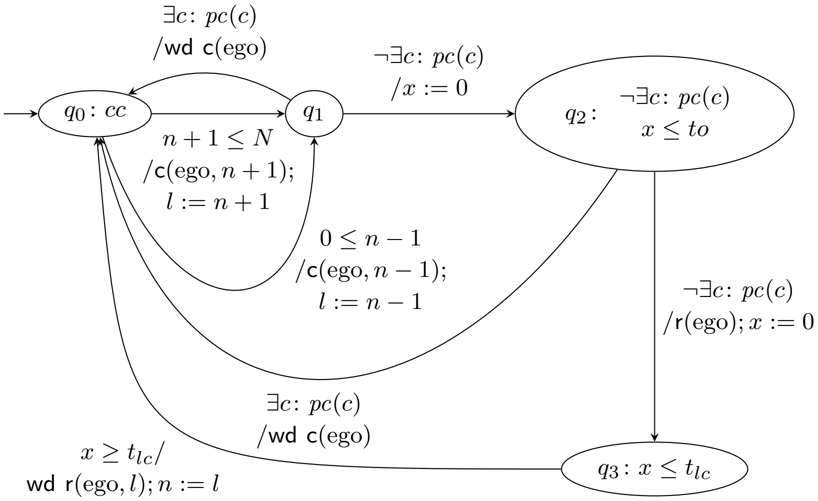

$\exists c\colon pc(c)$
$/\mathsf{wd\ c}(\text{ego})$
$\neg\exists c\colon pc(c)$
$/x := 0$
$q_0\colon cc$
q_1
$q_2\colon$
$\neg\exists c\colon pc(c)$
$x \leq to$
$n+1 \leq N$
$/\mathsf{c}(\text{ego}, n+1);$
$l := n+1$
$0 \leq n-1$
$/\mathsf{c}(\text{ego}, n-1);$
$l := n-1$
$\neg\exists c\colon pc(c)$
$/\mathsf{r}(\text{ego}); x := 0$
$\exists c\colon pc(c)$
$/\mathsf{wd\ c}(\text{ego})$
$x \geq t_{lc}/$
$\mathsf{wd\ r}(\text{ego}, l); n := l$
$q_3\colon x \leq t_{lc}$

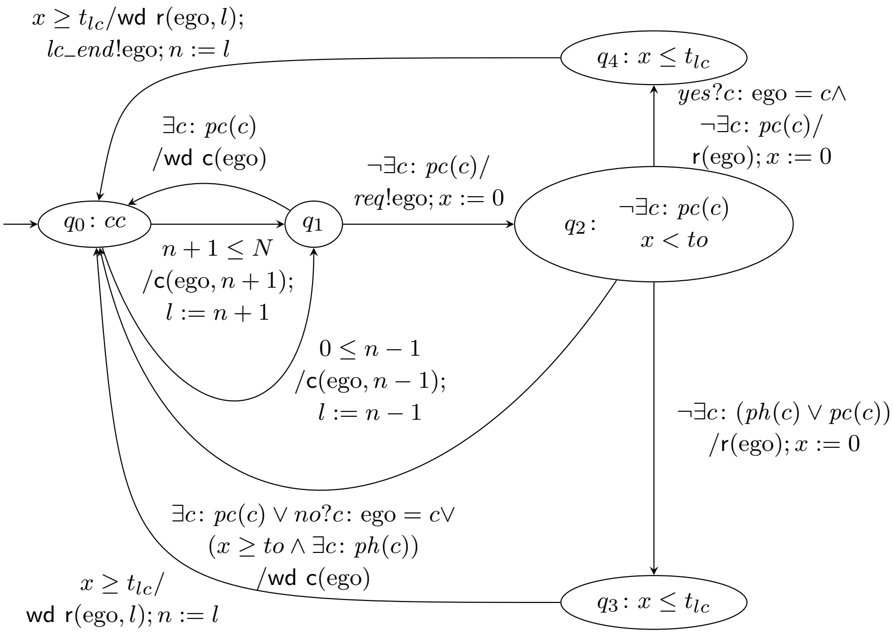
Fig. 2. Controller LCP for the Lane-Change Manoeuvre with Perfect Knowledge

3.2 A More Realistic Approach for Changing Lanes

The assumption that every car can perceive every safety envelope within its view is very strong. In this section, we define a controller which accomplishes a lane-change manoeuvre with much less information: each car knows only the size of the other cars, while it still knows its own safety envelope. Hence the sensor function for a view $V = (L, X, E)$ is defined conditionally by

$$\Omega_E(I, \mathcal{TS}) \equiv \mathsf{if}\ I = E\ \mathsf{then}\ se(I, \mathcal{TS})\ \mathsf{else}\ size(I)\ \mathsf{fi}.$$

In this setting, the potential collision check is not sufficient for the safety of the manoeuvre, since the actor cannot know whether the safety envelope of a car on the lane the actor wants to occupy overlaps with its own safety envelope. Our approach to overcome this problem is the definition of a *helper controller* HC (Fig. 4) implemented in addition to the lane-change controller LC (Fig. 3).



$x \geq t_{lc}/\mathsf{wd}\ \mathsf{r}(\text{ego}, l);$
$lc\_end!\text{ego}; n := l$
$q_4 : x \leq t_{lc}$
$yes?c\colon \text{ego} = c \wedge$
$\neg\exists c\colon pc(c)/$
$\mathsf{r}(\text{ego}); x := 0$
$\exists c\colon pc(c)$
$/\mathsf{wd}\ \mathsf{c}(\text{ego})$
$\neg\exists c\colon pc(c)/$
$req!\text{ego}; x := 0$
$q_0 : cc$
q_1
$q_2 :$
$\neg\exists c\colon pc(c)$
$x < to$
$n + 1 \leq N$
$/\mathsf{c}(\text{ego}, n + 1);$
$l := n + 1$
$0 \leq n - 1$
$/\mathsf{c}(\text{ego}, n - 1);$
$l := n - 1$
$\neg\exists c\colon (ph(c) \vee pc(c))$
$/\mathsf{r}(\text{ego}); x := 0$
$\exists c\colon pc(c) \vee no?c\colon \text{ego} = c \vee$
$(x \geq to \wedge \exists c\colon ph(c))$
$/\mathsf{wd}\ \mathsf{c}(\text{ego})$
$x \geq t_{lc}/$
$\mathsf{wd}\ \mathsf{r}(\text{ego}, l); n := l$
$q_3 : x \leq t_{lc}$

Fig. 3. Controller LC for the Lane-Change Manoeuvre with a Helper Car

The idea of the lane-change manoeuvre with the help of these controllers is similar to the previously described manoeuvre. The actor sets a claim and checks whether this claim overlaps with already existing claims and reservations. However, since the actor can perceive via $re(c)$ only the physical size of other cars c and not the whole of their safety envelopes, it cannot know whether its claim overlaps with a car driving behind the actor on the target lane. Hence the

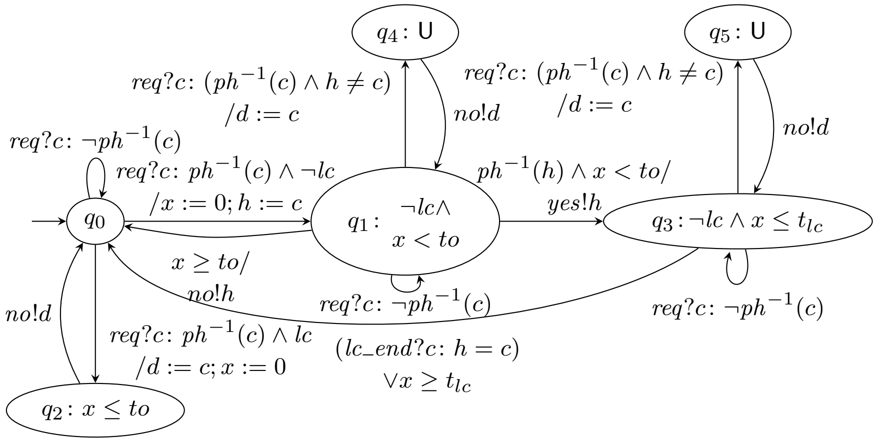

q_4 : U
q_5 : U
$req?c$: $(ph^{-1}(c) \wedge h \neq c)$
$/d := c$
$no!d$
$req?c$: $(ph^{-1}(c) \wedge h \neq c)$
$/d := c$
$no!d$
$req?c$: $\neg ph^{-1}(c)$
$req?c$: $ph^{-1}(c) \wedge \neg lc$
$/x := 0; h := c$
$ph^{-1}(h) \wedge x < to/$
$yes!h$
q_0
q_1 : $\neg lc \wedge$ $x < to$
q_3 : $\neg lc \wedge x \leq t_{lc}$
$x \geq to/$
$no!h$
$req?c$: $\neg ph^{-1}(c)$
$req?c$: $\neg ph^{-1}(c)$
$no!d$
$req?c$: $ph^{-1}(c) \wedge lc$
$/d := c; x := 0$
$(lc\_end?c$: $h = c)$
$\vee x \geq t_{lc}$
q_2 : $x \leq to$

Fig. 4. Controller HC for Helper Car

actor broadcasts a request *req* to find a potential helper. Such a *helper car* has to fulfill three conditions. It has to be on the target lane m, it has to be behind the actor, and it must not already be involved in a lane-change manoeuvre.

The formula to identify such a car from the viewpoint of the actor is called *potential helper check*:

$$ph(c) \equiv \langle re(c) \frown free \frown cl(\text{ego}) \rangle .$$

If a such helper car is approached by a broadcast request *req* from the actor, its controller HC checks for disjointness of its own reservation and the actor's claim, using the *inverse potential helper check* defined by

$$ph^{-1}(c) \equiv \langle re(\text{ego}) \frown free \frown cl(c) \rangle ,$$

and that it is not performing a lane-change manoeuvre, expressed by the formula

$$lc \equiv \left\langle \begin{matrix} \text{ego} \\ \text{ego} \end{matrix} \right\rangle .$$

If these two conditions are satisfied, it responds with the acknowledgment *yes*. Afterwards, it ensures that no other car may enter the lane in between the helper and the actor. This is done in the urgent states [13] q_4 and q_5 of the controller HC. Then, the actor may safely change the lane by extending its reserved space to lane m and remove its claim. Otherwise, if no helper is available, the actor waits for a certain time *to* without getting any response. Afterwards, it has to check, whether a car entered its view on lane m, before possibly extending its reservation to lane m. If m is free within the actor's horizon, the reservation gets extended, otherwise the actor removes its claim and returns to the initial state, since it cannot guarantee the disjointness of its claim and the reservation of the new car. After successfully changing the lane, the actor removes its reservation of lane n and drives solely on lane m.

4 Safety Proof

The desired safety property is that at any moment the spaces reserved by different cars are disjoint. To express this property we consider the formula

$$\mathit{Safe} \equiv \forall c, d : c \neq d \Rightarrow \neg \langle re(c) \wedge re(d) \rangle ,$$

which states that in each lane any two different cars have disjoint reserved spaces. The quantification over lanes arises implicitly by the negation of the somewhere modality in *Safe*. We call a traffic snapshot $\mathcal{TS}$ *safe* if $\mathcal{TS} \models \mathit{Safe}$ holds. The safety property depends on the following three assumptions.

Assumption A1. There is an *initial safe* traffic snapshot $\mathcal{TS}_0$.

Assumption A2. Every car C is equipped with a *distance controller* that keeps the safety property invariant under time and acceleration transitions, i.e., for every transition $\mathcal{TS} \xrightarrow{t} \mathcal{TS}'$ and $\mathcal{TS} \xrightarrow{\mathsf{acc}(C,a)} \mathcal{TS}'$ if $\mathcal{TS}$ is safe also $\mathcal{TS}'$ is safe.

Informally, this means that the distance controller admits a positive acceleration of C only if the space ahead permits this. Also, if the car ahead is slowing down, the distance controller has to initiate braking (with negative acceleration) of C to reduce the extension of its reservation (the safety envelope).

Assumption A3. Every car is equipped with a controller LCP as in Fig. 2.

Then the safety property is formalised by the following theorem.

Theorem 1 (Safety of LCP). *Suppose that the assumptions* A1–3 *hold. Then every traffic snapshot* $\mathcal{TS}$ *that is reachable from* $\mathcal{TS}_0$ *by time and acceleration transitions and transitions allowed by the controller LCP in Fig. 2 is safe.*

Proof. It suffices to prove safety from the perspective of each car, i.e., that there is no other car with intersecting reserved space. Formally, we fix an arbitrary car E and show that for all traffic snapshots $\mathcal{TS}$ reachable from $\mathcal{TS}_0$, all views V of E, and all valuations with $\nu(\text{ego}) = E$:

$$\mathcal{TS}, V, \nu \models \mathit{Safe}', \text{ where } \mathit{Safe}' \equiv \neg \exists\, c \neq \text{ego} \wedge \langle re(\text{ego}) \wedge re(c) \rangle . \tag{12}$$

We proceed by induction on the number k of transitions needed to reach $\mathcal{TS}$ from $\mathcal{TS}_0$.

Induction basis: $k = 0$. Then $\mathcal{TS} = \mathcal{TS}_0$ and (12) holds by A1.

Induction step: $k \to k + 1$. Consider some $\mathcal{TS}_1$ that is reachable from $\mathcal{TS}_0$ by k transitions and thus satisfy (12) by induction hypothesis. Let $\mathcal{TS}$ result from $\mathcal{TS}_1$ by one further transition, which we now examine.

For a transition $\mathcal{TS}_1 \xrightarrow{t} \mathcal{TS}$ or $\mathcal{TS}_1 \xrightarrow{\mathsf{acc}(C,a)} \mathcal{TS}$ of any car C property (12) holds for $\mathcal{TS}$ by A2. Of all other transitions allowed by the LCP controller of E, only a reservation transition $\mathcal{TS}_1 \xrightarrow{\mathsf{r}(E)} \mathcal{TS}$ could possibly violate property (12). In the LCP controller of E shown in Fig. 2 the only reservation transition starts in state q_2. This state satisfies the invariant

$$\neg \exists\, c : c \neq \text{ego} \wedge \langle cl(\text{ego}) \wedge c \rangle ,$$

which implies $\neg\exists\, c : c \neq \text{ego} \wedge \langle cl(\text{ego}) \wedge re(c)\rangle$. By taking the induction hypothesis (12) for $\mathcal{TS}_1$ into account, we thus have

$$\begin{aligned}\mathcal{TS}_1, V, \nu \models \quad & \neg\exists\, c : c \neq \text{ego} \wedge \langle cl(\text{ego}) \wedge re(c)\rangle \\ & \wedge \neg\exists\, c : c \neq \text{ego} \wedge \langle re(\text{ego}) \wedge re(c)\rangle \,.\end{aligned}$$

We transform this formula:

$$\begin{aligned}& (\neg\exists\, c : c \neq \text{ego} \wedge \langle cl(\text{ego}) \wedge re(c)\rangle) \wedge (\neg\exists\, c : c \neq \text{ego} \wedge \langle re(\text{ego}) \wedge re(c)\rangle) \\ \leftrightarrow\ & \neg\exists\, c : (c \neq \text{ego} \wedge \langle cl(\text{ego}) \wedge re(c)\rangle) \vee (c \neq \text{ego} \wedge \langle re(\text{ego}) \wedge re(c)\rangle) \\ \leftrightarrow\ & \neg\exists\, c : c \neq \text{ego} \wedge (\langle cl(\text{ego}) \wedge re(c)\rangle \vee \langle re(\text{ego}) \wedge re(c)\rangle) \\ \leftrightarrow\ & \{\text{somewhere distributes over disjunction: see (11)}\} \\ & \neg\exists\, c : c \neq \text{ego} \wedge \langle (cl(\text{ego}) \wedge re(c)) \vee (re(\text{ego}) \wedge re(c))\rangle \\ \leftrightarrow\ & \neg\exists\, c : c \neq \text{ego} \wedge \langle (cl(\text{ego}) \vee re(\text{ego})) \wedge re(c)\rangle \,.\end{aligned}$$

Applying the Reservation Lemma 1 to the latter formula yields

$$\mathcal{TS}, V, \nu \models \neg\exists\, c : c \neq \text{ego} \wedge \langle re(\text{ego}) \wedge re(c)\rangle \,,$$

which shows that (12) holds for $\mathcal{TS}$. □

We now connect formulae about reservations with the fact, that a newly created reservation occupies the same space as a previous claim (A proof is contained in the long version of this paper [15]).

Lemma 1 (Reservation). *Consider a reservation transition $\mathcal{TS} \xrightarrow{\mathrm{r}(C)} \mathcal{TS}'$ and an MLSL formula ϕ' not containing $cl(\gamma)$ as a subformula. Let ϕ result from ϕ' by replacing every occurrence of $re(\gamma)$ by $re(\gamma) \vee cl(\gamma)$. Then for all views $V = (L, X, E)$ with $C \in I_V$ and valuations ν with $\nu(\gamma) = C$ the following holds:*

$$\mathcal{TS}, V, \nu \models \phi \ \text{ if and only if } \ \mathcal{TS}', V, \nu \models \phi'.$$

4.1 Safety Proof for Changing Lanes with Help

As for the controller LCP, we want to prove that the property *Safe* is an invariant of the allowed transitions, but now the with assumption A3 modified as follows:

Assumption A3. Every car is equipped with the controllers LC as in Fig. 3 and HC of Fig. 4 running in parallel.

Since this scenario incorporates communication between two cars, the helper and the actor, we have to assume that a car changing a lane can perceive all cars whose safety envelopes reach up to its position.

Assumption A4. The horizon h of the standard view (Def. 3) is at least the length of the safety envelope of the fastest car with the smallest braking force.

Theorem 2 (Safety of LC and HC). *Suppose that the assumptions* A1–4 *hold. Then every traffic snapshot $\mathcal{TS}$ that is reachable from $\mathcal{TS}_0$ by time and acceleration transitions and transitions allowed by the controller LC in Fig. 3 and the helper controller HC in Fig. 4 is safe.*

Proof. We refine the proof of Theorem 1. Fix an arbitrary car E and show that for all traffic snapshots $\mathcal{TS}$ reachable from $\mathcal{TS}_0$, all helper cars H, standard views V of E and V_H of H, and valuations ν and ν' consistent with the respective views: $\mathcal{TS}, V, \nu \models \mathit{Safe}'$ and $\mathcal{TS}, V_H, \nu' \models \mathit{Safe}'$. Again, we proceed by induction on the number k of transitions needed to reach $\mathcal{TS}$ from $\mathcal{TS}_0$.

Induction basis: $k = 0$. Then $\mathcal{TS} = \mathcal{TS}_0$ and (12) holds by A1.

Induction step: $k \rightarrow k+1$. Consider some $\mathcal{TS}_1$ that is reachable from $\mathcal{TS}_0$ by k transitions and thus satisfy (12) by induction hypothesis. Let $\mathcal{TS}$ result from $\mathcal{TS}_1$ by one further transition, which we now examine. As in the proof of Theorem 1, the only possibly dangerous transition is a reservation transition $\mathcal{TS}_1 \xrightarrow{\mathsf{r}(E)} \mathcal{TS}$. In the controller LC of car E shown in Fig. 3 there are two such transitions, both starting in state q_2. Observe that in q_2, whenever there is a potential collision the manoeuvre is aborted. Hence, if a car D creates a new claim overlapping with E's claim, D or E withdraws its claim. Now, consider the transition from q_2 to q_3. By A4 and since the safety envelope starts at the position of its car, we may proceed as in the proof of Theorem 1.

Next, consider the transition from q_2 to q_4. We have to show that E's claim does not overlap with the reservation of the helper H. Let V_H be the standard view of H and ν' be a valuation consistent with V_H. Since the controller HC sends to E the message $\mathit{yes}!E$, it has taken the transition with the guard $\mathit{ph}^{-1}(E)$, exiting state q_1 of HC, so $\mathcal{TS}_1, V_H, \nu' \models \langle \mathit{re}(\text{ego}) \frown \mathit{free} \frown \mathit{cl}(E) \rangle$.

Since the state q_1 of HC has the invariant lc, the subformula *free* is satisfied by a subview with an extension greater than zero, and claims of E cannot overlap with existing reservations of E, this implies

$$\mathcal{TS}_1, V_H, \nu' \models \neg \langle \mathit{re}(\text{ego}) \wedge (\mathit{cl}(E) \vee \mathit{re}(E)) \rangle \, .$$

Since $\nu'(E) = E$, we can apply the Reservation Lemma 1, which yields

$$\mathcal{TS}, V_H, \nu' \models \neg \langle \mathit{re}(\text{ego}) \wedge \mathit{re}(E) \rangle \, . \qquad \square$$

5 Conclusion

The novelty in our paper is the identification of a level of abstraction that enables a purely spatial reasoning on safety. We proved safety for arbitrarily many cars on the motorway locally, by considering at most two cars at a time.

More on related work. Manoeuvres of cars have been extensively studied in the California PATH (Partners for Advanced Transit and Highways) project [16], which aimed at an Automated Highway System (AHS) to increase safety and throughput on highways. The project introduced the concept of a *platoon*, a tightly spaced convoy of cars driving on a motorway at a relatively high speed.

In Hsu et al. [17] the architecture of the AHS system is outlined, and at the platoon layer three manoeuvres are investigated: merge and split of platoons as well as lane change of free traffic agents, i.e., single cars. For these manoeuvres protocols are modelled as communicating finite state machines and tested within the automata-based tool COSPAN by R. Kurshan. The protocol for lane change does not take all possible traffic scenarios on neighbouring lanes into account. For example, the scenario where cars are driving on both the target lane and the lane next to it is not considered.

In Lygeros et al. [1] the analysis of [17] is refined by taking the hybrid controllers as the model. Sufficient conditions on the car dynamics are established for showing safety of the AHS system at the coordination layer (for communication and cooperation between cars) and the regulation layer (for hybrid controllers performing the traffic manoeuvres). The lane change manoeuvre is explicitly investigated in a multi-lane safety theorem. However, its proof, based on an induction argument on the number of cars, is only outlined. Moreover, the possible scenarios of lane change in dense traffic are only partially covered. The scenario where two cars wish to change to a common target lane is not taken into account.

The safety problem has also been studied for railway networks, which are simpler to handle because the movements of trains are more constrained than those of cars. Haxthausen and Peleska [18] give manual safety proof for trains driving in an arbitrary railway network. Faber et al. [19] provide an automatic verification of safety properties in railway networks.

Future work. On the application side we want to pursue an extension of the scope of our work. For example, we intend to study the scenarios of urban traffic as in [3]. Also, we would like to study variations of the assumptions made in our safety proofs. On the foundational side we would like to investigate the connection of MLSL with more traditional spatial logics based on topological models [20] and its meta properties like decidability. Here proof ideas from [21] might be helpful. This leads to question of automatic verification of the safety properties. Here the approach of [19] could be considered.

The semantics of MLSL may be extended to include a length measurement. Let ϕ^θ denote that ϕ holds for a length θ, where θ is a first-order term denoting a real value. The semantics on p. 7 is extended by

$$\mathcal{TS}, V, \nu \models \phi^\theta \qquad \Leftrightarrow \quad \mathcal{TS}, V, \nu \models \phi \text{ and } |X| = \nu(\theta).$$

The initial definition of MLSL semantics contained this case, but to our own surprise, we did not make use of this measurement in the controllers and the safety proofs, respectively. This is due to the fact that we reason at a very abstract level, and that differences in lengths are only taken care of in the assumptions.

To link our work to hybrid systems, a refinement of the spatial reasoning in this paper to the car dynamics is of interest. There we could benefit from the approaches in [5,6,7,4] and expect that length measurements are needed.

References

1. Lygeros, J., Godbole, D.N., Sastry, S.S.: Verified hybrid controllers for automated vehicles. IEEE Transactions on Automatic Control **43** (1998) 522–539

2. Jula, H., Kosmatopoulos, E.B., Ioannou, P.A.: Collision avoidance analysis for lane changing and merging. Technical Report UCB-ITS-PRR-99-13, California Partners for Advanced Transit and Highways (PATH), Univ. of California at Berkeley (1999)
3. Werling, M., Gindele, T., Jagszent, D., Gröll, L.: A robust algorithm for handling traffic in urban scenarios. In: Proc. IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands (2008) 168–173
4. Damm, W., Hungar, H., Olderog, E.R.: Verification of cooperating traffic agents. International Journal of Control **79** (2006) 395–421
5. Moor, T., Raisch, J., O'Young, S.: Discrete supervisory control of hybrid systems based on l-complete approximations. Discrete Event Dynamic Systems **12** (2002) 83–107
6. Moor, T., Raisch, J., Davoren, J.: Admissiblity criteria for a hierarchical design of hybrid systems. In: Proc. IFAD Conf. on Analysis and Design of Hybrid Systems, St. Malo, France (2003) 389–394
7. Habets, L.C.G.J.M., Collins, P., van Schuppen, J.: Reachability and control synthesis for piecewise-affine hybrid systems on simplices. IEEE Transactions on Automatic Control **51** (2006) 938–948
8. Moszkowski, B.: A temporal logic for multilevel reasoning about hardware. Computer **18** (1985) 10–19
9. Zhou, C., Hoare, C., Ravn, A.: A calculus of durations. Information Processing Letters **40** (1991) 269–276
10. Schäfer, A.: A calculus for shapes in time and space. In Liu, Z., Araki, K., eds.: ICTAC 2004. Volume 3407 of LNCS., Springer (2005) 463–478
11. Woodcock, J., Davies, J.: Using Z – Specification, Refinement, and Proof. Prentice Hall (1996)
12. Alur, R., Dill, D.L.: A theory of timed automata. TCS **126** (1994) 183 – 235
13. Behrmann, G., David, A., Larsen, K.G.: A tutorial on UPPAAL. In Bernardo, M., Corradini, F., eds.: Formal Methods for the Design of Real-Time Systems: 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM-RT 2004, Springer–Verlag (2004) 200–236
14. Hoare, C.A.R.: Communicating sequential processes. CACM **21** (1978) 666–677
15. Hilscher, M., Linker, S., Olderog, E.R., Ravn, A.P.: An abstract model for proving safety of multi-lane traffic maenoeuvres. Report 79, SFB/TR 14 AVACS (2011) ISSN: 1860-9821, avacs.org.
16. Varaija, P.: Smart cars on smart roads: problems of control. IEEE Transactions on Automatic Control **AC-38** (1993) 195–207
17. Hsu, A., Eskafi, F., Sachs, S., Varaija, P.: Protocol design for an automated highway system. Discrete Event Dynamic Systems **2** (1994) 183–206
18. Haxthausen, A.E., Peleska, J.: Formal development and verification of a distributed railway control system. IEEE Trans. on Software Engineering **26** (2000) 687–701
19. Faber, J., Ihlemann, C., Jacobs, S., Sofronie-Stokkermans, V.: Automatic verification of parametric specifications with complex topologies. In Méry, D., Merz, S., eds.: Integrated Formal Methods. Volume 6396 of LNCS., Springer (2010) 152–167
20. van Benthem, J., Bezhanishvili, G.: Modal logics of space. In Aiello, M., Pratt-Hartmann, I., Benthem, J., eds.: Handbook of Spatial Logics. Springer Netherlands (2007) 217–298
21. Schäfer, A.: Axiomatisation and decidability of multi-dimensional duration calculus. Information and Computation **205** (2007) 25–64