Constraint Systems from Traffic Scenarios for the Validation of Autonomous Driving*

Andreas Eggers, Matthias Stasch, Tino Teige,
Tom Bienmüller, and Udo Brockmeyer

BTC Embedded Systems AG, Gerhard-Stalling-Straße 19, 26135 Oldenburg, Germany
{eggers, stasch, teige, bienmueller, brockmeyer}@btc-es.de
http://www.btc-es.de

Abstract. One does not need the gift of clairvoyance to predict that in the near future autonomously driving cars will occupy a significant place in our everyday life. In fact, in designated and even public test-drive areas it is reality even now. Autonomous driving comes with the ambition to make road traffic safer, more efficient, more economic, and more comfortable – and thus to "make the world a bit better". Recent accidents with automatic cars resulting in severe injuries and death, however, spark a debate on the safety validation of autonomous driving in general. The biggest challenge for autonomous driving to become a reality is thus most likely not the actual development of intelligent vehicles themselves but their rigorous validation that would justify the necessary level of confidence. It is common sense that classical test approaches are by far not feasible in this emerging area of autonomous driving as these would induce billions of kilometers of real-world driving in each release cycle. To cope with the tremendous complexity of traffic situations that a self-driving vehicle must deal with – without doing any harm to any other traffic participants – a promising approach to safety validation is virtual simulation, i.e. driving the huge amount of test kilometers in a virtual but realistic simulation environment. A particular interest here is in the validation of the behavior of the autonomous car in rather critical traffic scenarios.

In this position paper, we concentrate on one important aspect in virtual safety validation of autonomous driving, namely on the specification and concretization of critical traffic scenarios. This aspect gives rise to complex and challenging constraint systems to be solved, to which we believe the SC^2 community may give essential contributions by their rich variety of diverse methods and techniques.

Keywords: Autonomous Driving, Traffic Scenarios, Constraint Systems, Constraint Solving

* This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL joint undertaking under grant agreement no 692455. This joint undertaking receives support from the European Union's Horizon 2020 Research and Innovation Programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

1 Introduction

While driver assistance systems that are in use today already have to be subjected to rigorous validation and verification efforts, there always is a human driver who can serve as a fallback when the system goes into a "fail safe" state. Latest when it comes to "mind-off" SAE level 4 automated driving [?], however, human drivers can no longer be expected to instantaneously and in-time take over control when a malfunction of the autonomous driving service occurs. As a conclusion, humans can not serve as fallback insurance in that case, meaning that with this degree of automation the level of required confidence in the controlling machine enormously increases. ISO 26262 [?] defines the term "safety validation" for safety goals on the vehicle level based on tests and examination. Applying traditional, today's state-of-the-art validation methods for this safety validation ends up in billions of kilometers to autonomously test-drive to reach the needed confidence, which obviously turns out to be impractical, in particular, when observing that any change in the algorithm requires to iterate such test-drives [?]. Note that this "billions of kilometers to drive" observation is based on today's statistics about injuries and fatalities, taking into account human drivers. On the one hand, a single human driver can act differently in very similar traffic situations – also depending on physical conditions, emotions, and abilities. On the other hand, humans will be able to do analogous decisions based on comparable situations – also relying on acquired knowledge and intelligence to transfer from experience. A digital machine does neither have emotions nor is it able to really transfer decisions from previous experience – validating safety for automated driving requires a paradigm change for safety goal validation.

Validating safety goals for automated driving requires two major ingredients. First, virtualization will be needed to shift safety goal validation from physical real world driving to a virtual world [?]. Second, traffic situations need to be (graphically) captured and expressed in a semantically unique, systematic, and comprehensive way [?,?,?,?,?]. Fundamental to virtual validation is the way how traffic scenarios are specified. On the one hand, a *huge mass* of scenarios will be required to show safe behavior of the automated driver. Here, abstract scenarios capturing the general scene such as an overtaking maneuver need to be varied, for example, under weather or road conditions. On the other hand, when looking at today's virtual simulation engines such as VIRES VTD[1] or IPG CarMaker[2], modeling reality as specifically as possible requires lots of details such as trees, buildings, road surfaces, etc. in order to end up in a very *concrete* scenario, which can serve as an input to a simulator.

We believe that dealing with the huge amount of scenarios and details needed for simulation pragmatically requires to provide an abstract specification method for base scenarios only capturing the essential core of a driving maneuver, and to equip this with automatic concrete scenario generation out of these abstract

[1] `https://vires.com/vtd-vires-virtual-test-drive/`

[2] `https://ipg-automotive.com/products-services/simulation-software/carmaker/`

descriptions. This paper aims at showing the fundamental experimental concepts of reducing this automatic generation to a constraint solving problem, while still abstracting from details like road surfaces or weather conditions.

Structure of the paper. We first sketch ideas on a graphical specification language for abstract traffic scenarios in Section 2.1 by means of an example. Section 2.2 describes the constraints of the example and a simple physical model of the traffic participants including the ego vehicle which allows modeling the abstract scenario on a straight road. A brief outline of how more complex road geometries could be handled is given in Section 2.3. In Section 2.4, we discuss the uncontrollability of the ego vehicle and its impact on the simulation of a scenario. Intuitively, as the ego vehicle's behavior cannot (and must not) be controlled by the test environment but the intended scenario is still to be simulated as expected, we view the other traffic participants as run by to-be-synthesized controllers whose aim is to interact with the ego vehicle in a reactive way such that the expected scenario is actually performed as accurately as possible. In Section 3, we give some further examples of abstract traffic scenarios which may serve as benchmarks for the SC^2 community, while Section 4 concludes the paper.

2 Problem Statement

As motivated in Section 1, one major challenge in the validation of autonomous driving functionality is to synthesize concrete critical traffic situations from a rather abstract, intuitive, graphical, and declarative description of a traffic scenario. In this section, we approach the problem by means of a concrete example, namely an *overtaking maneuver on a highway* for a truck as ego vehicle[3]. We first explain the abstract description of the example scenario and then derive the necessary constraint system to be solved in order to concretize the abstract scenario, i.e. to synthesize a concrete instance of the allowed traffic behavior matching the constraints of the original scenario.

The full problem class would involve challenging ingredients from several scientific disciplines like *continuous* behavior to enforce the driving physics of vehicles, *discrete choices* to start lane changes or braking actions, *geometric* shapes to specify curves by clothoids, or *material science* to describe the effect of the road surface on driving vehicles. Ultimately, solving the problem not only calls for finding value sequences for system inputs like acceleration or steering

[3] The *ego vehicle* in the terminology of autonomous driving is the vehicle that is controlled by the autonomous driving system and in the context of its validation is thus the system under test. To avoid some possible confusion, we shall point out that during an actual test of the ego vehicle, it has the freedom to control its own behavior, i.e. it is uncontrollable from the perspective of the test environment. For the scenario concretization problem which we are about to introduce in this section, however, we first ignore this limitation and assume that a concretization may in fact prescribe a behavior of all vehicles including the ego vehicle to concretize the scenario. Later in Sect. 2.4, we revisit this important topic.

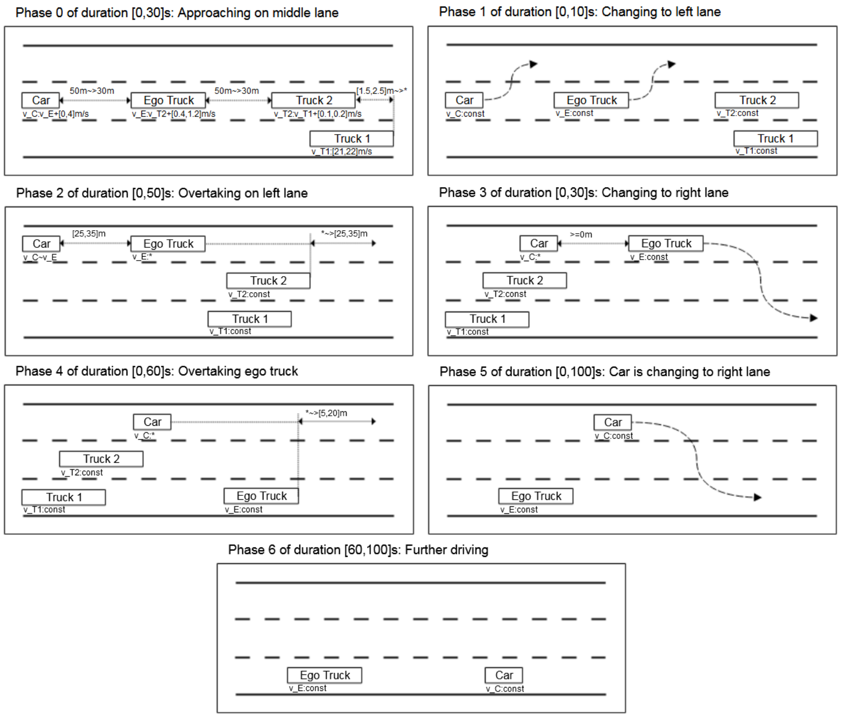

Fig. 1. Graphical description of the traffic scenario *overtaking maneuver on a highway.* (Note: we use as a convention traffic moving from left to right, i.e. on the bottom is the rightmost, slowest lane.)

movement of the vehicles, but aims for synthesizing a reactive controller of the surrounding traffic, which is *controllable*, that obeys the abstract traffic scenario under each behavior of the *uncontrollable* ego vehicle.

2.1 Example: Overtaking Maneuver on Highway

To approach the problem statement, we start by explaining the example of an *overtaking maneuver on a highway.* The graphical abstract scenario description is shown in Fig. 1. We first explain the meaning of the traffic scenario phase by phase.

Phase 0. Initially there are four vehicles on a three-lane road while these vehicles keep their corresponding lanes throughout this phase. `Truck 1` is driving on the rightmost lane with a velocity ranging between 21 and 22 m/s. `Truck 2` is driving in the middle lane with a velocity that is a bit greater than this of `Truck 1`, namely by 0.1 to 0.2 m/s. The front distance between `Truck 1` and `Truck 2` lies in the interval $[1.5\,\mathrm{m}, 2.5\,\mathrm{m}]$ initially and can then evolve arbitrarily within this phase. We remark that the distance will decrease, of course, due to the constraint on the velocities. However, this knowledge, in particular *how* the distance will evolve, is not specified but is implied by physics and potentially other dependent constraints, and its concrete evolution is left to solving the concretization problem. The `Ego Truck` shall drive a bit faster than `Truck 2` (also in the middle lane), thus closing the distance to `Truck 2` from 50 m initially to 30 m at the end of the phase. A similar behavior is specified for the vehicle `Car` (in the middle lane, too). Each traffic phase also consists of a global timing constraint on the allowed duration. The duration of phase 0 must be at least 0 s and at most 30 s.

Phase 1. In the beginning of this phase all four vehicles reside in their corrsponding lanes. This is required to ensure continuity between the phases, meaning that there cannot be unspecified behavior between two consecutive traffic phases. `Truck 1` is staying on the right lane throughout the phase and keeps a constant velocity. As a design decision for this scenario, the exact velocity is not specified, but instead it is constrained to be the final velocity of `Truck 1` at the end of phase 0. A similar behavior is defined for `Truck 2` which stays on the middle lane with a constant velocity. The `Ego Truck` and `Car` are also specified to use a constant velocity but they have to change from the middle lane to the left lane during the phase. This lane change must be completed within 10 s from the start of the phase as this is defined as the upper limit for the duration of the phase.

Phase 2. In this phase `Truck 1` and `Truck 2` are to continue their behavior from phase 1 in staying in their respective lanes at a constant velocity. The `Car` is driving on the left lane for the entirety of this phase while keeping a similar velocity as the `Ego Truck` denoted by $v\_C \sim v\_E$ and staying at a distance between 25 m and 35 m behind the `Ego Truck`. The `Ego Truck` is staying on the left lane during this phase and is not directly constrained in its velocity. The velocity of the `Ego Truck` is governed by the duration of the phase given by the interval $[0\,\text{s}, 50\,\text{s}]$ and the distance constraint which requires the `Ego Truck` to be between 25 m and 35 m in front of `Truck 2` at the end of the phase.

Phase 3. During this phase the `Ego Truck` changes from the left lane to the right lane while keeping a constant velocity. The exact velocity is not given and is therefore defined by the final velocity at the end of phase 2. The lane change must be completed within the phase duration of $[0\,\text{s}, 30\,\text{s}]$. The `Car` is staying on the left lane and has to keep a distance which is greater than or equal to 0 m to the `Ego Truck` while the velocity is unconstrained. `Truck 1` and `Truck 2` are both keeping their respective lanes and driving with constant velocities defined by the final velocities of phase 2.

Phase 4. In this phase the `Car` keeps staying on the left lane and overtakes the `Ego Truck`. The velocity of `Car` is undefined but it has to drive between 5 m and 20 m in front of the `Ego Truck` at the end of the phase. The duration of the phase is given by the interval $[0\,\text{s}, 60\,\text{s}]$. The `Ego Truck`, `Truck 1`, and `Truck 2` all keep their lanes and drive at a constant velocity.

Phase 5. In this phase, `Truck 1` and `Truck 2` are not present anymore as they are expected to be too far away having no further impact on the `Ego Truck`. The `Ego Truck` keeps driving on the right lane at constant velocity. The `Car` is changing from the left lane to the right lane while keeping a constant velocity. This lane change has do be finished within 100 s.

Phase 6. This is the final phase of the scenario which has a duration of at least 60 s and at most 100 s in which the `Ego Truck` and `Car` both keep driving on the right lane at a constant velocity.

2.2 Constraints of the Problem Statement

On the most abstract level, the scenario concretization problem consists of a sequence of phases $p_1, \dots, p_n$. Within each phase, constraints describe the initial state of the phase, invariants regarding its continuous evolution over time, and a final state which must overlap with the initial state of the next phase (if any). Let $\boldsymbol{x}$ be the vector of vehicle position (x, y), orientation (γ), velocity (v), angular velocity (ω), and acceleration (a) variables and their derivatives with respect to time for all vehicles involved in the scenario. Furthermore, let $\psi_t(\boldsymbol{x})$ denote the valuation of these variables at a time t. Then a *discrete scenario concretization* is a sequence of valuations $\psi_{t_0}(\boldsymbol{x}), \dots, \psi_{t_m}(\boldsymbol{x})$ such that there exist monotonically increasing time stamps $\tau_0, \dots, \tau_{n+1} \in \{t_0, \dots, t_m\}$ with $\tau_0 = t_0$ and $\tau_{n+1} = t_m$ such that

$$
\begin{aligned}
& \psi_{\tau_0}(\boldsymbol{x}) \models \mathit{init}_{p_0}(\boldsymbol{x}) \\
\wedge\ & \forall t \in [\tau_0, \tau_1] : \psi_t(\boldsymbol{x}) \models \mathit{invar}_{p_0}(\boldsymbol{x}) \\
\wedge\ & \mathit{min\_duration}_{p_0} \leq \tau_1 - \tau_0 \leq \mathit{max\_duration}_{p_0} \\
\wedge\ & \psi_{\tau_1}(\boldsymbol{x}) \models \mathit{final}_{p_0}(\boldsymbol{x}) \\
\wedge\ & \psi_{\tau_1}(\boldsymbol{x}) \models \mathit{init}_{p_1}(\boldsymbol{x}) \\
\wedge\ & \forall t \in [\tau_1, \tau_2] : \psi_t(\boldsymbol{x}) \models \mathit{invar}_{p_1}(\boldsymbol{x}) \\
\wedge\ & \mathit{min\_duration}_{p_1} \leq \tau_2 - \tau_1 \leq \mathit{max\_duration}_{p_1} \\
\wedge\ & \psi_{\tau_2}(\boldsymbol{x}) \models \mathit{final}_{p_1}(\boldsymbol{x}) \\
\wedge\ & \psi_{\tau_2}(\boldsymbol{x}) \models \mathit{init}_{p_2}(\boldsymbol{x}) \\
& \vdots \\
\wedge\ & \psi_{\tau_n}(\boldsymbol{x}) \models \mathit{final}_{p_{n-1}}(\boldsymbol{x}) \\
\wedge\ & \psi_{\tau_n}(\boldsymbol{x}) \models \mathit{init}_{p_n}(\boldsymbol{x}) \\
\wedge\ & \forall t \in [\tau_n, \tau_{n+1}] : \psi_t(\boldsymbol{x}) \models \mathit{invar}_{p_n}(\boldsymbol{x}) \\
\wedge\ & \mathit{min\_duration}_{p_n} \leq \tau_n - \tau_{n-1} \leq \mathit{max\_duration}_{p_n} \\
\wedge\ & \psi_{\tau_{n+1}}(\boldsymbol{x}) \models \mathit{final}_{p_n}(\boldsymbol{x}) \\
\wedge\ & \forall t \in [t_0, t_m] : \mathcal{C}(\psi_t(\boldsymbol{x})),
\end{aligned}
$$

i.e. each time point τ_i selects a valuation that allows at least one transition from one phase to the next, between such time points, the invariant and duration constraints of the then-current phase are satisfied, and there are finitely many (possibly none) intermediate valuations within each phase (the t_i without corresponding τ_j), at which the stimuli $(a,\, \omega)$ – which determine the vehicles' behaviors – can change, while all valuations can be connected by continuous evolutions of the physical system dynamics ($\mathcal{C}$ as detailed further below). The most precise interpretation of the continuous behavior and invariant constraints on them is a continuous-time semantics, i.e. $\forall t \in [\tau_i, \tau_{i+1}]$ really means for all points of time that exist in $[\tau_i, \tau_{i+1}] \subseteq \mathbb{R}$. Less strict interpretations can however be employed to reach approximations of this semantics, e.g. by considering only $t \in \{t_0, \ldots, t_m\}$ with $\tau_i \leq t \leq \tau_{i+1}$. For the purpose of the semantics description, we will take a continuous-time perspective, but for practical purposes, such discrete approximations may very well be suitable, too (also see Sect. 2.4). The intention of the final constraint $\forall t \in [t_0, t_m] : \mathcal{C}(\psi_t(\boldsymbol{x}))$ is to ensure that a solution $\psi_{t_0}(\boldsymbol{x}), \ldots, \psi_{t_m}(\boldsymbol{x})$ of the discrete scenario concretization problem can be understood as an excerpt of a continuous solution, i.e. a function that satisfies $\mathcal{C}$ over continuous time, keeps the stimuli constant except at times $t_0, \ldots, t_m$, and switches from one phase to the next at time points $\tau_1, \ldots, \tau_n$.

Intuitively, such a sequence of valuations describes stimuli (accelerations and angular velocities[4]) and the states reached when applying them such that the phases of the scenario are traversed. Between the valuations, no change to the stimuli is made, i.e. finding a discrete scenario concretization directly leads to showing that a finite number of stimuli changes suffices to perform the scenario. We expect that this can often be achieved with relatively low numbers of intermediate steps for realistic scenarios. However, in general, when not imposing any limit on the number of intermediate steps within each phase, this allows at least theoretically an arbitrarily fine resolution of stimuli changes.[5]

The ingredients of the above formula need to be described in more detail. Focusing on phase 0 of the example (see Fig. 2), we can identify the following types of constraints:

– *Distance evolution constraints:* the distance marker between the vehicles `Car` (with $x_{\texttt{Car}}$ as its longitudinal position) and `Ego Truck` (analogously $x_{\texttt{Ego Truck}}$) has been graphically annotated with a constraint "$50\,\mathrm{m} \rightsquigarrow 30\,\mathrm{m}$". This notation actually amounts to three specifications which contribute to (1) the initial predicate of the phase init_{p_0}, i.e. when entering the phase, the initial

[4] Alternatively, the use of an angular acceleration as stimulus and the angular velocity as resulting physical variable are of course possible, too.

[5] Note that the formula structure can be matched to the bounded model checking formulas used to analyze among others also discrete-continuous-hybrid systems, of which this problem is one special case [?]. Starting e.g. from the assumption of at most one valuation per phase transition, incremental solving could amount to increasing the number of intermediate valuations, effectively looking for simple solutions before looking for more complex ones.

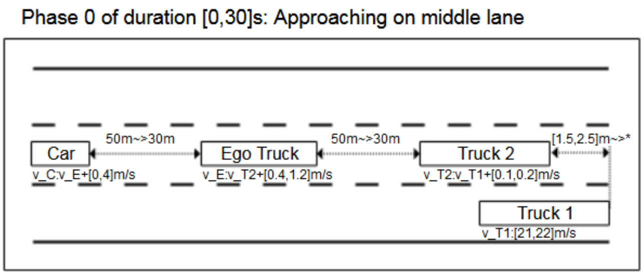

Fig. 2. Phase 0 of the traffic scenario *overtaking maneuver on a highway.*

state $\psi_{\tau_0}(\boldsymbol{x})$ of the phase must satisfy

$$x_{\texttt{Ego Truck}} - x_{\texttt{Car}} = 50\,\text{m},$$

(2) $invar_{p_0}$, such that all states traversed while passing through the phase must stay within the interval, i.e.

$$30\,\text{m} \leq x_{\texttt{Ego Truck}} - x_{\texttt{Car}} \leq 50\,\text{m},$$

and (3) $final_{p_0}$, i.e. the phase can only be left when

$$x_{\texttt{Ego Truck}} - x_{\texttt{Car}} = 30\,\text{m}.$$

- *Velocity constraints:* the vehicle `Truck 1` is annotated with the constraint "$v_{\texttt{Truck 1}} : [21, 22]\text{m/s}$". This constraint describes the velocity of the vehicle during the entire phase, i.e. it contributes the constraint $v_{\texttt{Truck 1}} \in [21, 22]\text{m/s}$ to $init_{p_0}$, $invar_{p_0}$, and $final_{p_0}$.
- *Velocity difference constraints:* the `Ego Truck` is annotated with "$v_{\texttt{Ego Truck}} : v_{\texttt{Truck 2}} + [0.4, 1.2]\text{m/s}$", which constrains the difference of velocities between the `Ego Truck` and the truck in front of it. Again, this constraint directly contributes to $init_{p_0}$, $invar_{p_0}$, and $final_{p_0}$ as $v_{\texttt{Ego Truck}} \in v_{\texttt{Truck 2}} + [0.4, 1.2]\text{m/s}$, i.e. must hold when entering, during, and when leaving phase 0.
- *Lane position constraints:* each vehicle has been placed graphically onto one of the lanes, e.g. `Truck 1` drives on the rightmost lane, which amounts to a constraint $y_{\texttt{Truck 1}} \in [mid\_lane\_right + 0.5 \cdot width_{\texttt{Truck 1}}, mid\_lane\_right - 0.5 \cdot width_{\texttt{Truck 1}}]$ where y is the lateral position of the vehicle, $width$ the vehicle's width, and mid_lane_right the midpoint of the rightmost lane. Also this constraint contributes directly to $init_{p_0}$, $invar_{p_0}$, and $final_{p_0}$. Analogous constraints are added for the other vehicles based on their widths and lane midpoint constants.
- *Phase duration constraints:* on top of the phase graph, a duration constraint is given, such that $min\_duration_{p_0} = 0\,\text{s}$ and $max\_duration_{p_0} = 30\,\text{s}$.

Looking at the remaining phases, we additionally get the following types of constraints:

- *Constant velocity constraints:* in phase 1, vehicle `Car` is annotated with "$v_{\texttt{Car}}$: const", which contributes as a constraint $\dot{v} = 0\,\mathrm{m/s^2}$, i.e. the time-derivative of v is zero and hence the velocity does not change.
- *Lane position evolution constraints:* in phase 1, e.g. vehicle `Car` changes from the middle to the left lane. While this could be annotated with a change rate, this has not been done here explicitly, so that only (1) init_{p_1} is supplied with a constraint

$$y_{\texttt{Car}} \in [\mathit{mid\_lane\_mid} + 0.5 \cdot \mathit{width}_{\texttt{Car}}, \mathit{mid\_lane\_mid} - 0.5 \cdot \mathit{width}_{\texttt{Car}}],$$

(2) invar_{p_1} is constrained by

$$y_{\texttt{Car}} \in [\mathit{mid\_lane\_mid} + 0.5 \cdot \mathit{width}_{\texttt{Car}}, \mathit{mid\_lane\_left} - 0.5 \cdot \mathit{width}_{\texttt{Car}}],$$

and (3) final_{p_1} contains the resulting state of the lane change – the leftmost lane has been reached – i.e.

$$y_{\texttt{Car}} \in [\mathit{mid\_lane\_left} + 0.5 \cdot \mathit{width}_{\texttt{Car}}, \mathit{mid\_lane\_left} - 0.5 \cdot \mathit{width}_{\texttt{Car}}].$$

Note, however, that the presence of a phase duration constraint of $[0\,\mathrm{s}, 10\,\mathrm{s}]$ implicitly defines a minimum rate such that the lane change takes no longer than 10 s.
- *Approximate velocity constraints:* in phase 2, the velocity of the vehicle `Car` is constrained to $v_{\texttt{Car}} \sim v_{\texttt{Ego Truck}}$. This constraint is only an abbreviated notation for a velocity difference constraint with an implicit ε-tolerance that is chosen small enough for the velocities to be similar, e.g. $\varepsilon = 0.4\,\mathrm{m/s}$.

More systematically, the constraints can be categorized in the following way. *Simple constraints* provide non-changing bounds within which the value of the variable may fluctuate arbitrarily. *Evolution constraints* use the initial and final states of a phase to provide an expectation about the development of the states during the time in which the phase is active, potentially even quite precisely with a change rate, including as a special case a change rate of 0 to constrain a value not to change. In Section 3, we will provide examples of more complex evolution constraints. Orthogonally, constraints may refer to a *single variable*, e.g. the velocity of one vehicle, or to *differences of variables*, e.g. the distance as a difference in positions, or the difference of velocities.

To model the physical behavior of the vehicles, $\mathcal{C}(\boldsymbol{x})$ contains the following relationships between the variables and their derivatives:

$$\begin{aligned} \dot{\gamma} &= \omega \\ \dot{v} &= a \\ \dot{x} &= \cos(\gamma) \cdot v \\ \dot{y} &= \sin(\gamma) \cdot v \end{aligned}$$

for each vehicle.

2.3 Road Geometries

Using only the constraints presented so far, a scenario concretization can be interpreted in the context of a sufficiently long straight road. The position of curves or even the height profile of a road, however, is not negligible when thinking about their impact on the criticality of a scenario. When intending to use scenarios to simulate and test autonomous driving functions, all kinds of road geometries may thus have to be used to ensure that e.g. the overtaking scenario can be carried out safely even if curves or slopes make it harder to detect all relevant surrounding traffic.

Regarding the constraint system that needs to be solved, some of these aspects can be ignored under the assumption that their impact can be compensated for by changing the stimuli. If e.g. the height of the road increases gently, the vehicle's acceleration can be adjusted by just a bit such that its velocity reaches the exact same values as it would on a flat surface (assuming that the simulation takes this slow-down effect caused by gravity into account). Similarly, the longer outside lanes in curves can be compensated for by driving a bit faster. A scenario found for a straight road can thus be adapted to more complex road geometries by adapting the stimuli. The price, however, is that the resulting stimuli may be considered too artificial to serve as test cases – since they accelerate e.g. at the beginning of a curve and brake at its end just to follow an "ideal" trajectory computed for a straight road – or even leave the envelope allowed by the constraints (e.g. a maximum velocity needs to be exceeded). If the phases of the scenario request two vehicles to be side by side at the end of a curve, the road-geometry-aware solution could instead e.g. simply give the vehicle on the outer lane a head start, which is consumed during the curve without any additional changes to the stimuli.

When aiming at such geometry-aware solutions, the constraint system needs to be augmented by constraints describing the direction of straight road segments, the radii of arc segments, the curvature change rates of clothoid segments that connect straight and arc segments, or the parameters of polynomial road segments, which form less regular connections. One possible formalization can be based on a function $r : l \mapsto \beta$ that yields a road orientation β for any given length l. For example for a straight road segment s with direction β_s, which starts at road length s_{start} and ends at road length s_{end}, this function would constantly return the same angle: $r_s(l) = \beta_s$ for all $l \in [s_{start}, s_{end}]$. For an arc segment, on the other hand, the returned angle would change with a constant rate that depends on the radius. For clothoid and polynomial segments, even that rate of orientation change itself changes over the length of the segment. The purpose of this piecewise-defined function is then to provide an expected direction for the vehicles such that they drive in the direction of the road. A constraint that shall keep a vehicle on its lane must enforce that the direction of the vehicle coincides with the road's direction, while during a lane-change this requirement must be relaxed such that the orientation can deviate from the road's direction sufficiently to reach the other lane. For distance constraints, the semantics must be clarified, by either using a road-length interpretation (i.e. when in a curve, the distance along the curve is taken) versus the Euclidean distance. A major technical challenge in the formalization may additionally be the lack of closed-form solutions for the points on clothoid road segments.

Since we believe that solving the scenario concretization problem on the straight-road geometry is already a very challenging task by itself and a reasonable intermediate step in the direction of solving this more complex problem, we have not yet attempted to fully formalize the road-geometry-aware case.

2.4 Variants of the Problem Statement

Looking at the validation of autonomous driving functions via simulated scenarios, the discrete scenario concretization has one significant drawback: it contains stimuli for all vehicles *including* the ego vehicle, which must not be controlled by the test environment since it is the system under test. While such solutions prove the existence of concretizations and thus the plausibility of the scenario (not an unimportant result at all), an actual test run will contain controllable vehicles – whose stimuli must be chosen such that the scenario is performed successfully – and the uncontrollable ego vehicle – which choses its stimuli by itself.

To solve this issue, the constraint system presented above must be understood more like a controller synthesis problem rather than a pure constraint solving problem. The solution would thus have a reactive element, e.g. in the form of an executable function, such that in each simulation step the state of and stimuli chosen by the ego vehicle can be used as an input and the stimuli of the other vehicles can then be computed by the function as an output such that these stimuli lead to a discrete scenario concretization in the sense presented above. Obviously, an ego vehicle can undermine the attempt to finish a scenario, e.g. by changing to the wrong lane, driving too slowly or too fast, etc. An *optimal* solution would thus be a reactive controller that concretizes the solution for all behaviors of the ego vehicle for which there still exists a possible completion.

The above problem statement can thus be considered a challenge with multiple degrees of solution:

- *Controller synthesis:* Find a controller that stimulates the surrounding traffic in such a way that the resulting sequence forms a discrete scenario concretization, as long as the ego vehicle's behavior allows for the existence of such a concretization.
- *Plausibility proof:* Find one discrete scenario concretization with stimuli for all vehicles including the ego vehicle. Though not directly simulatable, this result proves that the scenario can be realized under the assumption that the ego vehicle cooperates, i.e. the scenario specification as such is plausible and does not contradict itself (e.g. adjacent phases whose constraints allow no valid transitions).
    - *... on road geometry:* using road geometry constraints such that the vehicles follow a potentially complex road geometry during the scenario simulation.

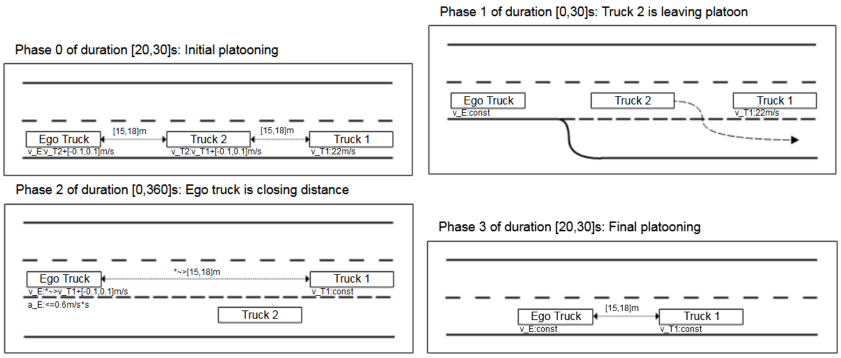

Fig. 3. Graphical description of the traffic scenario *truck platooning.*

- *…on a straight road:* not taking any road geometry constraints into account, i.e. driving just on a straight road and thus showing that the scenario is plausible in theory, but not necessarily on all concrete road geometries (e.g. because velocities would need to be higher than allowed by the constraints when driving on the longer outer lanes of a curve).

Orthogonally to these degrees of solutions, different levels of abstraction can be chosen. An approximative solution might e.g. be more easily obtained when ignoring phase invariant constraints between stimuli changes and limiting oneself to just checking a posteriori whether the invariant constraints hold during each simulation step. Using simulations with sufficiently small sample times, one will normally be able to detect invariant violations with practically sufficient accuracy during simulation. If violations are observed in this way, the concretization of the scenario can be repeated with a finer resolution for the intermediate steps, i.e. using more intermediate steps (potentially evenly distributed between the steps of the concretization in which the violation occurred) and thus reducing the risk of overlooking an invariant violation. Similarly, the continuous dynamics can be approximated, especially when using a reactive component that uses control strategies to achieve satisfaction of the constraints. In this case, the behavior that is provided by the simulation environment and the own approximation can be compared in each step and stimuli be adapted in such a way that small deviations caused by the approximation are corrected. Additionally, tolerances can be used to allow slight deviations from the "exact" constraints – e.g. a velocity is barely ever truly constant in the real world, so a small environment around the constraint may be considered a rather natural relaxation of the problem.

3 Further Examples of Abstract Traffic Scenarios

In this section, we give three further graphical descriptions of abstract traffic scenarios which may serve as benchmarks for the SC^2 community.

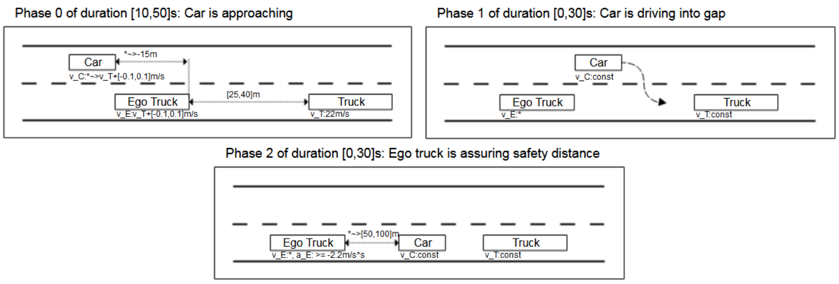

Fig. 4. Graphical description of the traffic scenario *dangerous lane change*.

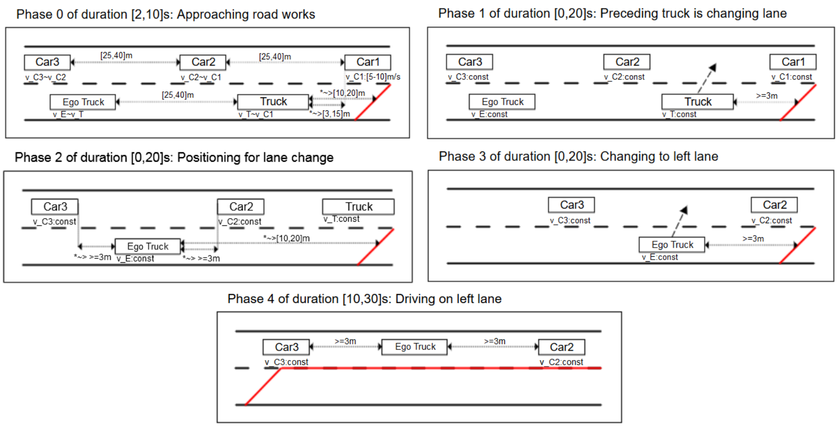

Fig. 5. Graphical description of the traffic scenario *late merge*.

Truck Platooning. The critical aspect of the *truck platooning* example, as shown in Fig. 3, is the very short distance between the trucks of the platoon, which is essential however, since such platoons of autonomous and cooperating trucks aim at reducing fuel consumption and optimizing utilization of roads.

We remark that phase 2 introduces a more complex *velocity evolution constraint*: in addition to the initial and final velocity constraints, the change rate of the velocity, i.e. the accelaration a_E, is bounded from above by $0.6\,{}^{m}/{}_{s^2}$, which is important in order to specify fuel-efficient driving.

Dangerous Lane Change. A *dangerous lane change* of a car from the left to the right lane into a small gap between two trucks is illustrated in Fig. 4. The task of the ego truck is to assure a safe distance to the car driving ahead while avoiding abrupt braking.

Late Merge. The abstract traffic scenario of a *late merge* in Fig. 5 describes the critical situation in dense traffic when two lanes are merged into one.

4 Conclusion and Future Work

The development and, in particular, the safety validation of autonomous vehicles is one of the biggest challenges in the near future. In this position paper, we focussed on a central aspect of the validation of autonomous driving, namely on the specification and concretization of abstract traffic scenarios which will establish the fundamental basis for safety argumentation. We sketched ideas on a graphical specification language for abstract traffic scenarios. By means of a concrete example we derived the constraint problem class to be solved in order to synthesize concrete traffic scenarios. We also outlined an extension of the problem statement by adding road geometry constraints and by asking for controller synthesis in addition to plausibility check of abstract traffic scenarios. In total, we provided four concrete examples of abstract scenarios which may serve as benchmarks for the SC^2 community.

With regard to potential approaches to solving such constraint systems arising from traffic scenarios, one has to cope with several dimensions of complexity. The problem class incorporates system evolution over *real time* while the behavior of vehicles is represented by *differential equations* involving *transcendental functions* like cos and sin. This indicates to employ constraint solvers being able to handle differential equations, which in turn creates doubts on applicability in industrial praxis. When trying to simplify the problem, it is not obvious whether these differential equations have closed-form solutions and, if not, whether good-enough approximations exist that still yield valid concrete scenarios. In addition to the intricacy of the constraints, it might be the case that the controller synthesis problem calls for *nested quantifiers* and thus for quantifier elimination techniques. A potential approach might also be along the lines of combining numerical optimization and SAT solving [**?**]. Another important question is whether *decomposition* of the abstract traffic scenario, e.g. phase-wise or even more fine-grained within a phase, or *abstraction-refinement* approaches might be beneficial when solving the scenario concretization problem.

Solving this problem class arising from traffic scenarios thus is a major challenge and is of fundamental importance for the validation of autonomous driving. Due to the remarkable progress that the two communities of *Symbolic Computation* and of *Satisfiability Checking* made in recent years to solve industrially relevant problems and due to their recent intention of “joining forces”, we believe that the SC^2 community may give essential contributions based in their rich variety of diverse methods and techniques.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments that have helped us clarifying the presentation in this paper. Additionally, we would like to thank David Specht for his feedback on one of the examples and Karsten Scheibler for helpful discussions on potential constraint representations. This work was supported by the H2020-FETOPEN-2016-2017-CSA project SC^2 (712689).

References

1. Damm, W., Kemper, S., Möhlmann, E., Peikenkamp, T., Rakow, A.: Traffic sequence charts - from visualization to semantics. AVACS Technical Report 117 (Oct 2017)
2. Damm, W., Kemper, S., Möhlmann, E., Peikenkamp, T., Rakow, A.: Using traffic sequence charts for the development of HAVs. In: Embedded Real Time Software and Systems - ERTS2018 (2018)
3. Damm, W., Möhlmann, E., Peikenkamp, T., Rakow, A.: A formal semantics for traffic sequence charts. In: Festschrift in honor of Edmund A. Lee (2017)
4. Giorgetti, N., Pappas, G., Bemporad, A.: Bounded model checking of hybrid dynamical systems. In: 44th IEEE Conference on Decision and Control and 2005 European Control Conference (CDC-ECC'05). pp. 672–677 (2005)
5. International Organization for Standardization: Road vehicles – Functional safety (ISO 26262) (2011)
6. Kalra, N., Paddock, S.M.: Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? (2016), `https://www.rand.org/pubs/research_reports/RR1478.html`
7. Kemper, S., Etzien, C.: A visual logic for the description of highway traffic scenarios. In: Aiguier, M., Boulanger, F., Krob, D., Marchal, C. (eds.) Complex Systems Design & Management, Proceedings of the Fourth International Conference on Complex Systems Design & Management CSD&M 2013, Paris, France, December 4-6, 2013. pp. 233–245. Springer (2013), `https://doi.org/10.1007/978-3-319-02812-5_17`
8. Maurer, M., Gerdes, J.C., Lenz, B., Winner, H.: Autonomes Fahren - Technische, rechtliche und gesellschaftliche Aspekte. Springer Berlin Heidelberg (May 2015)
9. Menzel, T., Bagschik, G., Maurer, M.: Scenarios for Development, Test and Validation of Automated Vehicles. ArXiv e-prints (Jan 2018), `https://arxiv.org/abs/1801.08598`
10. Priya Inala, J., Gao, S., Kong, S., Solar-Lezama, A.: REAS: Combining Numerical Optimization with SAT Solving. ArXiv e-prints (Feb 2018), `https://arxiv.org/abs/1802.04408`
11. SAE-International: J3016_201609: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (Sep 2016)