Counterexample-Guided Safety Contracts for Autonomous Driving

Jonathan DeCastro[1⋆], Lucas Liebenwein[2⋆], Cristian-Ioan Vasile[2], Russ Tedrake[1,2], Sertac Karaman[2], and Daniela Rus[2]

[1] Toyota Research Institute, Cambridge, MA 02139, USA,
[2] Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract. Ensuring the safety of autonomous vehicles is paramount for their successful deployment. However, formally verifying autonomous driving decisions systems is difficult. In this paper, we propose a framework for constructing a set of safety contracts that serve as design requirements for controller synthesis for a given scenario. The contracts guarantee that the controlled system will remain safe with respect to probabilistic models of traffic behavior, and, furthermore, that it will follow rules of the road. We create contracts using an iterative approach that alternates between falsification and reachable set computation. Counterexamples to collision-free behavior are found by solving a gradient-based trajectory optimization problem. We treat these counterexamples as obstacles in a reach-avoid problem that quantifies the set of behaviors an ego vehicle can make while avoiding the counterexample. Contracts are then derived directly from the reachable set. We demonstrate that the resulting design requirements are able to separate safe from unsafe behaviors in an interacting multi-car traffic scenario, and further illustrate their utility in analyzing the safety impact of relaxing traffic rules.

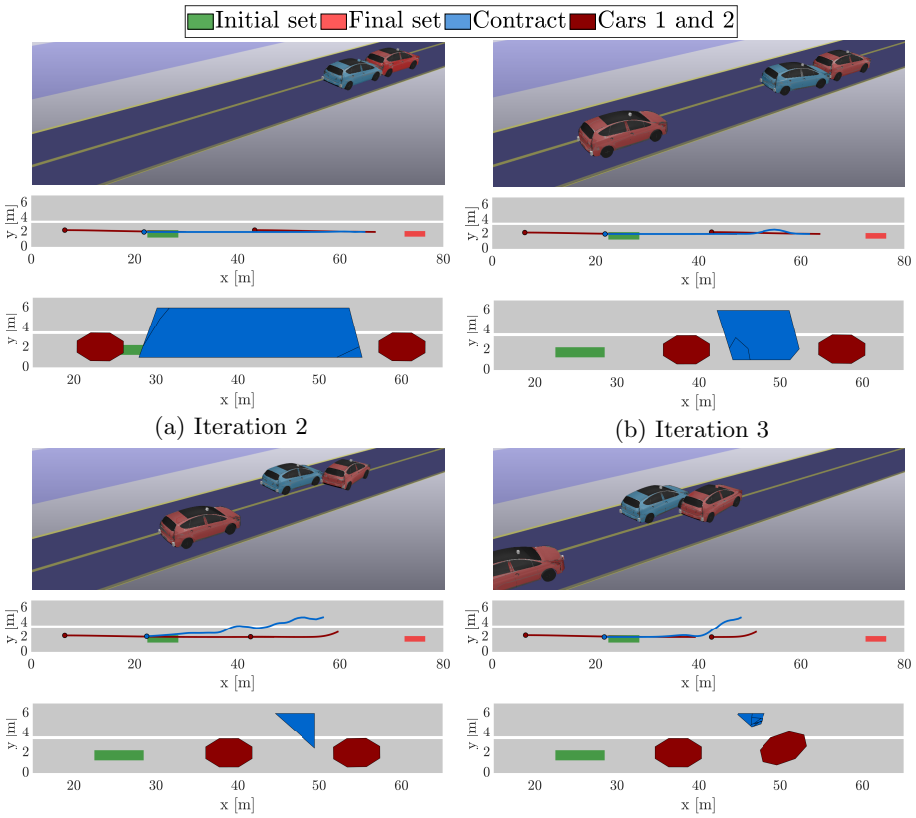
Keywords: Logic and Verification, Collision Avoidance, Falsification, Rules of the Road

1 Introduction

Traditional approaches to establishing trust in autonomous driving decision systems (ADDS) involve exhaustive road testing and numerical simulations. To address such needs, learning-based approaches have been gaining attention recently as being able to model realistic behaviors of traffic [34, 28, 24]. Nonetheless, testing-based strategies are still limited in their ability to cover all the possible scenarios encountered in the world, as they are costly to execute, and do not produce formal, interpretable safety certificates. Efforts focusing on formal proofs of safety guarantees with respect to vehicle, environment, and traffic models have therefore been proposed [2, 26].

For any realistic scenario involving a large number of interacting agents and complex ADDS implementations (some of which involve learned structures), obtaining correctness guarantees becomes prohibitively expensive. Moreover, the

⋆ equal contribution



(a) Iteration 2 (b) Iteration 3

(c) Iteration 7 (d) Iteration 14

Fig. 1: At each iteration, a counterexample to the traffic system is found that falsifies the proposed contract, i.e., we find a set of trajectories resulting in a collision, although the ego-car satisfies the proposed safety contract (upper two plots). Subsequently, we update the contract (the blue region in the lower plot) to guard against the counterexample. The procedure is repeated iteratively.

verification process typically must be executed anew whenever any part of the ADDS (e.g. planner, controller, perception systems) is modified. Verifying an ADDS is further challenged by the fact that the vehicle must operate in an unpredictable, probabilistic traffic environment, and be robust to a vast number of possible situations. Lastly, the concept of safety in autonomous driving incorporates notions of rules of the road. A formal approach should analyze the impact of such rules within the context of a scenario; for instance, in some situations it may be beneficial to relax certain rules for the sake of absolute safety of the traffic system.

In this work, we leverage formal *verification* and *falsification* to generate assume-guarantee *contracts* for safety of complex systems involving many interacting cars. Rather than certifying a particular ADDS, our system uses ver-

ification to guide the creation of constraints that guarantee the safety of the controlled system provided that the assumptions on the vehicle, traffic behaviors, and environment are met. The contracts consider certain rules of the road, such as "drive in the right lanes", in order to propose initial hypotheses on driving behaviors and to understand their relationship with respect to safety [37, 21]. Contracts may manifest as constraints used in the design of provably-safe controllers, e.g., within model-predictive control [16, 33]. They could enable safe autonomy in other ways; for instance, a contract could be part of an online assurance system, disengaging the decision system if it places the car's safety in jeopardy.

The contract synthesis method is based on reachability analysis [26] which computes, at each time step, the safe (collision-free) subset of the state space. To overcome tractability issues arising from the rich probabilistic behavior models of the other traffic participants, we employ falsification; searching for counterexamples of the required safety properties (e.g. collision-free behavior). We combine the two methods in an iterative framework, where, at each iteration: (a) a reachable set is computed for the autonomous vehicle (candidate contract), (b) counterexamples are generated which are used to synthesize constraints, and (c) the constraints are used to prune the unsafe states from the reachable set. Given a candidate contract, falsification solves for trajectories of the ego-car and the other vehicles to meet a chance constraint for collisions induced by a probabilistic model of traffic behaviors. For each counterexample trajectory found, the contract is updated, in a minimal way, with additional constraints. An example of adding such contracts is shown in Fig. 1. When no new counterexamples can be found, the output is a contract that certifies the safety of any controller up to a chance constraint on the modeled behavior of the traffic system.

The generated counterexamples depend on the driving style models of other vehicles as well as traffic rules. The framework can thus provide a means to assess, with probabilistic grounding, the impact of relaxing certain rules in order to preserve safety of the system as a whole. The case in Fig. 1 shows how the rule "drive in the right lanes" can be relaxed in order to keep the ego-car (blue) at a safe distance from the other cars in a highway overtake scenario.

The traffic models we use consider reactivity between agents, and are general enough to originate from black-box learning. We argue that finding falsification examples results in *explainable* verification: when a set of contracts is found, the counterexamples used in synthesis can explain the precise behaviors that the contracts guard against.

The contractual approach to verification and validation has been proposed for complex system design in many different domains [31]. It allows the separation of verification and design of controllers. Our work is similar to [22, 23], where a compositional approach to synthesizing contracts is presented for traffic networks that must adhere to global specifications. The idea of creating constraints based on a behavioral model may be viewed as an instance of robust explicit model-predictive control. The approach in [38] aims to synthesize controllers that satisfy a high-level task specification, while in [14] the goal is to synthesize simple partitions of the state-space based on reachable sets. In contrast to our

work, both approaches assume simple, non-probabilistic environment behaviors. The contracts in this paper are represented as simple state-space constraints on the car's behaviors, which are shown to be valid for highway-type scenarios, including lane keeping, lane changing, and merging.

Formal safety proofs can be obtained using a variety of methods. Reachability analysis for dynamical systems is one approach to formal verification that a large amount of research has been devoted to, e.g., finite [12], continuous [6, 10], and hybrid systems [3, 27, 4, 9]. Approximation algorithms for reachability analysis include zonotope-based computations [1], flowtubes [8], Hamilton-Jacobi formulations [27], and sampling-based approaches [25]. These have been used for safe motion planning [7, 17], reinforcement learning [19], and autonomous driving [2, 26]. Our work, compared to previous approaches, enables the inclusion of complex traffic behaviors and rules without sacrificing computational tractability.

Falsification [29, 10, 6] aims to find counterexamples that violate a given property, and enables the analysis of more complex systems than verification, at the cost of completeness. The problem becomes one of finding failures, rare events within distributions capturing realistic driving behaviors, which can be difficult to solve. Methods include counterexample guided abstraction refinement [11, 20] and sampling-based [6]. In contrast to sampling-based methods, e.g., cross-entropy methods [32], we contribute a gradient-based probabilistic optimization to falsification of systems involving a large number of agents in short scenarios with few discrete decisions. This approach allows us to quickly converge on solutions by optimizing their utility at each iteration step. The highly nonlinear nature of the problem prevents a globally-optimal solution, i.e., conclude that the added certificates are a formal proof of safety. However, our approach seeks to iteratively find locally-optimal counterexamples at each step of the contract-generation process and hence targets important failure cases that sampling-based approaches may miss.

We contribute the following:

1. A framework and software implementation for generating safety constraints (contracts) for ADDSs that consider rules of the road and probabilistic traffic behaviors for a wide array of any multi-lane highway-type scenarios.
2. A gradient-based falsification approach that allows to efficiently generate a wide variety of probabilistic traffic scenarios with tunable behavior via chance constraints.
3. Simulation results on real-world-inspired traffic scenarios.

2 Problem Statement

In this section, we introduce background and formally state the problem of computing contracts for driving scenarios.

2.1 Stochastic Models of the Traffic System

We start with uncertain continuous-time parameterized models of the form

$$\dot{x} = f_\rho(x, u, w) \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ are states, $u \in \mathcal{U} \subset \mathbb{R}^m$ are control inputs, $w \in \mathbb{R}^d$ is a Gaussian-distributed disturbance vector, $w \sim \mathcal{N}(0, \Sigma_\rho)$, where Σ_ρ is positive definite, and $\rho \in \mathcal{P}$ are fixed model parameters. Our system model $f_\rho(\cdot)$ is assumed to be C^1 continuous, and the sets $\mathcal{X}$ and $\mathcal{U}$ to be compact. We work from a decomposition of our system model as a coupling of N closed-loop parameterized traffic vehicle models, plus one additional system capturing the physics model for the ego vehicle:

$$\dot{x} = \begin{bmatrix} \dot{x}^0 \\ \dot{x}^1 \\ \vdots \\ \dot{x}^N \end{bmatrix} = \begin{bmatrix} f_{ego}(x^0, u) \\ f_{1,\rho_1}(x, w^1) \\ \vdots \\ f_{N,\rho_N}(x, w^N) \end{bmatrix} \tag{2}$$

Here, we decompose x as $x^i \in \mathcal{X}^i \subseteq \mathbb{R}^{n_{i,\rho_i}}$, w as $w^i \in \mathbb{R}^{d_{i,\rho_i}}$, $w^i \sim \mathcal{N}(0, \Sigma^i_{\rho_i})$, $i = 1, \ldots, N$, as representing an uncontrollable perturbation for each traffic vehicle i, explaining the uncertainties in how individual drivers behave. We dedicate u as being the driving commands for the ego vehicle, whose state is $x^0 \in \mathbb{R}^{n_{ego}}$. Given a discretization $k = \{0, \ldots, T\}$, we define a trajectory as the sequence of $\{x_k, u_k, w_k\}_{k=0}^T$, and denote $p(w_0, \ldots, w_T)$ as the joint probability density function over the disturbances $\{w_0, \ldots, w_T\}$.

Note that the above disturbance model satisfies many learning-based structures in the literature. For instance, to implement the model of [5], each vehicle's behavior model would take on a feedback form involving a nonlinear function of state and an additive Gaussian-distributed stochastic term, which is a special case of (1). The parameter ρ may characterize particular styles of driving behaviors, for instance the spectrum describing average driving to aggressive driving. We will illustrate this point further in Section 4.

2.2 Problem Formulation

Let a *scenario* be defined as a tuple $\mathcal{S} = (\mathcal{R}, \mathcal{P}, X_0, X_F^0)$ consisting of a specification of a road in $\mathbb{R}^2$ and its ruleset (a Boolean formula in states) $\mathcal{R}$, a fixed set of model parameters $\mathcal{P}$, and a set of possible initial conditions for each car $X_0 \subseteq \mathcal{X}$ and a final set for the ego car $X_F^0 \subseteq \mathcal{X}^0$.

Let φ be a *safety condition*, a Boolean formula denoting functions of states that describe the conditions for safety of the vehicle. φ can represent, for instance, collisions between cars, departing a lane, or breaking certain rules or liability bounds. We further define $\psi : \mathbb{R}^n \to \mathbb{R}$ to be a quantitative measure on the state space for the entire traffic system, and say that $x \in \mathbb{R}^n$ *satisfies* φ (i.e. $x \models \varphi$) if and only if $\psi(x) > 0$. Otherwise, the specification is *falsified* (i.e. $x \not\models \varphi$).

Our goal is, for a given scenario $\mathcal{S}$ and safety condition φ, to find a set of counterexamples to φ as bounded-time trajectories for all of the traffic participants. For each counterexample, we then seek a *contract* $\mathcal{C} \subset \mathcal{X}^0$ that can be applied as a rule for the ego vehicle to follow in order to guard against the counterexample and thereby locally satisfy φ. We impose the following requirements:

1. $\mathcal{C}$ yields certain constraints on the ego vehicle's trajectories that prevent violating a given ruleset (e.g. rules of the road),

2. $\mathcal{C}$ yields additional constraints on the ego vehicle's trajectories that prevent violation of φ with respect to the counterexamples associated with $\mathcal{C}$,
3. $\mathcal{C}$ generalizes to protect the ego vehicle against a continuum of possible traffic vehicle behaviors under w^i, in addition to those in the finite set of counterexamples, and
4. the counterexamples associated with $\mathcal{C}$ satisfy a chance constraint describing reasonable driver behaviors, i.e. $p(w_1, \ldots, w_T) \geq \alpha T$, where the tolerance $\alpha > 0$ bounds the likelihood of the behavior.

If a particular counterexample satisfies such a chance constraint, then we know that it is reasonably well-explained by the underlying behavior model of actual driver behaviors. On the other hand, if this check fails, then the counterexample can be considered to be "uncanny" behavior that does not resemble true driving behaviors and the ego vehicle needs not have a contract. Contracts with different road rules can be compared to examine the affordances or compromises to safety.

For the sake of simplicity of the contracts, the approach in this paper seeks to attain a *convex* contract representation that asserts, under the assumptions of the scenario $\mathcal{S}$, the ego vehicle is guaranteed to remain safe with respect to a finite, but diverse, set of counterexamples associated with $\mathcal{C}$. Our focus in this paper is pairwise collisions, as this keeps the problem within the continuous domain without sacrificing the variety in discovered counterexamples.

3 Constructing Safety Contracts

Contracts are created by an alternation between falsification and reachability analysis, under the scenario model (2). The overall approach is as shown in Algorithm 1. Starting with a set of initial contracts that enforce a certain ruleset, the falsification step (GenerateCounterexamples) generates counterexamples to these contracts (if any exist) by solving for possible ego-car and traffic behaviors that result in failure of φ. In the reachability step (GenerateContract), a reach-avoid problem is then solved to find the set of time-indexed states for the ego-car, to over-approximation, for which the ego car is able to steer away from the generated counterexample. The failure case is indicative of an undecidable result, where it is inconclusive whether the ego vehicle can take *any* action to remain safe under the given scenario.

Figure 2 depicts two iterations of the overall procedure. The left-hand side depicts the reachablility step, in which a ruleset and any existing contracts are considered as constraints in the reachable set computation. The right-hand side illustrates how we use falsification to find counterexamples with respect to the contracts. The counterexample is treated as an obstacle to avoid in the subsequent iteration, at which point, a set of constraints are created that separates the set difference between the reachable set at the previous step and the one at the current step.

Solving the reach-avoid problem, versus constructing contracts based only on counterexamples, serves two purposes. First, it allows selection of new contracts that minimize the volume of the reachable set treated as unsafe in the next

Algorithm 1 SynthesizeContracts

Input: $\mathcal{S} = (\mathcal{R}, \mathcal{P}, X_0, X_F^0)$: scenario, $\psi(\cdot)$: safety condition function, α: chance constraint, T: time horizon.
Output: $\mathcal{C}$: safety contract for each timestep $k \in \{0, \dots, T\}$.

$\mathcal{C} \leftarrow$ InitializeContract($\mathcal{S}$)
repeat
  $\bar{x}, p(\bar{w}) \leftarrow$ GenerateCounterexamples($\mathcal{S}, \mathcal{C}, \psi(\cdot), T$)
  $\mathcal{C} \leftarrow$ GenerateContract($\mathcal{S}, \mathcal{C}, \bar{x}$)
until $(p(\bar{w}) < \alpha T) \vee (\mathcal{C} = \emptyset)$
if $(\mathcal{C} = \emptyset) \vee \neg$IsReachable($X_F^0, \mathcal{C}_T$) **then**
  return failure
return $\mathcal{C}$

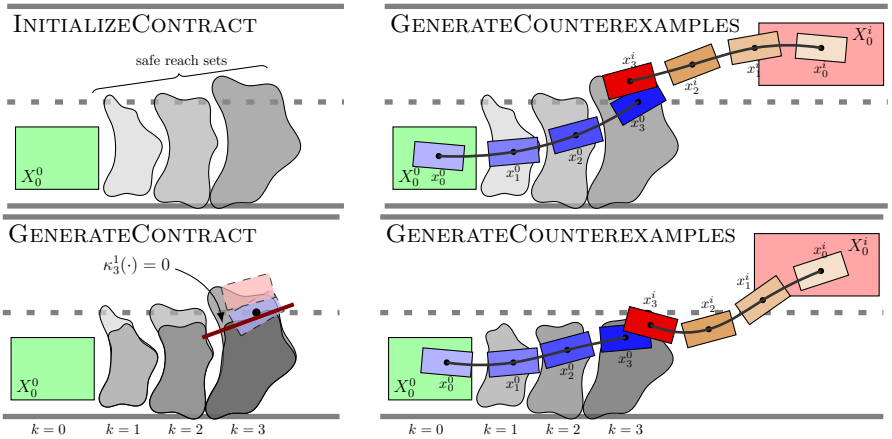


Fig. 2: Two iterations of the overall approach.

iteration under the contract. Second, we can verify whether it is feasible for the ego-car to reach the final set from within the initial set under the computed contracts (the check IsReachable in Algorithm 1).

3.1 Gradient-Based Probabilistic Falsification

In general, we are interested in finding counterexamples to the safety specification: dynamically feasible trajectories that falsify our safety condition φ within a given chance constraint on the underlying probabilistic behavior model. We do so using a formulation that takes into consideration analytical structure of these behavior models in order to perform gradient-based optimization to solve for a the most likely counterexample.

We solve a direct-collocation trajectory optimization problem [30], by discretizing time $k = \{0, \dots, T\}$, with time step h, where $t = hk$. We let $\bar{u}$, $\bar{w}$, $\bar{x}$ denote, respectively, the sequences $\bar{u} = \{u_k\}_{k=0}^T$, $\bar{w} = \{w_k\}_{k=0}^T$, and $\bar{x} = \{x_k\}_{k=0}^T$.

The counterexample trajectory is summarized by a collection of decision variables $\{h, \bar{u}, \bar{w}, \bar{x}\}$ that falsify the condition $\bar{x} \models \varphi$ but satisfy, at a minimum, the system dynamics (2), the initial conditions, and some threshold on the likelihood of selecting the random perturbations $\bar{w}$, denoted by $p(\bar{w})$. We aim to find the most likely explanations of the failure under the given model, motivating the following problem:

$$
\begin{aligned}
&\max_{h,\bar{u},\bar{w},\bar{x}} p(\bar{w}) \\
&\quad \text{s.t. } x_{k+1} - x_k = h f_{collocation}, \quad \forall k = 0, \dots, T-1 && \text{(dynamics)} \\
&\qquad x_k \in \mathcal{X}, \quad \forall k = 0, \dots, T \\
&\qquad u_k \in \mathcal{U}, \quad \forall k = 0, \dots, T-1 \\
&\qquad x_0 \in \mathcal{X}_0, \quad u_0 \in \mathcal{U} && \text{(initial conditions)} \\
&\qquad \psi(x_T) \leq 0 && \text{(safety specification)} \\
&\qquad \kappa_k^j(x_k) \leq 0, \quad \forall j = 1, \dots, Q, \ \forall k = 0, \dots, T && \text{(contracts)} \\
&\qquad p(\bar{w}) \geq \alpha T && \text{(chance constraint)}
\end{aligned}
\tag{3}
$$

where $f_{collocation} = \frac{1}{6}(f_k + 4\tilde{f} + f_{k+1})$, $f_k = f_\rho(x_k, u_k, w_k)$, and

$$
\tilde{f} = f_\rho\left(\frac{1}{2}(u_k + u_{k+1}) + \frac{h}{8}(f_k + f_{k+1}), \frac{1}{2}(u_k + u_{k+1}), \frac{1}{2}(w_k + w_{k+1})\right).
$$

The function $\kappa_k^j(\cdot) \in \mathcal{C}$ represents constraints of the form $(a_k^j)^T x_k \leq b_k^j$, $a_k^j \in \mathbb{R}^{n_{ego}}$, $b_k^j \in \mathbb{R}$ at time step k due to a contract j, the safety-preserving contracts the ego vehicle must adhere to. When (3) is solvable, we end up with corner cases to the hypothesis for $\kappa^j(\cdot)$ found thus far. Every time a new constraint is added to $\mathcal{C}$, the condition φ becomes harder and harder to falsify. We revisit the computation of $\kappa^j(\cdot)$ in Section 3.3. Notice that we can choose to leave out the last constraint in (3), since the optimal choice of $\bar{w}$ is a maximizer for $p(\bar{w})$ and hence a check of the optimal values is sufficient to verify the chance constraint.

The task now is to find the representation $p(\bar{w})$ and express $J(\bar{w})$ as a convex cost such that $\arg\max_{\bar{w}} p(\bar{w}) = \arg\max_{\bar{w}} J(\bar{w})$. Taking $w_k \sim \mathcal{N}(0, \Sigma)$ (where Σ is block-diagonal of Σ^i) and noting the probability of action w_k is drawn from the distribution $p(w_k) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2} w_k^T \Sigma^{-1} w_k)$ we can easily obtain the log-likelihood representation of the probability as

$$
\log p(\bar{w}) = \sum_{k=0}^{N} \log p(w_k) = -\frac{nN}{2} \log 2\pi - \frac{N}{2} \log |\Sigma| - \frac{1}{2} \sum_{k=0}^{N} w_k^T \Sigma^{-1} w_k.
$$

Due to monotonicity of the log operator, the cost function may be more simply written as $J(\bar{w}) = \sum_{k=0}^{N} w_k^T \Sigma^{-1} w_k$, and the chance constraint $\log p(\bar{w}) \geq \log \alpha - \log T$. By maximizing $p(\bar{w})$, we ensure that we are always searching for the most realistic counterexample.

Note that the problem in (3) is a nonconvex one to solve in general. Hence, we cannot guarantee a solution will be found, and hence cannot hope to exhaust all possible counterexamples (i.e. achieve completeness). However, it is important to note that we can achieve *soundness* of our solutions to (3).

3.2 Collision-Free Safety Conditions

Let $\mathcal{B}(x_k^i) \subset \mathbb{R}^2$ be the orientation-dependent footprint of the i^{th} vehicle at time k, that is, the Cartesian space occupied at state x^i. Our safety criteria is one where we wish to avoiding crashes with other vehicles, $\varphi = \bigwedge_{i,k}(\mathcal{B}(x_k^0) \cap \mathcal{B}(x_k^i) = \emptyset)$. We apply the mild assumption that we only search for conditions in which the ego vehicle is in collision with only one other vehicle at a time, which makes it easy to reduce a potentially combinatorial problem into one in which we solve (3) sequentially.

Unfortunately, finding analytical forms for collision of two rectangular objects involves Minkowski operations, which is difficult to solve analytically. We instead express collision in terms of two inscribing ellipses using the following result.

Lemma 1. *Let a_i and b_i be the length and width of vehicle i, $z_i = [\, x_i \; y_i \,]^T$ be its Cartesian coordinates, and θ_i its angle. Let $C_i = R(\theta_i) \begin{bmatrix} a_i & 0 \\ 0 & b_i \end{bmatrix}$, where $R(\theta_i)$ is a rotation matrix, and let $\tilde{z} = C_0^{-1}(z_1 - z_0)$. Then,*

$$\mathcal{B}(x^0) \cap \mathcal{B}(x^1) \neq \emptyset \Rightarrow \tilde{z}^T (C_1 + R_1)^{-1} (C_1 + R_1)^{-T} \tilde{z} \leq 1. \tag{4}$$

Proof. (sketch) Condition (4) can be obtained directly by transforming one of the ellipses to the unit disc, then applying the same transform to the other ellipse and writing out the expression for containment of the origin.

We note that the constraint (4) preserves soundness of the falsification problem; when a trajectory is found that satisfies this condition, that trajectory falsifies φ.

3.3 Reachability with Contracts

Let $F(t_k; X_0^0)$ denote an over-approximation to the reachable set at time t_k at iteration j of the main loop in Algorithm 1, i.e. the time-indexed set of states $x_k^0 \in \mathcal{X}^0$ for which there exists a control $u : \mathbb{R}_{\geq 0} \mapsto \mathcal{U}$ containing the trajectories satisfying $\dot{x}^0 = f_{ego}(x^0, u)$ when starting in the initial set X_0^0.

Our objective is essentially the converse of the falsification problem: to compute a safe reachable set for the ego vehicle $F_{safe}(t_k; X_0^0) \subseteq F(t_k; x_0^0)$ such that it preserves the ruleset $\mathcal{R}$ and is not in collision with any other traffic vehicle at all timesteps. An overview of the approach may be found in Algorithm 2. For simplicity, we only present the computation of forward reachable sets, but this can be extended to backward reachable sets with the modifications explained in [26]. Once a contract is created, we extend the reachable set to verify that it intersects the goal region. Similar [26], we consider overapproximation to the reachable set, which is in line with the goal of safety since the resulting counterexample might not even be dynamically feasible, but the safety contract can still guard against it.

Given a safe reachable set $F_{safe}(t_k; X_0^0)$ computed at some iteration j of the main loop in Algorithm 1, we want to select a hyperplane for each of the i vehicles of the form $(a_k^i)^T x_k^0 \leq b_k^i$ such that our new safe set $F_{safe}(t_k; X_0^0)$ at time step k

Algorithm 2 GenerateContract ($\oplus$ denotes the Minkowski sum)

Input: $\mathcal{S} = (\mathcal{R}, X_0, \mathcal{P})$: scenario, $\mathcal{C}'$: previous contract, $\bar{x}$: a trajectory for the system of traffic cars, T: time horizon.
Output: $\mathcal{C}$: safety contract for each timestep $k \in \{0, \dots, T\}$.

1: $F_{safe}(t_0; X_0^0) \leftarrow X_0^0 \cap C_0'$
2: $\mathcal{C}_k \leftarrow \mathcal{X}^0, \quad \forall k \in \{0, \dots, T\}$
3: **for all** $k \in \{0, \dots, T\}$ **do**
4: $\quad F(t_k; X_0^0) \leftarrow F(h; F_{safe}(t_{k-1}; X_0^0)) \cap \mathcal{C}_k'$ ▷ Compute the reachable set
5: $\quad O_k^i \leftarrow \mathcal{B}'(x_k^i, \mathcal{R}) \oplus \mathcal{B}(F(t_k; X_0^0))$ ▷ Compute the traffic constraints
6: $\quad$ **for all** $i \in \{1, \dots, N\}$ **such that** $O_k^i \neq \emptyset$ **do**
7: $\quad\quad \mathcal{C}_k \leftarrow$ ComputeContract$(F(t_k; X_0^0) \cap \mathcal{C}_k', O_k^i) \cap \mathcal{C}_k$
8: $\quad F_{safe}(t_k; X_0^0) \leftarrow F(t_k; X_0^0) \cap \mathcal{C}_k$ ▷ Compute the safe reachable set
9: **return** $\mathcal{C}$

is a valid reachable set that is safe with respect to the counterexample. Precisely, in line 4, values for a_k^i, b_k^i are selected such that $(a_k^i)^T \chi^i > b_k^i$ for all vertices χ^i of O_k^i, the i^{th} footprint of the counterexample, so that we obtain

$$F_{safe}(t_k; X_0^0) = F(t_k; X_0^0) \cap \{x \mid (a_k^i)^T x_k^0 \leq b_k^i, \forall i = 1, \dots, N\}. \tag{5}$$

That is, we select a_k^i and b_k^i such that we may treat it as an obstacle in computing the reachable set at future times. In line 5, we let $\mathcal{B}(X)$ denote an orientation-dependent Cartesian expansion of some set $X \subset \mathbb{R}^{n_{ego}}$, and let $\mathcal{B}'(x, \mathcal{R})$ denote a state-dependent inflation of $\mathcal{B}(x)$ according to the ruleset $\mathcal{R}$, as explained in Section 3.4.

Within ComputeContract, we select one hyperplane, i.e. a_k^i and b_k^i, in such a way as to maximize the volume of the resulting safe reachable set $F_{safe}(\cdot)$. If we assume $F(\cdot)$ is a union of polytopes, we can easily choose one from among the facets that maximizes the union of the remaining polytopes in (5) and satisfies $(a_k^i)^T \chi^i > b_k^i$. $\mathcal{C}_k$ is returned as the intersection of $\mathcal{C}_k'$ and the new contract.

3.4 Rules of the Road

In the following we consider the subset of rules from the *Vienna Convention on Traffic Rules* [15], see Table 1. We select these rules as they form a subset of engagement rules for highway scenarios, and exclude rules involving traffic signals and other discrete conditionals. For simplicity, we show the constraint sets for straight road segments, and equally sized cars.

We assume that the centerline of carriageway is along the x axis of the ego-car for straight road segments. The length of the road segment is denoted by L, the width of a lane by W, and the number of left and right lanes by n_{left} and n_{right}, respectively. A sequence $0 \leq \xi_x^1 < \zeta_x^1 < \xi_x^2 < \ldots < \zeta_x^{n_{solid}} \leq L$ defines the solid line segments $(\xi_x^\ell, \zeta_x^\ell)$ along the centerline of the road. The pose and longitudinal speed of the vehicles are denoted by (x_c^i, y_c^i, θ^i) and v^i, respectively, where $0 \leq i \leq N$, and $i = 0$ represents the ego-car. The average speed of

Table 1: Rules of the road for highway scenarios.

No.	Rule	Constraint set
1	Don't drive in the left lanes.	$\{0 \le x_c^0 \le L, -n_{right} \cdot W \le y_c^0 \le 0\}$
2	If driving behind another car, keep a reasonable distance away to avoid collision if it suddenly stops.	$\{x_c^i - x_c^0 \ge \epsilon_x^{safe}\, v^0 \mid \forall i \,.\, x_c^i - x_c^0 \ge 0 \wedge \lvert y_c^i - y_c^0\rvert < W\}$
3	If you want to slow down, give clear warning and do not inconvenience drivers behind you.	$\{x_c^0 - x_c^i \ge \epsilon_x^{safe}\, v^0 \mid \forall i \,.\, x_c^0 - x_c^i \ge 0 \wedge \lvert y_c^i - y_c^0\rvert < W\}$
4	Don't cross solid lines.	$\{\xi_x^\ell \le x_c^0 \le \zeta_x^\ell \wedge -n_{right} \cdot W \le y_c^0 \le 0 \mid 1 \le \ell \le n_{solid}\}$
5	Overtake on the left when it is safe.	$\{y_c^0 - y_c^i > W \wedge v^0 > v^i \mid$ $\forall i \,.\, v^i > 0 \wedge \lvert x_c^0 - x_c^i\rvert \le \epsilon_x^{overtake} \wedge$ $\nexists j \,.(\, \lvert x_c^j - x_c^i\rvert \le \epsilon_x^{safe-overtake} \wedge y_c^0 - y_c^j \le W)\}$
6	If another vehicle is trying to overtake you keep right and don't accelerate. If necessary, slow down and pull over.	$\{u_a^0 \le 0 \wedge y_c^i - y_c^0 \ge W \wedge y_c^0 \le 0 \mid$ $\forall i \,.\, y_c^i - y_c^0 \le 1.5W \wedge v^i > 0 \wedge \lvert x_c^i - x_c^0\rvert \le \epsilon_x^{overtake}\}$
7	If passing oncoming traffic, leave sufficient lateral space to not get hit. If obstructed, slow down.	$\{y_c^i - y_c^0 \ge \epsilon_y^{safe} \mid y_c^i \ge 0 \wedge v^i \le 0\}$
8	Don't drive abnormally slowly such that you impede the progress of other vehicles. Don't drive above the speed limit or abnormally fast.	$\{\lvert v^0 - \bar{v}\rvert \le \epsilon_v, \lvert v^0\rvert \le \epsilon_v^{legal}\}$

vehicles around the ego-car and in the same lane is denoted by $\bar{v}$. The safe distances to other vehicles ahead and behind the ego is expressed as $\epsilon_x^{safe} > 0$, while the lateral safe distance to oncoming vehicles is expressed as $\epsilon_y^{safe} > 0$. Overtaking maneuvers are performed within a stretch of the road segment of length $2\,\epsilon_x^{overtake} > 0$ centered on the car that is being overtaken. Overtaking is safe if there are no other cars in the left lane where the ego-car performs the maneuver within a distance of $\epsilon_x^{safe-overtake}$ around the car being overtaken. Lastly, the legal speed limit for a lane is given by $\epsilon_v^{legal} > 0$.

Those rules that are only a function of the ego car (rules 1, 4, 8) are included in InitializeContract, while those that are functions of the joint state space are included only when a counterexample is obtained from the falsification step. Hence, these rules are included in ComputeContract as a modification to the traffic car footprint, i.e. $\mathcal{B}'(\cdot)$.

4 Results

We implemented the falsification algorithm, scenario, and system models using the Drake toolbox [35]. We use the SNOPT optimization package [18] for solving the sequential quadratic program (SQP) in (3). We furthermore parallelize the constraint evaluation before passing to the solver in order to speed up the solve time. To generate new contracts, we compute the reachable sets using a Taylor expansion to the nonlinear dynamics with sets being expressed as zonotopes; we do this with the aid of the CORA reachability tool [1]. Set operations are carried out using the MPT toolbox, which is based on a polytopic representation of sets.

Table 2: Parameters used to model driver behaviors for the traffic cars.

	Description	Symbol	Driving Style Normal	Aggressive
IDM	Reference speed (m/s)	v_{ref}	10	1.5
	Maximum acceleration (m/s^2)	a	1	4
	Comfortable deceleration (m/s^2)	b	3	6
	Minimum-desired net distance (m)	s_0	1	0.5
	Time headway to lead vehicle (s)	t_h	0.1	0.05
	Free-road exponent	δ	4	4
Pure-Pursuit	Lookahead distance (m)	s_{look}	15	10
Perception	Range (m)	$s_{perception}$	100	100
Disturbances	Steering angle variance (rad^2)	σ_δ	0.1	5
	Acceleration variance (m^2/s^4)	σ_a	0.1	2.5

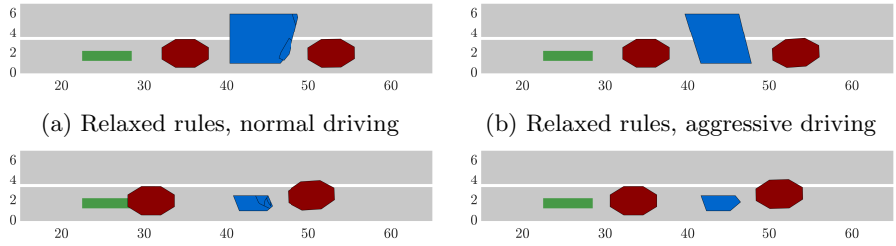
We consider the following model for both the ego vehicle and traffic vehicles:

$$\dot{x} = \begin{bmatrix} \dot{x_c} \\ \dot{y_c} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \frac{v}{L}\tan(u_\delta) \\ u_a \end{bmatrix}, \quad u = \begin{bmatrix} u_\delta \\ u_a \end{bmatrix}.$$

As previously, x_c and y_c are Cartesian positions at the center of the vehicle, θ is the heading angle, and v is the forward speed, while u_δ and u_a denote the steering angle and acceleration inputs, respectively.

To represent naturalistic behaviors for the traffic cars, we consider the intelligent driver model (IDM) [36], a model whose parameters are typically fit to driver data and which is used to represent the longitudinal actions (acceleration) of actual drivers. We consider a pure-pursuit controller [13] to model the lateral actions (steering) of drivers. Essentially, the IDM model allows cars to react to one another, while adapting to a driver's preferences for speed, acceleration and time headway between vehicles. The pure-pursuit controller allows steering to be adjusted smoothly so that the vehicle converges to a desired curve. In our experiements, we set the desired curve to be fixed as the centerline of a target lane to drive to. We randomize these traffic behaviors by defining Σ^i, where $\Sigma^i = \text{diag}\{\sigma_\delta, \sigma_a\}$, and be treating the disturbance signal as additive noise to the nominal acceleration and steering commands provided by IDM and pure pursuit. We adapt to different driving styles by using the complete list of parameters in Table 2. We furthermore augment the implementation by ignoring vehicles beyond a limited perception range.

The road geometry is configurable for highway scenarios in which any number of lanes and lane sizes can be chosen for the scenario. Our aggregate model of the entire traffic system is implemented to account for all of the traffic participants in the context of their positions on the road. The models are encoded such that partial derivatives can be easily obtained via automatic differentiation.



(a) Relaxed rules, normal driving

(b) Relaxed rules, aggressive driving

(c) Strict rules, normal driving

(d) Strict rules, aggressive driving

Fig. 3: The contract for timestep $t = 4.8s$ at iteration 4 for each set of parameters.

4.1 Leading/Trailing Car Scenarios

In this scenario, we consider the ego-car sandwiched between two traffic cars in the right lane of a two-lane highway with opposing traffic lane, which may be used for overtaking if free. We synthesized contracts using both the ruleset in Table 1 and again with a relaxed ruleset, in which we disable rules 1 and 3 to enable evasive maneuvers onto the other lane. For both rulesets, we explore traffic models having two levels of aggressiveness (normal and aggressive) using the parameters in Table 2. The overall computation time for the synthesis roughly ranged from 8h to 10h. In Fig. 1, we depict different iterations of Algorithm 1 for the relaxed rules and normal driving style. We compare the contracts obtained at a fixed iteration of the algorithm for each case in Fig. 3 and, for each case, report the log-likelihood of the counterexample normalized on $|\Sigma|$ in Fig. 4.

We observe that with more iterations (and more unlikely behaviors of the traffic cars), more contracts are added, making the contract more restrictive, but also harder to falsify, as indicated by the log-likelihood. With a greater number of rules and more aggressive traffic, we note that the contract gets smaller and more prohibitive (see Fig. 3). We also note that relaxing the ruleset (e.g. allowing lane switches) enables more behaviors for the ego-car, demonstrating that safety can be preserved at the expense of rule-breaking in some scenarios. Moreover, the ego-car can readily estimate the cost of violating rules of the road by observing the varying contracts depending on the set of actively enforced rules.

We note that for the normal driving style, the log-likelihood quickly decreases, whereas for the aggressive driving styles, the log-likelihood remains high as contracts are added, indicating that aggressive traffic can induce failure regardless of the ego-car's behavior. In both of the agressive-driving cases, empty contracts were returned before exhausting possible counterexamples. Of the normal-driving cases, the relaxed set provides a contract with 14 counterexamples, whereas the strict set provides five counterexamples, indicating that changing lanes presents more possible failure events to guard against.

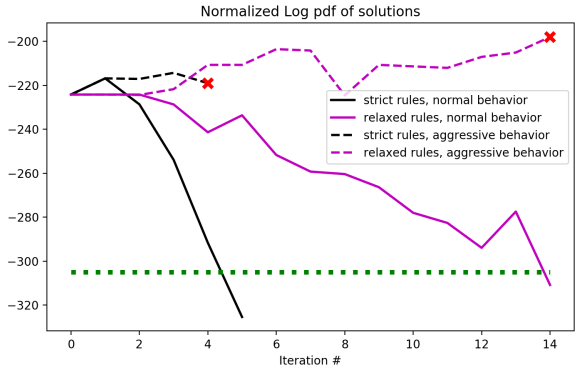


Fig. 4: The log-likelihood for each test case across all iterations. The red $\times$ marks iterations where the contract terminated with an empty set, and the green dashed line indicates the chance constraint α.

5 Conclusion

In this paper, we presented a novel framework for the synthesis of safety constraints for autonomous decision systems that can be applied and used by a wide variety of real-world systems. The framework allows for incorporating a large variety of scenarios, with a diverse set of probabilistic traffic behaviors, and for subsequently generating the appropriate safety constraints. We overcome issues of computational tractability by iteratively generating a set of safety constraints, based on reachability analysis, and generating counterexamples, i.e., traffic scenarios, using gradient-based probabilistic falsification. We judiciously account for rules of the roads, in terms of state space constraints enforced during reachability analysis.

The empirical results on a variety of real-world inspired scenarios validate the favorable performance of our approach, and reaffirm the practical applicability. We envision that our method can be used to inform the decision-making and planning system of an autonomous agent about the appropriate safety constraints applicable in a particular traffic scenario. In future work, we plan to extend our method to allow for behavior sequencing (such as collisions leading to further collisions), synthesize safety constraints that are simultaneously applicable across a wide variety of traffic scenarios, and show its effectiveness in real-world experiments.

Acknowledgements

Toyota Research Institute ("TRI") provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors, and not TRI or any other Toyota entity.

References

1. Althoff, M.: An introduction to CORA 2015. In: Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems. pp. 120–151 (2015)
2. Althoff, M., Dolan, J.: Online verification of automated road vehicles using reachability analysis. IEEE Transactions on Robotics 30, 903–918 (2014)
3. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The Algorithmic Analysis of Hybrid Systems. Theoretical Computer Science 138(1), 3–34 (1995)
4. Alur, R., Dang, T., Ivančić, F.: Predicate Abstraction for Reachability Analysis of Hybrid Systems. ACM Transactions on Embedded Computing Systems (TECS) 5(1), 152–199 (2006)
5. Baram, N., Anschel, O., Caspi, I., Mannor, S.: End-to-end differentiable adversarial imitation learning. In: ICML (2017)
6. Bhatia, A., Frazzoli, E.: Incremental Search Methods for Reachability Analysis of Continuous and Hybrid Systems. In: Hybrid Systems: Computation and Control (2004)
7. Chen, M., Hu, Q., Fisac, J.F., Akametalu, K., Mackin, C., Tomlin, C.J.: Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways. Journal of Guidance, Control, and Dynamics 40(6), 1360–1373 (2017)
8. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Flow*: An Analyzer for Non-Linear Hybrid Systems. In: Intl Conference on Computer Aided Verification (2013)
9. Chen, X., Schupp, S., Makhlouf, I., Ábrahám, E., Frehse, G., Kowalewski, S.: A Benchmark Suite for Hybrid Systems Reachability Analysis. In: NASA Formal Methods Symposium (2015)
10. Cheng, P., Kumar, V.: Sampling-based Falsification and Verification of Controllers for Continuous Dynamic Systems. The International Journal of Robotics Research 27(11-12), 1232–1245 (2008)
11. Clarke, E., Grumberg, O., Jha, S., Lu, Y., Veith, H.: Counterexample-Guided Abstraction Refinement. In: Intl Conference on Computer Aided Verification (2000)
12. Clarke, E., Grumberg, O., Long, D.: Verification Tools for Finite-State Concurrent Systems. In: Workshop/School/Symposium of the REX Project (Research and Education in Concurrent Systems) (1993)
13. Coulter, R.C.: Implementation of the pure pursuit path tracking algorithm. Tech. Rep. CMU-RI-TR-92-01, Carnegie Mellon University, Pittsburgh, PA (1992)
14. DeCastro, J.A., Kress-Gazit, H.: Nonlinear controller synthesis and automatic workspace partitioning for reactive high-level behaviors. In: ACM Intl Conference on Hybrid Systems: Computation and Control (HSCC). Vienna, Austria (2016)
15. Economic Commission for Europe – Inland Transport Committee, Vienna, Austria: Convention on Road Traffic, E/CONF.56/16/Rev.1/Amend.1 edn. (1968)
16. Erlien, S.M., Fujita, S., Gerdes, J.C.: Shared steering control using safe envelopes for obstacle avoidance and vehicle stability. IEEE Transactions on Intelligent Transportation Systems 17, 441–451 (2016)
17. Fisac, J., Bajcsy, A., Herbert, S., Fridovich-Keil, D., Wang, S., Tomlin, C., Dragan, A.: Probabilistically safe robot planning with confidence-based human predictions. In: Proceedings of Robotics: Science and Systems. Pittsburgh, PA (June 2018)
18. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM Rev. 47(1), 99–131 (2005)
19. Ivanovic, B., Harrison, J., Sharma, A., Chen, M., Pavone, M.: Barc: Backward reachability curriculum for robotic reinforcement learning. arXiv:1806.06161 (2018)

20. Kapinski, J., Deshmukh, J., Sankaranarayanan, S., Arechiga, N.: Simulation-guided Lyapunov Analysis for Hybrid Dynamical Systems. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control (2014)
21. Karlsson, J., Vasile, C.I., Tumova, J., Karaman, S., Rus, D.: Multi-vehicle motion planning for social optimal mobility-on-demand. In: IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia (2018)
22. Kim, E.S., Arcak, M., Seshia, S.A.: Compositional controller synthesis for vehicular traffic networks. IEEE Conf on Decision and Control (CDC) pp. 6165–6171 (2015)
23. Kim, E.S., Sadraddini, S., Belta, C., Arcak, M., Seshia, S.A.: Dynamic contracts for distributed temporal logic control of traffic networks. In: IEEE Conference on Decision and Control (CDC). pp. 3640–3645 (2017)
24. Kuefler, A., Morton, J., Wheeler, T.A., Kochenderfer, M.J.: Imitating driver behavior with generative adversarial networks. IEEE Intelligent Vehicles Symposium (IV) pp. 204–211 (2017)
25. Liebenwein, L., Baykal, C., Gilitschenski, I., Karaman, S., Rus, D.: Sampling-based approximation algorithms for reachability analysis with provable guarantees. In: Proceedings of Robotics: Science and Systems. Pittsburgh, PA (June 2018)
26. Liebenwein, L., Schwarting, W., Vasile, C.I., DeCastro, J., Alonso-Mora, J., Karaman, S., Rus, D.: Compositional and contract-based verification for autonomous driving on road networks. In: Intl Symposium on Robotics Research (ISRR) (2017)
27. Mitchell, I.M., Bayen, A.M., Tomlin, C.J.: A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. IEEE Transactions on Automatic Control 50(7) (2005)
28. Morton, J., Kochenderfer, M.J.: Simultaneous policy learning and latent state inference for imitating driver behavior. IEEE International Conference on Intelligent Transportation Systems (ITSC) pp. 1–6 (2017)
29. Plaku, E., Kavraki, L., Vardi, M.: Falsification of LTL Safety Properties in Hybrid Systems. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems (2009)
30. R. Hargraves, C., Paris, S.: Direct trajectory optimization using nonlinear programming and collocation. AIAA J. Guidance 10, 338–342 (1987)
31. Sangiovanni-Vincentelli, A., Damm, W., Passerone, R.: Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems. In: 2011 Control and Decision Conference and European Control Conference (2012)
32. Sankaranarayanan, S., Fainekos, G.: Falsification of temporal properties of hybrid systems using the cross-entropy method. In: ACM International Conference on Hybrid Systems: Computation and Control. pp. 125–134 (2012)
33. Schwarting, W., Alonso-Mora, J., Paull, L., Karaman, S., Rus, D.: Parallel autonomy in automated vehicles: safe motion generation with minimal intervention. In: IEEE International Conference on Robotics and Automation (ICRA) (2017)
34. Shalev-Shwartz, S., Shammah, S., Shashua, A.: Safe, multi-agent, reinforcement learning for autonomous driving. CoRR abs/1610.03295 (2016)
35. Tedrake, R., the Drake Development Team: Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems (2016), http://drake.mit.edu
36. Treiber, M., Kesting, A.: Traffic Flow Dynamics (2013)
37. Vasile, C.I., Tumova, J., Karaman, S., Belta, C., Rus, D.: Minimum-violation scLTL motion planning for mobility-on-demand. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1481–1488. Singapore (2017)
38. Wongpiromsarn, T., Topcu, U., Murray, R.M.: Receding horizon temporal logic planning for dynamical systems. In: IEEE Conference on Decision and Control (CDC) and Chinese Control Conference. pp. 5997–6004 (2009)