Monitoring of Traffic Manoeuvres with Imprecise Information*

Heinrich Ody
Department of Computing Science
University of Oldenburg
Oldenburg, Germany
heinrich.ody@uni-oldenburg.de

In monitoring, we algorithmically check if a single behavior satisfies a property. Here, we consider monitoring for Multi-Lane Spatial Logic (MLSL). The behavior is given as a finite transition sequence of MLSL and the property is that a spatial MLSL formula should hold at every point in time within the sequence. In our procedure we transform the transition sequence and the formula to the first-order theory of real-closed fields, which is decidable, such that the resulting formula is valid iff the MLSL formula holds throughout the transition sequence. We then assume that temporal data may have an error of up to ε, and that spatial data may have an error of up to δ. We extend our procedure to check if the MLSL formula ε-δ-robustly holds throughout the transition sequence.

Keywords. Similarity of timed words, monitoring, autonomous cars, spatio-temporal logic, robustness

1 Introduction

Multi-Lane Spatial Logic (MLSL) comprises an abstract model of a motorway and a spatial logic to reason about traffic configurations [11]. MLSL can be used to, e.g. analyse controllers of (semi) automated driving systems.

In offline monitoring we are given a recorded behavior π, a specification ψ and we want to check if π satisfies ψ, denoted as $\pi \models \psi$. In this work we perform monitoring for MLSL. While MLSL has been extended with CTL-like branching time temporal modalities [15], they are not suitable for monitoring. We formalise what it means for an MLSL formula ϕ to hold *globally* in linear time, where we denote 'globally ϕ' as $\Box\phi$. This means that here π is a transition sequence and we instantiate ψ with $\Box\phi$, where ϕ is an arbitrary MLSL formula.

We define a procedure to check if an MLSL formula holds globally in an MLSL transition sequence. For this we adapt a procedure to check satisfiability of a restricted form of MLSL formulas [9]. In this extension we transform the MLSL formula that should hold globally, and the transition sequence to the first-order theory of real closed fields, which is decidable [21] (there called elementary algebra), such that the transformed formula is valid iff $\pi \models \Box\phi$ holds.

However, it is idealistic to assume that the data we are working with is exact. Here, we consider errors in positional data (spatial imprecision) and imprecisions of when reservations and claims are set and withdrawn (temporal imprecision). For temporal robustness other approaches use that they have the satisfaction of temporal atoms as a signal over time. Here, our temporal formula is $\Box\phi$ and ϕ is our temporal atom. We do not have the truth value of ϕ as a signal over time. For this reason we decided

*Work of the author is supported by the Deutsche Forschungsgemeinschaft (DFG) within the Research Training Group DFG GRK 1765 SCARE.

L. Bulwahn, M. Kamali, S. Linker (Eds.): First Workshop on Formal Verification of Autonomous Vehicles (FVAV 2017).
EPTCS 257, 2017, pp. 43–58, doi:10.4204/EPTCS.257.6

to base the temporal aspect of MLSL on *timed words* [3], from which we then derive MLSL transition sequences. Then we define temporal robustness by deviating the time stamps in a timed word. We combine this temporal robustness with our previous work on spatial robustness [17] and define spatio-temporal similarity with a metric. We then define what it means that an MLSL formula globally holds, even if the transition sequence is subject to spatio-temporal perturbations. Lastly, we extend our previous transformation to accomodate the spatio-temporal perturbations.

Related Work There is a lot of work on monitoring temporal properties in dense time formalisms. This was then extended to checking how robustly (in the spatial sense) a signal satisfies a Metric Temporal Logic formula [7, 6]. This was then extended to consider spatio-temporal robustness of Signal Temporal Logic [5], a temporal logic that works with dense time and dense data. In [8] the authors considered robust satisfaction of Duration Calculus. In all of these works the authors define a multi-valued semantics for their temporal logic. For MLSL we have not been able to define a useful multi-valued semantics, because the atoms do not have quantitative data, which is crucial in the works mentioned above. In [2] the authors perform online monitoring of spatial properties for a driving car. In contrast to our work, they take a very low level view (little abstraction) and they can not easily check arbitrary spatial properties. In [19] the authors formalise traffic and traffic rules in a theorem prover. However, their goal is analysing meta properties, such as ambiguity of traffic rules, rather than automation. Urban MLSL is an extension of MLSL that allows for logical reasoning about traffic scenarios in an urban setting [12, 20].

2 Abstract Model for Motorways

We use an abstract formal model for motorway traffic [11], where the traffic configuration at a specific point in time is given by a *traffic snapshot*. In a traffic snapshot the motorway is represented by two dimensions, a discrete vertical dimension, which represents lanes and a continuous horizontal dimension, which represents the position along a lane. Then a *reservation* of a car represents space the car physically occupies plus some safety margin, which we assume to be the braking distance. When a car changes lanes it may have multiple adjacent reservations. A *claim* of a car represents that the car would like to reserve the claimed space. With claims we model the turn-signal of a real car. Additionally, a traffic snapshot has information about the speed and acceleration of each car. The evolution of traffic over time is modelled as a labelled transition system, where each state is a traffic snapshot. We give an example traffic snapshot and MLSL formulas to develop some intuition for the formalism.

Example 2.1. MLSL Formulas are evaluated on a restricted area of a traffic snapshot called *view*. We show an example traffic snapshot and view in Figure 1. In the traffic snapshot, with the given view, the formula

$$\langle \text{free} \frown \text{re}(e) \frown \text{free} \rangle$$

holds. Here, $\langle \cdot \rangle$ is an abbreviation and means that the subformula holds somewhere in the view, $\frown$ is used to separate adjacent segments within the lane, free indicates that the lane segment is free of claims and reservations and $\text{re}(e)$ means that the segment has a reservation from car e. Note that in formulas we use lower case letters to refer to cars. The formula

$$\langle \text{free} \frown \text{cl}(e) \frown \text{re}(d) \frown \text{free} \frown \text{re}(c) \frown \text{free} \rangle$$

also is satisfied by the traffic snapshot and the view in Figure 1. With $\text{cl}(e)$ we indicate that the lane segment has a claim of car e. Note that $\text{cl}(e)$ and $\text{re}(d)$ are not exclusive, i.e. in the lane segment where

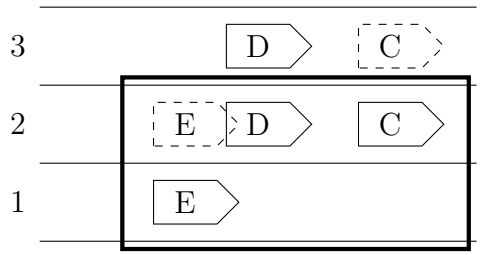

Figure 1: Visualisation of a traffic snapshot, where car C has a reservation (solid line) and a claim (dashed line), car D has two reservations and car E also has a reservation and a claim. The claim of car E and a reservation of car D overlap. Additionally, we show a view (rectangle with thick line). Note that one reservation of car D and the claim of car C are outside of the view.

the claim of E and the reservation of D overlap, both, $\mathrm{cl}(e)$ and $\mathrm{re}(d)$ are satisfied. We can stack formulas to express that on the lower lane the lower formula holds, and that on the upper lane the upper formula holds. That is, the formula

$$\begin{gathered}\text{free} \frown \mathrm{cl}(e) \frown \mathrm{re}(d) \frown \text{free} \frown \mathrm{re}(c) \frown \text{free} \\ \text{free} \frown \mathrm{re}(e) \frown \text{free}\end{gathered}$$

is satisfied with the complete view, not just somewhere within the view. △

Let $\mathbb{I}$ be a set of cars and $\mathbb{L}$ be a set of lanes let $\mathscr{P}(\mathbb{L})$ be the powerset over $\mathbb{L}$. The composition of data from the cars in $\mathbb{I}$ is a *traffic snapshot*. We add a function Ω, which gives the braking distance of a car, to the traffic snapshot from [11].

Definition 2.2 (Traffic Snapshot)**.** For every car C let $\mathrm{length}_{\mathrm{car}}(C)$ be the physical length of C. Then a traffic snapshot is defined as $TS = (pos, \Omega, spd, acc, res, clm)$, where $pos : \mathbb{I} \to \mathbb{R}$ is the position of the rear of a car, $\Omega : \mathbb{I} \to \mathbb{R}_{>0}$ is the length of a reservation of a car including its physical length, $spd : \mathbb{I} \to \mathbb{R}$ is the current speed, $acc : \mathbb{I} \to \mathbb{R}$ is the current acceleration, $res : \mathbb{I} \to \mathscr{P}(\mathbb{L})$ is the set of reserved lanes. $clm : \mathbb{I} \to \mathscr{P}(\mathbb{L})$ is the set of claimed lanes. △

We model the evolution of traffic snapshots as labelled transitions, where we use discrete and continuous transitions. The discrete transitions for a car C are to change the acceleration ($\mathrm{a}(C,a)$ with $a \in \mathbb{R}$), set a claim for a lane ($\mathrm{c}(C,n)$ with $n \in \mathbb{L}$), change an existing claim into a reservation $\mathrm{r}(C)$), withdraw an existing claim (wd $\mathrm{c}(C)$) and withdraw a reservation from a lane (wd $\mathrm{r}(C,n)$ with $n \in \mathbb{L}$). The continuous transitions are similar to delay transitions in timed automata, i.e. we update the data affected by time (here position, speed and the derived braking distance). To define the transitions we use substitution and function overriding, i.e. let $TS[f/f \oplus \{C \mapsto x\}]$ be TS, except that the function f is replaced by $f \oplus \{C \mapsto x\}$, which maps C to the value x and agrees on everything else with f.

Definition 2.3 (Transitions)**.** Let $n, n' \in \mathbb{L}$ with $n' \in \{n-1, n+1\}$ and $a, z \in \mathbb{R}$. Further, to compute the braking distance of a car we assume a maximum deceleration value $\mathrm{dec}_{\max}$ that all cars are capable off. We define

$$\begin{aligned}
TS \xrightarrow{\mathrm{a}(C,a)} TS' &\iff TS' = TS[acc/acc \oplus \{C \mapsto a\}] \\
TS \xrightarrow{\mathrm{c}(C,n)} TS' &\iff TS' = TS[clm/clm \oplus \{C \mapsto n\}] \wedge res(C) = \{n'\} \wedge clm(C) = \emptyset \\
TS \xrightarrow{\mathrm{r}(C)} TS' &\iff TS' = TS[res, clm/res \oplus \{C \mapsto res(C) \cup clm(C)\}, clm \oplus \{C \mapsto \emptyset\}]
\end{aligned}$$

$$TS \xrightarrow{\text{wd c}(C)} TS' \iff TS' = TS[clm/clm \oplus \{C \mapsto \emptyset\}]$$
$$TS \xrightarrow{\text{wd r}(C,n)} TS' \iff TS' = TS[res/res \oplus \{C \mapsto \{n\}\}] \wedge n \in res(C)$$
$$TS \xrightarrow{z} TS' \iff TS' = TS[pos, spd, \Omega/pos', spd', \Omega'] \text{ where}$$
$$pos' = \{C \mapsto pos(C) + spd(C) \cdot z + \frac{1}{2} acc(C) \cdot z^2 \mid C \in \mathbb{I}\}$$
$$spd' = \{C \mapsto acc(C) \cdot z + spd(C) \mid C \in \mathbb{I}\}$$
$$\Omega' = \{C \mapsto \frac{(spd(C) + acc(C) \cdot z)^2}{\text{dec}_{\max}} + \text{length}_{\text{car}}(C) | C \in \mathbb{I}\} \qquad \triangle$$

While we give a definition of Ω for all cars, our results also hold with a different definition of Ω for each car. Such an individual definition could depend on properties of the cars, e.g. one definition for light cars and another for heavy cars. However, our results only hold when the function used is a polynomial, i.e. we do not allow exponentiation and trigonometric functions.

Here we take the view that underlying a transition sequence, there is a *timed word* [3]. A timed word is a sequence of events and time stamps.

Definition 2.4 (Timed Words)**.** For a set of cars $\mathbb{I}$ and a car $C \in \mathbb{I}$ we denote the cars actions as $\Sigma_C = \{\text{c}(C,n), \text{r}(C), \text{wd c}(C), \text{wd r}(C,n), \text{a}(C,a) \mid n \in \mathbb{L}, a \in \mathbb{R}\}$ and the set of actions of all cars as $\Sigma = \bigcup_{C \in \mathbb{I}} \Sigma_C$. The joint behavior of the cars in $\mathbb{I}$ is a timed word $\rho = (\sigma, \tau)$ where $\sigma \in \Sigma^*$ and τ is a weakly monotonic increasing sequence of time stamps over $\mathbb{R}_{\geq 0}$. We assume that all timed words have as their last element in σ a special marker end $\notin \Sigma$. For a timed word $\rho = (\sigma, \tau)$ with $\sigma = \sigma_1 \dots \sigma_n$ and $\tau = \tau_1 \dots \tau_n$ we denote the *projection* to $\Sigma' \subseteq \Sigma$ as $\rho|\Sigma' = (\sigma', \tau')$ with $\sigma' = \sigma_{i_1} \dots \sigma_{i_k}$, $\tau' = \tau_{i_1} \dots \tau_{i_k}$ and $1 \leq i_1, \dots, i_k \leq n$ such that σ' is the longest subsequence of σ that only has letters from $\Sigma' \cup \{\text{end}\}$. Let the $\text{span}(\rho)$ of a timed word ρ be the interval $[0, \tau_n]$. We define the *time-bounded prefix* ρ_t with $t \in \text{span}(\rho)$ as $(\sigma_1, \tau_1) \dots (\sigma_i, \tau_i)(\text{end}, t)$, where i is the largest index such that $\tau_i \leq t$. Note that we might have $\tau_i = t$. $\triangle$

We define that the application of a timed word to a traffic snapshot gives a transition sequence. The idea is that we first let time advance to the ith time stamp and then perform the ith discrete action. Note that we interpret 'end' as a delay of zero time.

Definition 2.5 (From Timed Words to Transition Sequences)**.** Given a timed word $\rho = (\sigma, \tau)$ with $\sigma = \sigma_1 \dots \sigma_{n-1}\text{end}$ and $\tau = \tau_1 \dots \tau_n$ and a traffic snapshot TS_1, we define the transition sequence $\rho(TS_1)$ as

$$TS_1 \xrightarrow{\tau_1} TS_2 \xrightarrow{\sigma_1} TS_3 \xrightarrow{\tau_2 - \tau_1} \dots \xrightarrow{\tau_{n-1} - \tau_{n-2}} TS_{2n-2} \xrightarrow{\sigma_{n-1}} TS_{2n-1} \xrightarrow{\tau_n - \tau_{n-1}} TS_{2n} \xrightarrow{0} TS_{2n} \ .$$

Further, for $t \in \text{span}(\rho)$ we define the *time-bounded transition sequence until* t as $\rho_t(TS)$ and we denote the last traffic snapshot in $\rho_t(TS)$ as $\rho(TS)@t$, i.e. $\rho(TS)@t$ is the traffic snapshot at time t. $\triangle$

In the rest of this work we will only consider transition sequences that result from timed words, and that satisfy the constraints from Definition 2.3. Additionally, we assume that all transitions labelled with $\text{r}(C), \text{wd c}(C), \text{wd r}(C,n)$ change the state, i.e. a car makes a reservation only if it has a claim, it withdraws a claim only when it has a claim and it withdraws a reservation only if it has two reservations.

We give an example of a timed word and how we create a transition sequence from it. In our examples we give constants representing physical quantities always with their units, i.e. m for distances, s for time, m s^{-1} for speed, and m s^{-2} for acceleration.

Example 2.6. Let us assume that the global, maximal deceleration constant is given as $\text{dec}_{\max} = 12\text{m s}^{-2}$ and that each car has a physical length of 3m. Consider a timed word

$$\rho = (\text{wd r(D,3)}, 1\text{s})\ (\text{r(E)}, 1.1\text{s})\ (\text{wd r(E,2)}, 6.1\text{s})\ (\text{end}, 6.1\text{s})$$

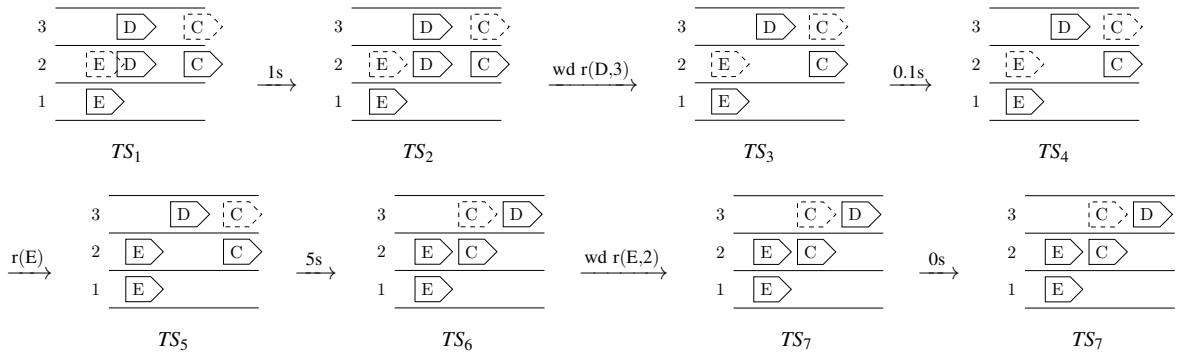

Figure 2: Visualisation of the transition sequence in Example 2.6. Claims are shown with dashed and reservations with solid lines. We do not show the last 0s transition to save space

and a traffic snapshot $TS = (pos, \Omega, spd, acc, res, clm)$ defined as

$$pos = \{\mathrm{C} \mapsto 60\mathrm{m}, \mathrm{D} \mapsto 16\mathrm{m}, \mathrm{E} \mapsto 6\mathrm{m}\} \qquad acc = \{\mathrm{C} \mapsto 0\mathrm{m\,s}^{-2}, \mathrm{D} \mapsto 0\mathrm{m\,s}^{-2}, E \mapsto 0\mathrm{m\,s}^{-2}\}$$
$$\Omega = \{\mathrm{C} \mapsto 6\mathrm{m}, \mathrm{D} \mapsto 30\mathrm{m}, \mathrm{E} \mapsto 15\mathrm{m}\} \qquad res = \{\mathrm{C} \mapsto \{2\}, \mathrm{D} \mapsto \{2,3\}, \mathrm{E} \mapsto \{1\}\}$$
$$spd = \{\mathrm{C} \mapsto 6\mathrm{m\,s}^{-1}, \mathrm{D} \mapsto 18\mathrm{m\,s}^{-1}, \mathrm{E} \mapsto 12\mathrm{m\,s}^{-1}\} \qquad clm = \{\mathrm{C} \mapsto \{3\}, \mathrm{D} \mapsto \emptyset, \mathrm{E} \mapsto \{2\}\}$$

Note that TS is a formulisation of the traffic snapshot from Figure 1. By applying ρ to TS, we get the transition sequence

$$\rho(TS) = TS \xrightarrow{1\mathrm{s}} TS_2 \xrightarrow{\text{wd r(D,3)}} TS_3 \xrightarrow{0.1\mathrm{s}} TS_4 \xrightarrow{\text{r(E)}} TS_5 \xrightarrow{5\mathrm{s}} TS_6 \xrightarrow{\text{wd r(E,2)}} TS_7 \xrightarrow{0\mathrm{s}} TS_7 \xrightarrow{0\mathrm{s}} TS_7$$

depicted in Figure 2. Note that the two 0s delays in the timed word above result from the delay between wd r$(\mathrm{E},2)$ and 'end', and from our representation of 'end' in the transition sequence as a 0s delay. We use this transition sequence as our running example in this work. △

In MLSL we reason about traffic configurations from the local perspective of a car, called *view*.

Definition 2.7 (View)**.** A view is a tuple $V = (L, X, E)$, where $L = [l, n] \subseteq \mathbb{L}$ is the interval of visible lanes, $X = [r, t] \subseteq \mathbb{R}$ is the extension of the visible lanes and $E \in \mathbb{I}$ is the owner of the view. We say that $V' = (L', X', E)$ is a subview of V if $L' \subseteq L$ and $X' \subseteq X$, where we interpret $[l', n'] = \emptyset$ if $l' > n'$ and $\emptyset \subseteq L$ for any set L. We define $V^{L'} = (L', X, E)$ and $V_{X'} = (L, X', E)$. △

Additionally, we assume a set of variables $CVar$ ranging over $\mathbb{I}$, a special variable ego and a variable valuation $\nu : CVar \cup \{ego\} \to \mathbb{I}$ such that $\nu(ego) = E$. We define that $M = (TS, V, \nu)$ is an MLSL model.

We lift transition sequences over traffic snapshots to transition sequences over models, as in [15] by moving the view along with its owner. Let ρ be a timed word and TS_1 a traffic snapshot with $\rho(TS_1) = TS_1 \xrightarrow{\lambda_1} \ldots \xrightarrow{\lambda_{n-1}} TS_n$, where $\lambda_1, \ldots, \lambda_{n-1} \in \Sigma \cup \mathbb{R}$. For $M_1 = (TS_1, V_1, \nu)$, $V_1 = (L, [r_1, t_1], E)$ we define

$$\rho(M_1) = M_1 \xrightarrow{\lambda_1} \ldots \xrightarrow{\lambda_{n-1}} M_n$$

with $M_i = (TS_i, V_i, \nu)$, $V_i = (L, [r_i, t_i], E)$ and $r_{i+1} = r_1 + (pos_{i+1}(E) - pos_i(E)), t_{i+1} = t_1 + (pos_{i+1}(E) - pos_i(E))$.

The syntax of MLSL is

$$\phi ::= \gamma = \gamma' \mid \mathrm{free} \mid \mathrm{re}(\gamma) \mid \mathrm{cl}(\gamma) \mid \ell = q \mid \neg\phi \mid \phi \wedge \phi \mid \exists c.\, \phi \mid \phi \frown \phi \mid \binom{\phi}{\phi}$$

where $\gamma, \gamma' \in CVar \cup \{ego\}$ and $q \in \mathbb{Q}$. We denote the set of all MLSL formulas with Φ. We briefly sketch the idea of the logic: The atom $\mathrm{re}(\gamma)$ (resp. $\mathrm{cl}(\gamma)$) is satisfied when the current view is filled by the reservation (resp. claim) of the car that γ points to. The atom free is satisfied if the current view does not have a subview where $\mathrm{re}(\gamma)$ or $\mathrm{cl}(\gamma)$ is satisfied for any car and $\ell = q$ is satisfied if the extension of the current view has length q. The horizontal chop $\phi_1 \frown \phi_2$ (resp. vertical chop $\binom{\phi_2}{\phi_1}$) is satisfied if we can cut the current view into two horizontally (resp. vertically) adjacent subviews on which ϕ_1 and ϕ_2 are satisfied. In the semantics of the vertical chop operator we follow [9], i.e. we distinguish whether the view contains any lanes before chopping.

Definition 2.8 (Semantics)**.** Let $c \in CVar$, $q \in \mathbb{Q}$ and $\gamma, \gamma' \in CVar \cup \{ego\}$. Given a traffic snapshot TS, a view $V = ([l,n],[r,t],E)$ and a valuation ν with $\nu(ego) = E$ we define the *satisfaction* of a formula by a model $M = (TS, V, \nu)$ as follows:

$$
\begin{array}{lll}
M \models \gamma = \gamma' & \text{iff} & \nu(\gamma) = \nu(\gamma') \\
M \models \text{free} & \text{iff} & (l \notin res(C) \cup clm(C) \text{ or } [pos(C), pos(C) + \Omega(C)] \cap (r,t) = \emptyset) \\
 & & \text{for every } C \in \mathbb{I}, \text{ and } l = n \text{ and } r < t \\
M \models \mathrm{re}(\gamma) & \text{iff} & l \in res(\nu(\gamma)) \text{ and } [r,t] \subseteq [pos(\nu(\gamma)), pos(\nu(\gamma)) + \Omega(\nu(\gamma))] \text{ and } l = n \text{ and } r < t \\
M \models \mathrm{cl}(\gamma) & \text{iff} & l \in clm(\nu(\gamma)) \text{ and } [r,t] \subseteq [pos(\nu(\gamma)), pos(\nu(\gamma)) + \Omega(\nu(\gamma))] \text{ and } l = n \text{ and } r < t \\
M \models \ell = q & \text{iff} & t - r = q \\
M \models \neg\phi & \text{iff} & M \not\models \phi \\
M \models \phi_1 \wedge \phi_2 & \text{iff} & M \models \phi_1 \text{ and } M \models \phi_2 \\
M \models \exists c.\, \phi & \text{iff} & (TS, V, \nu \oplus \{c \mapsto C\}) \models \phi, \text{ for some } C \text{ in } \mathbb{I} \\
M \models \phi_1 \frown \phi_2 & \text{iff} & (TS, V_{[r,s]}, \nu) \models \phi_1 \text{ and } (TS, V_{[s,t]}, \nu) \models \phi_2, \\
 & & \text{for some } s, \text{ where } r \leq s \leq t \\
M \models \binom{\phi_2}{\phi_1} & \text{iff} & l \leq n \text{ implies} \\
 & & \quad (TS, V^{[l,m]}, \nu) \models \phi_1 \text{ and } (TS, V^{[m+1,n]}, \nu) \models \phi_2 \\
 & & \quad \text{for some } m, \text{ where } l - 1 \leq m \leq n, \text{ and} \\
 & & l > n \text{ implies } (TS, V, \nu) \models \phi_1 \text{ and } (TS, V, \nu) \models \phi_2
\end{array}
$$

△

We use common abbreviations like *true*, *false*, $\vee$ and $\forall$. For an MLSL formula ϕ we also use the spatial *somewhere* modality from [11] that is defined as

$$\langle\phi\rangle \equiv true \frown \binom{true}{\phi} \frown true \ .$$

3 Monitoring Globally Properties

In this section we first formalise for an MLSL model M, a timed word ρ and an MLSL formula ϕ what the statement 'ϕ holds globally in $\rho(M)$' means. The intuition is that we check for every point in time t within the time span of ρ, whether the model in the transition sequence $\rho(M)$ at time t satisfies ϕ, which in symbols is $\rho(M)@t \models \phi$. Afterwards, we define a transformation that takes as inputs ρ, M and ϕ,

and creates a formula $\psi \equiv \psi_M \wedge \psi_\rho \implies \psi_\phi$ from the first-order theory of real-closed fields [21] (there called elementary algebra). In our transformation we mimic the afore mentioned intuition. The general idea is that ψ_M represents the initial model, ψ_ρ changes the transformed initial model and ψ_ϕ is checked on the changed model. We use a universally quantified variable t_f and freeze the transformed model at the time given by the value assigned to t_f and discard later changes. Then we check whether ψ_ϕ holds in the frozen model. As t_f is universally quantified and ranges over the time span of the timed word ρ, ψ is valid iff ϕ holds globally in $\rho(M)$.

In [15] the authors extend MLSL with branching CTL-like temporal modalities. As branching time modalities are not suited for monitoring, we define a linear time *globally* modality, which is satisfied if the subformula is satisfied at every point in time.

Definition 3.1 (Global Satisfaction)**.** A transition sequence $\rho(M)$ globally satisfies a spatial property ϕ (denoted as $\rho(M) \models_{\text{seq}} \Box\phi$) iff at every point in time t within the span of ρ the formula ϕ is satisfied. Formally,

$$\rho(M) \models_{\text{seq}} \Box\phi \text{ iff } \forall t \in \text{span}(\rho).\, \rho(M)@t \models \phi \ . \qquad \triangle$$

We consider formulas from first-order theory of real-closed fields with the signature $\langle \mathbb{R}, +, \cdot, 0, 1, <, \rangle$ and standard interpretation. The satisfiability problem of this logic is decidable [21]. We denote the set of all formulas as Ψ and the set of real-valued variables as *RVar*. This logic shares symbols with MLSL, such as $=$, $\neg$ and $\wedge$. However, from the context it will be clear to which logic symbols belong. We denote the *variable assignment* with $[\![\cdot]\!]$, which assigns variables a value.

In our transformation the state of a car C at a time point is given by the variables from *RVar* in the tuple $(res_{C,i}, res'_{C,i}, pos_{C,i}, \Omega_{C,i}, clm_{C,i}, acc_{C,i}, spd_{C,i})$. For any car C and timed word ρ let $\rho_C = (\sigma_C, \tau_C) = \rho|\Sigma_C$. Now, let $vars_\mathrm{d}$ (d for data) be a list of length $|\mathbb{I}|$ such that it has for each car C a list of length $|\sigma_C| + 1$ and at $vars_\mathrm{d}(C)(i)$ we have an aforementioned tuple of variables. Note that the lists for the cars may be of different lengths. We refer to the list that has for each car the first (resp. final) entry with $vars_{\mathrm{d,init}}$ (resp. $vars_{\mathrm{d,f}}$).

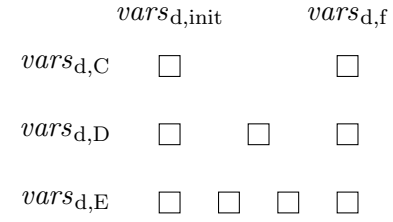

Figure 3: Visualisation of $vars_\mathrm{d}$ structure for the timed word from Example 2.6. Only the first and the last column represent the system at the same point in time

Example 3.2. Consider the timed word ρ from Example 2.6. Then let

$$\begin{aligned}
\rho_\mathrm{C} &= \rho|\Sigma_\mathrm{C} = (\text{end}, 6.1\text{s}) \\
\rho_\mathrm{D} &= \rho|\Sigma_\mathrm{D} = (\text{wd r}(\mathrm{D}, 3), 1\text{s})(\text{end}, 6.1\text{s}) \\
\rho_\mathrm{E} &= \rho|\Sigma_\mathrm{E} = (\text{r}(\mathrm{E}), 1.1\text{s})(\text{wd r}(\mathrm{E}, 2), 6.1\text{s})(\text{end}, 6.1\text{s})
\end{aligned}$$

For $\mathbb{I} = \{\mathrm{D}, \mathrm{C}, \mathrm{E}\}$ and ρ we show the structure of $vars_\mathrm{d}$ in Figure 3. $\triangle$

For real-valued variables, which we consider as not assigned, we introduce a special value $\bullet$ such that $\bullet \notin \mathbb{L}$. Given a traffic snapshot *TS* we assume w.l.o.g. that for all cars $C \in \mathbb{I}$ we have $res(C) = \{n, \bullet\}$ if C only reserves lane $n \in \mathbb{L}$ and $clm(C) = \{\bullet\}$ if C does not have a claim. Further, let $vars_\mathrm{d}$ be globally available.

Definition 3.3 (Transforming Initial Models)**.** For a traffic snapshot *TS* over a set of cars $\mathbb{I}$, let for a car C $n_C \in \mathbb{L}, n'_C \in \mathbb{L} \cup \{\bullet\}$ be the values in the set $res(C)$ and $n''_C \in \mathbb{L} \cup \{\bullet\}$ be the value in the set $clm(C)$.

With the variables $vars_{\mathrm{d,init}}$ we define

$$\mathrm{tr}_{\mathrm{init}}(TS) := \bigwedge_{C\in\mathbb{I}} pos_{C,1} = pos(C) \wedge res_{C,1} = n_C \wedge res'_{C,1} = n'_C \wedge spd_{C,1} = spd(C) \wedge acc_{C,1} = acc(C) \wedge clm_{C,1} = n''_C \wedge \Omega_{C,1} = \Omega(C) \quad . \quad \triangle$$

For each car $C \in \mathbb{I}$ the transformation of an action ($\mathrm{tr}_{\mathrm{act}}$) is split into a transformation of a delay action ($\mathrm{tr}_{\mathrm{delay}}$) and into a transformation of a discrete action ($\mathrm{tr}_{\mathrm{d-act}}$). We point out that we treat the 'end' marker as an action that does not change anything.

Definition 3.4 (Transforming Actions)**.** For some car $C \in \mathbb{I}$ and an index $i \in \mathbb{N}$ let $\sigma_i \in \Sigma_C \cup \{\mathrm{end}\}$. For $n \in \mathbb{L}, a \in \mathbb{R}$ and a variable $z \in RVar$ indicating a delay we define

$$\mathrm{tr}_{\mathrm{act}}(\sigma_i, z, i, C) := \mathrm{tr}_{\mathrm{d-act}}(\sigma_i, i, C) \wedge \mathrm{tr}_{\mathrm{delay}}(z, i, C)$$

$$\mathrm{tr}_{\mathrm{d-act}}(\sigma_i, i, C) := \begin{cases} clm_{C,i+1} = n \wedge \mathrm{id}_{C,i}(res, res', acc) & \text{if } \sigma_i = \mathrm{c}(C,n) \\ res'_{i+1} = clm_{C,i} \wedge clm_{C,i+1} = \bullet \wedge \mathrm{id}_{C,i}(res, acc) & \text{if } \sigma_i = \mathrm{r}(C) \\ res_{C,i+1} = n \wedge res'_{C,i+1} = \bullet \wedge \mathrm{id}_{C,i}(clm, acc) & \text{if } \sigma_i = \mathrm{wd\ r}(C,n) \\ clm_{i+1} = \bullet \wedge \mathrm{id}_{C,i}(res, res', acc) & \text{if } \sigma_i = \mathrm{wd\ c}(C) \\ acc_{C,i+1} = a \wedge \mathrm{id}_{C,i}(res, res', clm) & \text{if } \sigma_i = \mathrm{a}(C,a) \\ \mathrm{id}_{C,i}(res, res', clm, acc) & \text{if } \sigma_i = \mathrm{end} \end{cases}$$

$$\mathrm{tr}_{\mathrm{delay}}(z, i, C) := pos_{C,i+1} = pos_{C,i} + spd_{C,i} \cdot z + \frac{1}{2} acc_{C,i} \cdot z^2 \wedge spd_{C,i+1} = acc_{C,i} \cdot z + spd_{C,i} \wedge \Omega_{C,i+1} = \frac{(spd_{C,i} + acc_{C,i} \cdot z)^2}{\mathrm{dec}_{\max}} + \mathrm{length}_{\mathrm{car}}(C)$$

with $\mathrm{id}_{C,i}(res) := res_{C,i+1} = res_{C,i}$ and similar for the other variables. $\triangle$

Now we can define a transformation for the time-bounded prefix $\rho_{t_{\mathrm{f}}}$, where ρ is a timed word and $t_{\mathrm{f}} \in RVar$. The model at time t_{f} is stored in the variables $vars_{\mathrm{d,f}}$. To achieve this we ignore all changes after time t_{f}. To define our transformation of time-bounded prefixes, we assume a structure $vars_{\mathrm{t}}$ that has entries $t_{C,i} \in RVar$, similar to $vars_{\mathrm{d}}$. For a car C and a projected timed word $\rho|\Sigma_C = (\sigma_C, \tau_C)$ we identify time stamps with variables that have the constraint $t_{C,i+1} = \tau_{C,i}$ and $t_{C,1} = 0$.

Definition 3.5 (Transforming Timed-Bounded Transition Prefixes)**.** For $C \in \mathbb{I}$ let $\rho = (\sigma, \tau)$, $\rho|\Sigma_C = (\sigma_C, \tau_C)$ and $t_{\mathrm{f}} \in RVar$. Then we define

$$\mathrm{tr}_{\mathrm{word}}(\sigma, t_{\mathrm{f}}) := \bigwedge_{C\in\mathbb{I}} \mathrm{tr}_{C-\mathrm{word}}(\sigma_C, t_{\mathrm{f}}, C)$$

$$\mathrm{tr}_{C-\mathrm{word}}(\sigma, t_{\mathrm{f}}, C) := \bigwedge_{i\in\{1,\ldots,|\sigma|\}}$$

$$(t_{C,i} \le t_{C,i+1} \le t_{\mathrm{f}} \implies \mathrm{tr}_{\mathrm{act}}(t_{C,i+1} - t_{C,i}, \sigma_i, i, C)) \quad (1)$$

$$\wedge\, (t_{C,i} \le t_{\mathrm{f}} < t_{C,i+1} \implies \mathrm{tr}_{\mathrm{act}}(t_{\mathrm{f}} - t_{C,i}, \mathrm{end}, i, C)) \quad (2)$$

$$\wedge\, (t_{\mathrm{f}} < t_{C,i} \implies \mathrm{tr}_{\mathrm{act}}(0\mathrm{s}, \mathrm{end}, i, C)) \quad (3)$$

$\triangle$

The first implication (1) considers the case where the effect of the action takes place before or at time t_f. Hence, we completely represent the effect in our transformation. If the condition of the second implication (2) is satisfied, we know that delaying by $t_{C,i+1} - t_{C,i}$ time units takes us past t_f. Hence, we only delay by $t_\mathrm{f} - t_{C,i}$ time units, exactly to time point t_f and do not transform σ_i. Instead of σ_i we transform 'end', which ensures that all variables retain their values. The third implication (3) ensures that we do not manipulate the model anymore, after time point t_f. Note that in the conditions of the implications for each i exactly one condition is satisfied.

We need a method to check if an MLSL model satisfies an MLSL formula. In [9] the authors defined a transformation to check satisfiability of an MLSL formula ϕ that is restricted to a finitely bounded set of cars (called well-scoped MLSL with scopes). Their transformation creates a quantified linear integer-real arithmetic formula that is valid iff ϕ is satisfiable. We simplify their transformation to instead check whether for a *given model M* it holds that $M \models \phi$. The adapted transformation takes two parameters: the first is a tuple $\Upsilon = (CS, l, n, x_{\mathrm{f,l}}, x_{\mathrm{f,r}}, \nu)$, defining the cars to consider (here we have $CS = \mathbb{I}$), the current lanes $[l,n]$ with $l,n \in \mathbb{N}$, the current extension as variables $x_\mathrm{l}, x_\mathrm{r} \in \mathit{RVar}$, and the valuation function ν. The second parameter is the MLSL formula. The formula that $\mathrm{tr_f}$ creates is from the first-order theory of real-closed fields and represents the semantics of MLSL. For this the formula creates suitable constraints on $\mathit{vars}_{\mathrm{d,f}}$. Note that negation in MLSL is represented with $\mathrm{tr_f}$ by negation in the first-order theory of real-closed fields, i.e. for all Υ and MLSL formulas ϕ we have $\mathrm{tr_f}(\Upsilon, \neg\phi) = \neg\mathrm{tr_f}(\Upsilon, \phi)$. The following claim states that we can algorithmically determine if an MLSL formula is satisfied by a model.

Claim 3.6. *Let $M = (TS, V, \nu)$ with $V = ([l,n],[r,t],E)$ and let for $\mathbb{I}$ the variables $\mathit{vars}_{\mathrm{d,init}}$ be available. We constrain $x_\mathrm{l}, x_\mathrm{r} \in \mathit{RVar}$ with $x_\mathrm{l} = r, x_\mathrm{r} = t$ and define $\Upsilon = (\mathbb{I}, l, n, x_\mathrm{l}, x_\mathrm{r}, \nu)$. Then for any MLSL formula ϕ we have*

$$\mathrm{tr_{init}}(TS) \implies \mathrm{tr_f}(\Upsilon, \phi) \text{ is valid} \qquad \text{iff} \qquad M \models \phi \ ,$$

where $\mathrm{tr_f}(\Upsilon, \phi)$ is evaluated on the variables $\mathit{vars}_{\mathrm{d,init}}$.

Now we can define our transformation to check globally properties. The intuition of the transformation is that it checks if we can stop the evolution of $\rho(M)$ at all time points t_f and store the model at that time in the variables subscripted with 'f' and then evaluate ϕ on this stored model. Note that we use the variables $x_{\mathrm{f,l}}, x_{\mathrm{f,r}}$ to represent the extension at time t_f.

Definition 3.7 (Transforming Globally Properties)**.** Given a model M and a timed word ρ over a finite set $\mathbb{I}$ we use the variables $x_{\mathrm{f,l}}, x_{\mathrm{f,r}} \in \mathit{RVar}$ with the constraints $x_{\mathrm{f,l}} = r + (\mathit{pos}_{E,\mathrm{f}} - \mathit{pos}_{E,1})$ and $x_{\mathrm{f,r}} = t + (\mathit{pos}_{E,\mathrm{f}} - \mathit{pos}_{E,1})$. Let $\Upsilon = (\mathbb{I}, l, n, x_{\mathrm{f,l}}, x_{\mathrm{f,r}}, \nu)$, then for an MLSL formula ϕ we define

$$\mathrm{tr}_\Box(\rho, M, \phi) := \forall t_\mathrm{f} \in \mathrm{span}(\rho).\, (\mathrm{tr_{word}}(\sigma, t_\mathrm{f}) \wedge \mathrm{tr_{init}}(TS)) \implies \mathrm{tr_f}(\Upsilon, \phi) \ ,$$

where $\mathrm{tr_f}(\Upsilon, \phi)$ is evaluated over $\mathit{vars}_{\mathrm{d,f}}$. △

Claim 3.8. *Given a timed word ρ, an MLSL model M and an MLSL formula ϕ*

$$\rho(M) \models_{\mathrm{seq}} \Box\phi \ \text{ iff } \ \mathrm{tr}_\Box(\rho, M, \phi) \text{ is valid} \ .$$

The previous claim states that we can reduce checking $\rho(M) \models_{\mathrm{seq}} \Box\phi$ to checking $\mathrm{tr}_\Box(\rho, M, \phi)$ for validity. This is equivalent to $\neg\mathrm{tr}_\Box(\rho, M, \phi)$ being unsatisfiable. As the satisfiability of first-order theory of real closed fields is decidable [21], we get the following theorem, assuming that the above claim holds.

Theorem 3.9. *It is decidable whether an MLSL formula holds globally in an MLSL transition sequence.*

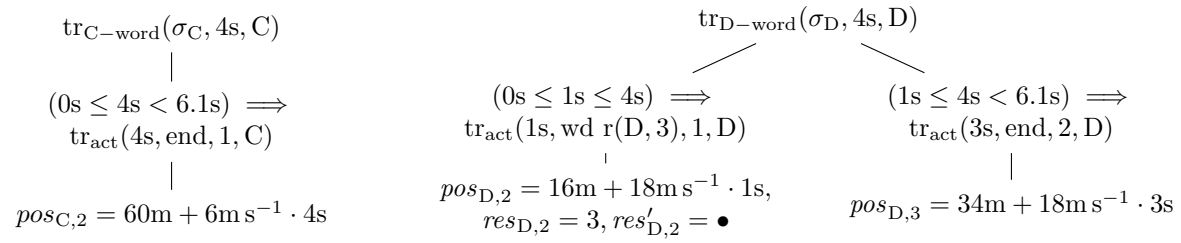

Figure 4: Transformations of the words projected to actions from car C and car D

Example 3.10. Consider the timed word ρ and the traffic snapshot *TS* from Example 2.6 and the MLSL formula *no potential collision*

$$\mathrm{npc} \equiv \forall c, c'.\, c \neq c' \implies \neg\langle(\mathrm{cl}(c) \vee \mathrm{re}(c)) \wedge (\mathrm{cl}(c') \vee \mathrm{re}(c'))\rangle \ ,$$

which is a generalisation of the potential collision formula from [11]. The formula npc states that nowhere in the current view, there is an overlap of the claims or reservations from two different cars. Let the view be $V = ([1,3],[0,90],\mathrm{E})$ and the valuation be $\nu = \{ego \mapsto \mathrm{E}\}$, then we define $M = (TS, V, \nu)$. We give an overview of how our procedure works to find that $\rho(M) \models_{\mathrm{seq}} \Box\,\mathrm{npc}$ does not hold.

To test whether 'there is never a potential collision' holds in $\rho(M)$ we check $\mathrm{tr}_{\Box}(\rho, M, \mathrm{npc})$ for validity. We show that $\mathrm{tr}_{\Box}(\rho, M, \mathrm{npc})$ is not valid by giving a satisfying assignment for its negation. The negation $\neg\mathrm{tr}_{\Box}(\rho, M, \mathrm{npc})$ evaluates to

$$\exists t_{\mathrm{f}} \in \mathrm{span}(\rho).\, \mathrm{tr}_{\mathrm{word}}(\sigma, t_{\mathrm{f}}) \wedge \mathrm{tr}_{\mathrm{init}}(TS)) \wedge \neg\mathrm{tr}_{\mathrm{f}}(\Upsilon, \mathrm{npc}) \ .$$

The formula npc is violated already in the initial model, because the claim of E overlaps with the reservation of D (cf. Figure 2). However, to give a better insight into our construction we choose to show that npc is violated *during* the transition from TS_5 to TS_6, at time $[\![t_{\mathrm{f}}]\!] = 4\mathrm{s}$.

We show the constraints generated by $\mathrm{tr}_{\mathrm{word}}(\sigma, 4\mathrm{s})$ for the cars C, D in Figure 4. We see that the position of C at time $[\![t_{\mathrm{f}}]\!] = 4\mathrm{s}$ is its initial position, plus the distance covered in 4s, i.e. $[\![pos_{C,2}]\!] = 84\mathrm{m}$. For car D we see that at time $[\![t_{\mathrm{D},2}]\!] = 1\mathrm{s}$ the withdrawal of a reservation is performed and that the position of D is updated to $[\![pos_{\mathrm{D},2}]\!] = 34\mathrm{m}$. Then, at time $[\![t_{\mathrm{f}}]\!] = 4\mathrm{s}$ car D is moved for 3s multiplied with its speed to $[\![pos_{\mathrm{D},3}]\!] = 88\mathrm{m}$.

We have $\neg\mathrm{tr}_{\mathrm{f}}(\Upsilon, \mathrm{npc}) = \mathrm{tr}_{\mathrm{f}}(\Upsilon, \neg\mathrm{npc})$. The formula $\neg$npc is evaluated on $vars_{\mathrm{d,f}}$ and the view updated by the movement of E. After 4s car E has moved $12\mathrm{m\,s^{-1}} \cdot 4\mathrm{s} = 48\mathrm{m}$. Hence, the updated left and right extension of the view are $[\![x_{\mathrm{l,f}}]\!] = 48\mathrm{m}$ and $[\![x_{\mathrm{r,f}}]\!] = 138\mathrm{m}$. Now we can check if $\neg$npc, which states that there is a subview where the claims or reservations of two different cars overlap, is satisfied. As C has a braking distance of $[\![\Omega_{\mathrm{D},3}]\!] = 6\mathrm{m}$ it claims the interval $[84\mathrm{m}, 84\mathrm{m} + 6\mathrm{m}]$ on lane 3. As car D has a reservation on lane 3 and its position is within $[84\mathrm{m}, 90\mathrm{m}]$ the claim of C and the reservation of D overlap. Thus, we have shown $\rho(M) \not\models_{\mathrm{seq}} \Box\,\mathrm{npc}$. △

4 Monitoring Globally Properties with Imprecise Information

In this section we extend our transformation to check, whether an MLSL formula holds globally in a transition sequence with ε-δ-robustness. This allow us to check if, e.g. a behavior given as a transition sequence is *barely safe* or if it is *robustly safe*.

Here, we consider errors in positional data and imprecisions of when reservations and claims are set and withdrawn. Similarity on timed words has originally been defined in [10]. However, usually the requirement is imposed that the order of events is equal in similar words. For distributed systems this requirement seems too strong. Here, we weaken this requirement and allow the order of independent actions to change in similar words. For a single car C, setting and withdrawing claims and reservations are independent of changing acceleration. Between different cars, all actions are independent. We first define an *independence relation* for actions.

Definition 4.1 (Independence Relation)**.** Let Σ_C be the action alphabet for car C. We define the *independence relation* as

$$\begin{aligned} I_C = &(\{\mathrm{c}(C,n), \mathrm{r}(C), \text{wd c}(C), \text{wd r}(C,n) \mid n \in \mathbb{L}\} \times \{\mathrm{a}(C,a) \mid a \in \mathbb{R}\}) \\ &\cup (\{\mathrm{a}(C,a) \mid a \in \mathbb{R}\} \times \{\mathrm{c}(C,n), \mathrm{r}(C), \text{wd c}(C), \text{wd r}(C,n) \mid n \in \mathbb{L}\}) \\ I = &\bigcup_{C,C' \in \mathbb{I}, C \neq C'} (\Sigma_C \times \Sigma_{C'}) \cup \bigcup_{C \in \mathbb{I}} I_C \end{aligned}$$ △

We define that two timed words have *equal causality* iff their untimed words are in the same *equivalence class* in the sense of Mazurkiewicz traces [16]. Two words are in the same equivalence class iff we can create one word from the other by repeatedly swapping letters that are adjacent and independent.

Definition 4.2 (Causality Equivalence)**.** Let I be our independence relation. For two words $\sigma, \sigma' \in \Sigma^*$ with $\sigma = \sigma_1 \dots \sigma_n$ we define that σ and σ' are in the same *equivalence class* (denoted $\sigma \in [\sigma']$) as

$$\begin{aligned} \sigma \in [\sigma'] \text{ iff } \sigma = \sigma' \text{ or } &\text{ there is } \sigma'' \in \Sigma^* \text{ and } i \in \{1, \dots, n\} \text{ such that} \\ &(\sigma_i, \sigma_{i+1}) \in I \text{ and } \sigma'' = \sigma_1 \dots \sigma_{i-1} \sigma_{i+1} \sigma_i \sigma_{i+2} \dots \sigma_n \text{ and } \sigma'' \in [\sigma'] \end{aligned}$$

Two timed words $\rho = (\sigma, \tau), \rho' = (\sigma', \tau')$ are *causally equivalent* iff $\sigma \in [\sigma']$. △

Example 4.3. Consider σ from Example 2.6 and $\sigma_1, \sigma_2, \sigma_3$ shown below:

$$\begin{aligned} \sigma &= \text{wd r(D,3) r(E) wd r(E,2) end} & \sigma_1 &= \text{r(E) wd r(D,3) wd r(E,2) end} \\ \sigma_2 &= \text{r(E) wd r(E,2) wd r(D,3) end} & \sigma_3 &= \text{wd r(D,3) wd r(E,2) r(E) end} \end{aligned}$$

As $(\text{wd r(D,3)}, \text{r(E)}) \in I$ we have $\sigma_1 \in [\sigma]$. Further, $(\text{wd r(D,3) wd r(E,2)}) \in I$ means that $\sigma_2 \in [\sigma_1]$. By transitivity, this implies $\sigma_2 \in [\sigma]$. However, $(\text{wd r(E,2)}, \text{r(E)}) \notin I$, which means that $\sigma_3 \notin [\sigma]$. △

To formalise similarity usually metrics on a set are introduced to define distances between elements. Then, similarity can be quantified with these metrics. To capture positional similarity of two models we assign a distance of ∞ if any other data than position, extension or sensor function differ. Otherwise, we assign the maximal difference of these values. The definition is taken from [17].

Definition 4.4 (Metric on MLSL Models)**.** Given two models MLSL M, M' we define $d_{\text{model}}(M, M') := \infty$ if $res \neq res'$ or $clm \neq clm'$ or $L \neq L'$ or $v \neq v'$ and

$$d_{\text{model}}(M, M') := \max_{C \in \mathbb{I}} \{|pos(C) - pos'(C)|, |pos(C) + \Omega(C) - (pos'(C) + \Omega'(C))|, |r - r'|, |t - t'|\}$$

otherwise. For $\delta \in \mathbb{R}_{>0}$ we say that two models M, M' are δ-similar, if $d_{\text{model}}(M, M') \leq \delta$. △

Additionally, we define a metric on timed words. We assume that between two similar timed words the time stamps of all acceleration actions are equal. The reason for this restriction is that if we allow acceleration time stamps to differ, perturbations may accumulate. In this work we do not consider such issues. Furthermore, we require that the time span of two similar timed words is equal. This is a technical restriction that likely can be removed.

Definition 4.5 (Metric on Timed Words)**.** Let $\Sigma_a = \{a(C,a) \mid C \in \mathbb{I}, a \in \mathbb{R}\}$. Given two timed words ρ, ρ' we define $d_{\text{time}}(\rho,\rho') = \infty$ if they are not causally equal, $\rho|\Sigma_a \neq \rho'|\Sigma_a$, or if $\text{span}(\rho) \neq \text{span}(\rho')$. Otherwise, for a car C let $\rho_C = (\sigma_C, \tau_C) = \rho|\Sigma_C \setminus \Sigma_a$ and $\rho'_C = (\sigma'_C, \tau'_C) = \rho'|\Sigma_C \setminus \Sigma_a$, where σ_C, σ'_C both have length n_C. We define

$$d_{\text{time}}(\rho,\rho') := \max_{C\in\mathbb{I}}\{d_C(\rho_C,\rho'_C)\} \qquad \text{and} \qquad d_C(\rho_C,\rho'_C) := \max_{i\in\{1,\dots,n_C\}}\{|\tau_{C,i} - \tau'_{C,i}|\} \ . \qquad \triangle$$

Note that d_C essentially is taken from [10]. Further, we point out that because ρ, ρ' in the above definition are causally equivalent, for $\rho|\Sigma_C \setminus \Sigma_a = (\sigma_C, \tau_C)$ and $\rho'|\Sigma_C \setminus \Sigma_a = (\sigma'_C, \tau'_C)$ we have $\sigma_C = \sigma'_C$. We lift the metric on timed words to a metric on transition sequences. For two models M, M' we define $d_{\text{seq}}(\rho(M), \rho'(M')) = \infty$ if $M \neq M'$ and otherwise

$$d_{\text{seq}}(\rho(M), \rho'(M')) := d_{\text{time}}(\rho,\rho') \ .$$

As for models, two transition sequences are ε-similar, if d_{seq} assigns them a distance $\leq \varepsilon$.

We define that a model δ-robustly satisfies a formula if all δ-similar models also satisfy the formula.

Definition 4.6 (Robust Satisfaction of MLSL Formulas)**.** Given a model M, a desired error allowance $\delta \in \mathbb{R}_{>0}$ and a formula ϕ, we define that M satisfies ϕ with robustness δ as

$$M \models^{\delta} \phi \ \text{ iff } \ \forall M'.\, d_{\text{model}}(M,M') \leq \delta \implies M' \models \phi \ . \qquad \triangle$$

We define what it means for a transition sequence to robustly satisfy $\Box\phi$.

Definition 4.7 (Robust Global Satisfaction)**.** Let M be an initial model, and let ρ be a timed word. Then for a formula ϕ, an allowed spatial error $\delta \in \mathbb{R}_{>0}$ and an allowed temporal error $\varepsilon \in \mathbb{R}_{>0}$, we define

$$\begin{aligned} &\rho(M) \models^{\delta}_{\text{seq}} \Box\phi \iff \forall t.\, t \in \text{span}(\rho) \implies \rho(M)@t \models^{\delta} \phi \\ &\rho(M) \models^{\varepsilon,\delta}_{\text{seq}} \Box\phi \ \text{ iff } \ \forall \rho(M)'.\, d_{\text{seq}}(\rho(M), \rho(M)') \leq \varepsilon \implies \rho(M)' \models^{\delta}_{\text{seq}} \Box\phi \end{aligned} \qquad \triangle$$

For example, with our notion of similarity the two timed words $\rho = (a(C,5,1))(r(C),1.1)$ and $\rho' = (r(C),1)(a(C,5,1.1))$ are 0.1-similar. However, in our transformation it is cumbersome to consider for a single car different possible sequences of events. Hence, we assume that two discrete actions of the same car are strictly more than 2ε time units apart. This assumption has the additional benefit that when we consider two ε-similar timed words, they are causally equal because all dependent actions have the same order in both timed words.

Assumption. For a timed word ρ and for any car C let $\rho_C = (\sigma_C, \tau_C) = \rho|\Sigma_C$ and let σ_C be of length n_C. Then, for $\varepsilon \in \mathbb{R}_{>0}$ we assume

$$\min_{C\in\mathbb{I}, i,j\in\{1,\dots,n_C\} \text{ with } i\neq j}\{|\tau_{C,i} - \tau_{C,j}|\} > 2\varepsilon \ . \qquad \triangle$$

Note that this is not a discretisation, as, e.g. actions from different cars may still be arbitrary close.

Before we can define the transformation we introduce perturbed versions of $vars_t$ and $vars_{d,f}$, which we call $\widetilde{vars}_t$ and $\widetilde{vars}_d$. We need to operate on the unperturbed, and the perturbed variables. To the transformation from Section 3 we add $\tilde{d}$ to indicate that we operate on the perturbed data variables, and d otherwise. Further, we do not need the unperturbed temporal variables, which we indicate with $\tilde{t}$. With $\forall\widetilde{vars}_t$ or $\forall\widetilde{vars}_{d,f}$ we mean that all variables in $\widetilde{vars}_t$ or $\widetilde{vars}_{d,f}$ are universally quantified, and

similar for $\exists\widetilde{vars}_{\mathrm{t}}$ or $\exists\widetilde{vars}_{\mathrm{d,f}}$. Additionally, we define for $C \in \mathbb{I}, i \in \mathbb{N}$ that $pos\Omega_{C,i} = pos_{C,i} + \Omega_{C,i}$ and $\widetilde{pos\Omega}_{C,i} = \widetilde{pos}_{C,i} + \widetilde{\Omega}_{C,i}$ and for $v, v' \in RVar, r \in \mathbb{R}$ let $v \in v' \pm r = v' - r \leq v \leq v' + r$.

To check if ϕ holds ε-δ-robustly we first perturb the timestamps of non-acceleration events ($\mathrm{shake}_\varepsilon$). Then we encode the temporally perturbed transition sequence ($\mathrm{tr}^{\mathrm{d}}_{\mathrm{init}}$ and $\mathrm{tr}^{\mathrm{d},\tilde{\mathrm{t}}}_{\mathrm{word}}$), and finally we evaluate the MLSL formula on the perturbed final model (shake_δ and $\mathrm{tr}^{\tilde{\mathrm{d}}}_{\mathrm{f}}$).

Definition 4.8 (Transforming Robust Globally Properties)**.** For a timed word ρ and a car C let $(\sigma_C, \tau_C) = \rho|\Sigma_C$ with σ_C having length n_C, $\Sigma_{\mathrm{a}} = \{\mathrm{a}(C,a) \mid a \in \mathbb{R}\}$, $\tilde{x}_{\mathrm{l,f}}, \tilde{x}_{\mathrm{r,f}} \in RVar$ and $\Delta_E = pos_{E,\mathrm{f}} - pos_{E,1}$. Then, given a model $M = (TS, V, \nu)$ with $V = ([l,n],[r,t],E)$, a tuple $\Upsilon = (\mathbb{I}, l, n, \tilde{x}_{\mathrm{l,f}}, \tilde{x}_{\mathrm{r,f}}, \nu)$ an MLSL formula ϕ and $\varepsilon, \delta \in \mathbb{R}_{>0}$ we define

$$\begin{aligned}
\mathrm{tr}^{\varepsilon,\delta}_{\square}(\rho, M, \phi) := {} & \forall\widetilde{vars}_{\mathrm{d,f}}, \tilde{x}_{\mathrm{l,f}}, \tilde{x}_{\mathrm{r,f}}.\, \forall\widetilde{vars}_{\mathrm{t}}, t_{\mathrm{f}} \in \mathrm{span}(\rho).\\
& \mathrm{tr}^{\mathrm{d}}_{\mathrm{init}}(TS) \wedge \mathrm{shake}_\varepsilon(\rho) \wedge \mathrm{shake}_\delta \wedge \mathrm{tr}^{\mathrm{d},\tilde{\mathrm{t}}}_{\mathrm{word}}(\sigma, t_{\mathrm{f}}) \implies \mathrm{tr}^{\tilde{\mathrm{d}}}_{\mathrm{f}}(\Upsilon, \phi)\\
\mathrm{shake}_\varepsilon(\rho) := {} & \bigwedge_{C \in \mathbb{I}, i \in \{1,\ldots,n_C-1\}} (\sigma_{C,i} \notin \Sigma_{\mathrm{a}} \implies \tilde{t}_{C,i+1} \in \tau_{C,i} \pm \varepsilon) \wedge (\sigma_{C,i} \notin \Sigma_{\mathrm{a}} \implies \tilde{t}_{C,i+1} = \tau_{C,i})\\
\mathrm{shake}_\delta := {} & \bigwedge_{C \in \mathbb{I}} \widetilde{pos}_{C,\mathrm{f}} \in pos_{C,\mathrm{f}} \pm \delta \wedge \widetilde{pos\Omega}_{C,\mathrm{f}} \in pos\Omega_{C,\mathrm{f}} \pm \delta \wedge \tilde{x}_{\mathrm{l,f}} \in (r + \Delta_E) \pm \delta \wedge\\
& \tilde{x}_{\mathrm{r,f}} \in (t + \Delta_E) \pm \delta \wedge \widetilde{clm}_{C,\mathrm{f}} = clm_{C,\mathrm{f}} \wedge \widetilde{res}_{C,\mathrm{f}} = res_{C,\mathrm{f}} \wedge \widetilde{res}'_{C,\mathrm{f}} = res'_{C,\mathrm{f}}
\end{aligned} \qquad \triangle$$

Claim 4.9. *Given a timed word ρ and an MLSL model M and an MLSL formula ϕ and $\varepsilon, \delta \in \mathbb{R}_{>0}$ we have*

$$\rho(M) \models^{\varepsilon,\delta}_{\mathrm{seq}} \square\phi \text{ iff } \mathrm{tr}^{\varepsilon,\delta}_{\square}(\rho, M, \phi) \text{ is valid } .$$

From the previous claim it follows that $\rho(M) \models^{\varepsilon,\delta}_{\mathrm{seq}} \square\phi$ holds iff $\mathrm{tr}^{\varepsilon,\delta}_{\square}(\rho, M, \phi)$ is valid. This is equivalent to $\neg\mathrm{tr}^{\varepsilon,\delta}_{\square}(\rho, M, \phi)$ being unsatisfiable. We assume that the above claim holds. Then, as satisfiability is decidable for the first-order theory of real-closed fields [21], we get the following theorem.

Theorem 4.10. *For $\varepsilon, \delta \in \mathbb{R}_{>0}$ it is decidable whether an MLSL formula holds globally in an MLSL transition sequence with ε-δ-robustness.*

Example 4.11. Consider the timed word $\rho = (\sigma, \tau)$ and traffic snapshot TS from Example 2.6 and the initial model $M = (TS, V, \nu)$ with $V = ([1,3],[0,90],\mathrm{E})$, $\nu = \{ego \mapsto \mathrm{E}\}$ from Example 3.10. We use the formula

$$\mathrm{safe} \equiv \forall c, c'.\, c \neq c' \implies \neg\langle \mathrm{re}(c) \wedge \mathrm{re}(c') \rangle$$

from [11], which states that there do not exist two different cars with overlapping reservations. In the following let $\varepsilon = 0.1\mathrm{s}$ and $\delta = 1\mathrm{m}$. To determine that $\square\mathrm{safe}$ does *not* hold ε-δ-robustly in $\rho(M)$, we give a satisfying assignment for the formula $\neg\mathrm{tr}^{0.1\mathrm{s},1\mathrm{m}}_{\square}(\rho, M, \mathrm{safe})$. This formula evaluates to

$$\exists\widetilde{vars}_{\mathrm{d,f}}, \tilde{x}_{\mathrm{l,f}}, \tilde{x}_{\mathrm{r,f}}.\, \exists\widetilde{vars}_{\mathrm{t}}, t_{\mathrm{f}} \in \mathrm{span}(\rho).\, \mathrm{tr}^{\mathrm{d}}_{\mathrm{init}}(TS) \wedge \mathrm{shake}_{0.1\mathrm{s}}(\rho) \wedge \mathrm{shake}_{1\mathrm{m}} \wedge \mathrm{tr}^{\mathrm{d},\tilde{\mathrm{t}}}_{\mathrm{word}}(\sigma, t_{\mathrm{f}}) \wedge \neg\mathrm{tr}^{\tilde{\mathrm{d}}}_{\mathrm{f}}(\Upsilon, \mathrm{safe}) \ ,$$

where $\Upsilon = (\mathbb{I}, 1, 3, \tilde{x}_{\mathrm{l,f}}, \tilde{x}_{\mathrm{r,f}}, \nu)$. Further, we point out that similar to Example 3.10 we have $\neg\mathrm{tr}^{\tilde{\mathrm{d}}}_{\mathrm{f}}(\Upsilon, \mathrm{safe}) = \mathrm{tr}^{\tilde{\mathrm{d}}}_{\mathrm{f}}(\Upsilon, \neg\mathrm{safe})$.

We give an explanation of how our construction works for this example. The perturbed variables $\tilde{t}_{\mathrm{D},2}, \tilde{t}_{\mathrm{E},2}$ from $\widetilde{vars}_{\mathrm{t}}$ represent perturbations of the time stamps $\tau_{\mathrm{D},1} = 1\mathrm{s}$ and $\tau_{\mathrm{E},1.1\mathrm{s}}$. For the perturbed variables we choose $[\![\tilde{t}_{\mathrm{D},2}]\!] = 1.1\mathrm{s}$ and $[\![\tilde{t}_{\mathrm{E},2}]\!] = 1\mathrm{s}$. This ensures that the order of the perturbed time stamps is switched. The resulting perturbed timed word is

$$\rho' = (\mathrm{r(E)}, 1\mathrm{s})\ (\mathrm{wd\ r(D,3)}, 1.1\mathrm{s})\ (\mathrm{wd\ r(E,2)}, 6.1\mathrm{s})\ (\mathrm{end}, 6.1\mathrm{s}) \ .$$

When applying ρ' to *TS* we get a transition sequence where car D and car E both simultaneously have two reservations for a duration of 0.1s.

We choose to evaluate ¬safe at $[\![t_{\mathrm{f}}]\!] = 1$s, i.e. after E sets a new reservation and before D withdraws it reservation, and discard all later changes. After 1s car D is at position $[\![pos_{\mathrm{D,f}}]\!] = 34$m and car E is at position $[\![pos_{\mathrm{E,f}}]\!] = 18$m with a braking distance of $[\![\Omega_{\mathrm{E,f}}]\!] = 15$m. We perturb $pos_{\mathrm{D,f}}$ by -1m and $\Omega_{\mathrm{E,f}}$ by $+1$m. The other variables are not perturbed. We get $[\![\widetilde{pos}_{\mathrm{D,f}}]\!] = 33$m and $[\![\widetilde{\Omega}_{\mathrm{E,f}}]\!] = 16$m.

After moving the extension along with E and not perturbing it, we evaluate $\mathrm{tr}_{\mathrm{f}}^{\tilde{\mathrm{d}}}(\Upsilon, \neg\mathrm{safe})$ with $\Upsilon = (\mathbb{I}, 1, 3, \tilde{x}_{\mathrm{l,f}}\tilde{x}_{\mathrm{r,f}}, \nu)$, $[\![\tilde{x}_{\mathrm{l,f}}]\!] = 12$m, $[\![\tilde{x}_{\mathrm{r,f}}]\!] = 102$m on the perturbed variables $\widetilde{vars}_{\mathrm{d,f}}$. The formula ¬safe states that there is a subview where the reservations of two different cars overlap. As (after perturbation) car D and car E both have a reservation on lane 2 and as the position of D (33m) is within the space reserved by E ($[18\mathrm{m}, 18\mathrm{m} + 16\mathrm{m}]$), the formula ¬safe is satisfied. Hence, $\rho(M) \models_{\mathrm{seq}}^{\varepsilon,\delta} \Box\,\mathrm{safe}$ does not hold. △

5 Discussion

Spatio-temporal robustness has been studied before for more abstract formalisms [5, 18]. However, here the data for which we want to achieve robustness has a specific meaning, i.e. the underlying model of MLSL is dedicated to modelling motorway traffic. To this end, we study spatio-temporal robustness, taking the meaning of data into account.

In real-time systems we distinguish between time-driven and event-driven real-time systems [13]. In MLSL we have two kinds of data values: the event-driven values *clm*, *res* and *acc* and the time-driven values *pos*, *spd* and Ω. We study temporal robustness only for the event-driven values *clm* and *res*. For this we use the methodology from timed languages, where time stamps are perturbed [10]. Additionally, we study spatial robustness for the time-driven values *pos* and Ω in a static 'timeless' manner at the level of traffic snapshots. In [7] such a 'timeless' approach to spatial robustness has been done for Metric Temporal Logic.

One of the goals in the definition of MLSL was to reduce complexity of spatial reasoning by separating the spatial aspects from the car dynamics [11]. In this sense, the introduction of temporal robustness by perturbing time stamps seems well suited for MLSL, because we separate temporal robustness from spatial robustness, which simplifies reasoning.

A disadvantage of our approach is that at the linking of time-driven and event-driven values (here *acc* values, as they are event-driven and affect future evolution of time-driven values) we do not achieve temporal robustness, as it affects spatial robustness.

For our approach to temporal robustness we consider similarity of timed words. A common definition to quantify similarity of timed words is defined in [10]. However, there the requirement is made that timed words have an infinite distance if they do not agree on the order of events. In [1] the authors define a quantitative notion of (bi)similarity. However, they define that the *i*th position in one sequence is compared to the *i*th position in another sequence, i.e. they do not consider that the order of events may not always be relevant. Here the timed words originate from a distributed system, which makes it unreasonable to always consider the order of events as relevant. Hence, we define an independence relation in the sense of [16] and in our quantification of similarity allow independent events to change their order. To the best of our knowledge, a quantitative comparison of timed words under consideration of causality has not been used before.

On the side of efficiency we add the following observation: If in the static MLSL formula ϕ all horizontal chop operators are below an odd number of negations, then the arithmetic formulas $\neg\mathrm{tr}_{\Box}(\rho, M, \phi)$ and $\neg\mathrm{tr}_{\Box}^{\varepsilon,\delta}(\rho, M, \phi)$ only contain existentially quantified variables over the reals. If we use an SMT solver

[4] to check satisfiability of the formulas, we can interpret all variables as *uninterpreted constants*, for which the solver tries to find a satisfying assignment. This yields a significant speedup.

6 Conclusion

In this work we define a linear version of a dense-time globally operator for MLSL. While there has been a temporal extension of MLSL [15], it is not suitable for monitoring, as it is a branching time temporal extension. Our first main result is a transformation that takes an MLSL transition sequence $\rho(M)$ and an MLSL formula ϕ to create a formula from the decidable first-order theory of real-closed fields [21], such that the resulting formula is valid iff ϕ holds globally in $\rho(M)$.

We then extend our transformation to accomodate for imprecise spatio-temporal data. For this we defined a causality respecting notion of spatio-temporal similarity, which we base on timed words. Our second main result is a transformation that additionally to the transition sequence $\rho(M)$ and the static MLSL formula ϕ takes a maximal temporal error ε and a maximal spatial error δ, such that the resulting formula is valid iff $\Box\phi$ holds ε-δ-robustly in $\rho(M)$. Again, the resulting formula is from the first-order theory of real-closed fields, and can algorithmically be checked for satisfiability.

Note that, while we consider only uni-directional traffic, our results easily extend to bi-directional traffic. Speed and acceleration, the braking distance and the physical length of a car would then take negative values for cars going in the other direction and need to be updated accordingly when a car starts driving in the other direction.

In this work we define only a linear time globally operator for MLSL. For future work we would like to define a fully fledged temporal extension of MLSL, where temporal operators are basically taken from Metric Temporal Logic [14] and atoms are MLSL formulas. It is desirable to extend our transformation to such an extended temporal version of MLSL.

We stated several claims in this work. However, we did not provide proofs for them. In future work proofs for our claims are certainly desirable.

Another line of research is to create temporal signals for MLSL formulas. Such a temporal signal then represents for every instant in time if the MLSL formula currently is satisfied. Then, we could use the significant work done for monitoring of Metric Temporal Logic [7, 6, 5] and similar logics.

References

[1] Luca de Alfaro, Marco Faella & Mariëlle Stoelinga (2004): *Linear and Branching Metrics for Quantitative Transition Systems*. In Josep Díaz, Juhani Karhumäki, Arto Lepistö & Donald Sannella, editors: *ICALP*, *LNCS* 3142, Springer, pp. 97–109, doi:10.1007/978-3-540-27836-8_11.

[2] Matthias Althoff & John M Dolan (2014): *Online Verification of Automated Road Vehicles Using Reachability Analysis*. *IEEE Transactions on Robotics* 30(4), pp. 903–918, doi:10.1109/TRO.2014.2312453.

[3] Rajeev Alur & David L. Dill (1994): *A Theory of Timed Automata*. *Theor. Comput. Sci.* 126(2), pp. 183–235, doi:10.1016/0304-3975(94)90010-8.

[4] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia & Cesare Tinelli (2009): *Satisfiability Modulo Theories*. In Armin Biere, Marijn Heule, Hans van Maaren & Toby Walsh, editors: *Handbook of Satisfiability*, *Frontiers in Artificial Intelligence and Applications* 185, IOS Press, pp. 825–885, doi:10.3233/978-1-58603-929-5-825.

[5] Alexandre Donzé & Oded Maler (2010): *Robust Satisfaction of Temporal Logic over Real-Valued Signals*. In Krishnendu Chatterjee & Thomas A. Henzinger, editors: *FORMATS*, *LNCS* 6246, Springer, pp. 92–106, doi:10.1007/978-3-642-15297-9_9.

[6] Georgios E. Fainekos & George J. Pappas (2006): *Robustness of Temporal Logic Specifications*. In Klaus Havelund, Manuel Núñez, Grigore Roşu & Burkhart Wolff, editors: *FATES*, LNCS, Springer, pp. 178–192, doi:10.1007/11940197_12.

[7] Georgios E. Fainekos & George J. Pappas (2009): *Robustness of Temporal Logic Specifications for Continuous-Time Signals*. *Theor. Comput. Sci.* 410(42), pp. 4262–4291, doi:10.1016/j.tcs.2009.06.021.

[8] Martin Fränzle & Michael R. Hansen (2005): *A Robust Interpretation of Duration Calculus*. In Dang Van Hung & Martin Wirsing, editors: *ICTAC*, *LNCS* 3722, Springer, pp. 257–271, doi:10.1007/11560647_17.

[9] Martin Fränzle, Michael R Hansen & Heinrich Ody (2015): *No Need Knowing Numerous Neighbours - Towards a Realizable Interpretation of MLSL*. In Roland Meyer, André Platzer & Heike Wehrheim, editors: *Correct System Design*, *LNCS* 9360, Springer, pp. 152–171, doi:10.1007/978-3-319-23506-6_11.

[10] Vineet Gupta, Thomas A. Henzinger & Radha Jagadeesan (1997): *Robust Timed Automata*. In Oded Maler, editor: *Hybrid and Real-Time Systems*, *Lecture Notes in Computer Science* 1201, Springer, pp. 331–345, doi:10.1007/BFb0014736.

[11] M. Hilscher, S. Linker, E.-R. Olderog & A. P. Ravn (2011): *An Abstract Model for Proving Safety of Multi-Lane Traffic Manoeuvres*. In Shengchao Qin & Zongyan Qiu, editors: *ICFEM*, *LNCS* 6991, Springer, pp. 404–419, doi:10.1007/978-3-642-24559-6_28.

[12] Martin Hilscher & Maike Schwammberger (2016): *An Abstract Model for Proving Safety of Autonomous Urban Traffic*. In Augusto Sampaio & Farn Wang, editors: *ICTAC*, *LNCS* 9965, pp. 274–292, doi:10.1007/978-3-319-46750-4_16.

[13] Hermann Kopetz (1991): *Event-Triggered Versus Time-Triggered Real-Time Systems*. In Arthur I. Karshmer & Jürgen Nehmer, editors: *Operating Systems of the 90s and Beyond*, *LNCS* 563, Springer, pp. 87–101, doi:10.1007/BFb0024530.

[14] Ron Koymans (1990): *Specifying Real-Time Properties with Metric Temporal Logic*. *Real-Time Systems* 2(4), pp. 255–299, doi:10.1007/BF01995674.

[15] Sven Linker & Martin Hilscher (2015): *Proof Theory of a Multi-Lane Spatial Logic*. *Logical Methods in Computer Science* 11(3), doi:10.2168/LMCS-11(3:4)2015.

[16] Antoni W. Mazurkiewicz (1986): *Trace Theory*. In Wilfried Brauer, Wolfgang Reisig & Grzegorz Rozenberg, editors: *Advances in Petri Nets*, *LNCS* 255, Springer, pp. 279–324, doi:10.1007/3-540-17906-2_30.

[17] Heinrich Ody (2015): *Undecidability Results for Multi-Lane Spatial Logic*. In Martin Leucker, Camilo Rueda & Frank D. Valencia, editors: *ICTAC*, *LNCS* 9399, Springer, pp. 404–421, doi:10.1007/978-3-319-25150-9_24.

[18] Jan-David Quesel (2013): *Similarity, Logic, and Games - Bridging Modeling Layers of Hybrid Systems*. Ph.D. thesis, University of Oldenburg.

[19] Albert Rizaldi & Matthias Althoff (2015): *Formalising Traffic Rules for Accountability of Autonomous Vehicles*. In: *ITSC*, IEEE, pp. 1658–1665, doi:10.1109/ITSC.2015.269.

[20] Maike Schwammberger (2017): *Imperfect Knowledge in Autonomous Urban Traffic Manoeuvres*. FVAV.

[21] Alfred Tarski (1951): *A Decision Method for Elementary Algebra and Geometry*. University of California Press.