

Traffic Scenario Logic: A Spatial-Temporal Logic for Modeling and Reasoning of Urban Traffic Scenarios

Ruolin Wang^{1,2*}, Yuejiao Xu^{1,2*}, Jianmin Ji^{1,3†}

¹University of Science and Technology of China, 230026, China

²Suzhou Shuzhi Technology Group Co., Ltd., Suzhou 215000, Jiangsu, China

³Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, 230088, China

{rl_wang, yuejiao}@mail.ustc.edu.cn, jianmin@ustc.edu.cn

Abstract

Formal representations of traffic scenarios can be used to generate test cases for the safety verification of autonomous driving. However, most existing methods are limited to highway or highly simplified intersection scenarios due to the intricacy and diversity of traffic scenarios. In response, we propose Traffic Scenario Logic (TSL), which is a spatial-temporal logic designed for modeling and reasoning of urban pedestrian-free traffic scenarios. TSL provides a formal representation of the urban road network that can be derived from OpenDRIVE, i.e., the de facto industry standard of high-definition maps for autonomous driving, enabling the representation of a broad range of traffic scenarios without discretization approximations. We implemented the reasoning of TSL using Telingo, i.e., a solver for temporal programs based on Answer Set Programming, and tested it on different urban road layouts. Demonstrations show the effectiveness of TSL in test scenario generation and its potential value in areas like decision-making and control verification of autonomous driving. The code for TSL reasoning has been open-sourced.

Code — <https://github.com/USTCAVSA/TSL>

1 Introduction

Autonomous driving technology is promising to revolutionize transportation by enhancing safety, efficiency, and accessibility. The rigorous testing and verifying their safety under various traffic scenarios is central to developing and deploying autonomous vehicles. Formal representations of these scenarios are crucial for generating comprehensive test cases and ensuring the reliability of autonomous driving systems (Zhang et al. 2021).

Existing methods for formalizing traffic scenarios often face limitations, particularly in modeling intricate urban environments with diverse traffic patterns. While some approaches excel in representing highway scenarios or simplified intersections, they struggle to capture the complexity inherent in urban driving environments (Hilscher and Schwammbeger 2016; Yao and Li 2023; Li, Tao, and Wotawa 2020). Consequently, there is a pressing need for a

versatile and robust formal logic framework capable of modeling a broad spectrum of urban traffic scenarios.

In response to this challenge, we introduce *Traffic Scenario Logic (TSL)*, a novel spatial-temporal logic specifically tailored for modeling and reasoning about urban pedestrian-free traffic scenarios. Unlike previous methods, TSL is designed to provide a comprehensive formal representation of the urban road network without discretization approximations, enabling the modeling of various traffic configurations encountered in complex urban environments.

One of the main benefits of TSL is its compatibility with OpenDRIVE, which is the industry-standard format used for high-definition (HD) maps in autonomous driving (Dupuis, Strobl, and Grezlikowski 2010). By utilizing OpenDRIVE data, TSL seamlessly translates real-world road networks into logical representations, enabling the representation of a wide range of urban traffic scenarios.

TSL also distinguishes itself with a unique spatial representation. Instead of using a simple discretized position description, TSL captures a vehicle’s longitudinal position based on its relationship with other vehicles. By merging scenarios with the same vehicle relationships, TSL improves efficiency without losing comprehensive coverage of scenarios.

We first give the syntax and semantics of TSL for uni-direction multi-lane highway scenarios in Section 3, then extend it to urban scenarios in Section 4. Finally, we implement the reasoning of TSL with Telingo (Cabalar et al. 2019), a solver for temporal programs based on Answer Set Programming. Section 5 demonstrates several examples on different urban road layouts, showing the effectiveness of TSL in test scenario generation and potential applications in decision-making and control verification of autonomous driving.

In summary, this paper makes the following contributions:

- Introduces Traffic Scenario Logic (TSL) for formal modeling and reasoning of urban pedestrian-free traffic scenarios.
- Provides a formal representation of an urban road network that is compatible with the OpenDRIVE HD map format and an efficient and complete spatial representation.
- Implements the reasoning mechanism with Telingo

*These authors contributed equally.

†Corresponding author.

solver and provides examples of different road layouts.

2 Related Work

Constructing a generic spatial-temporal model for the physical world is a challenging and significant field of study. Prior research has primarily focused on developing different combinations of temporal and spatial logics. Kontchakov et al. (2007) have provided a comprehensive summary of this research.

Hilscher et al. (2011) were one of the first few work on using spatial-temporal logic to model traffic scenarios, proposing *Multi-Lane Spatial Logic (MLSL)* which provides an abstract model of a uni-directional multi-lane motorway. In MLSL, vehicles' position, speed, and acceleration constitute a traffic snapshot, and the traffic snapshots at different moments constitute a traffic scenario. Changes between traffic snapshots are described by transition rules, inspired by *ITL* (Moszkowski 1982). Hilscher, Linker, and Olderog (2013) proposed an extended version of MLSL that supports a bi-directional motorway and length measurement for road segments. Hilscher and Schwammberger (2016) further proposed *Urban Multi-Lane Spatial Logic (UMLS)* which contains the modeling of intersections, simplifying an intersection of two two-way, single-lane highways into four crossing road segments. Meanwhile, Xu and Li (2016) introduced vertical lanes that intersect with the horizontal lanes by extending the position to two dimensions to create an urban road network, formalized in a grid way. Based on UMLS, Schwammberger and Alves (2021) proposed *Urban Spatial Logic for Traffic Rules (USL-TR)* to add static objects and non-autonomous road users. Xu et al. (2019) represented an urban road network with road segments and the arcs that link them together, which was further formalized into a hierarchical structure by Yao and Li (2023). Some other research includes modeling an intersection controlled by traffic lights (Loos and Platzer 2011); using the State-Clock Logic for the representation of time (Bischopink and Olderog 2022); and employing ontology (Bagschik, Menzel, and Maurer 2018; Li, Tao, and Wotawa 2020).

The above methods for describing spatial relationships rely on either road segments or a grid-based occupancy approach, which have their limitations. Distinguishing between consecutive free grids or road segments is usually not necessary, and it can be crucial to describe the positional relationships of vehicles within the same grid or road segment. In this regard, a simple discretized description of the road network is often insufficient for test scenario generation.

3 Highway Traffic Scenario Logic

3.1 Abstract Model

This section formally defines a scenario on a uni-direction multi-lane highway, where one or more vehicles travel on a road with multiple lanes. The road length is infinite, and the number of lanes remains constant throughout. Please note that a road with variable lane numbers can be modeled as separate roads, which will be discussed in Section 4. Vehicles can perform lane-changing maneuvers between lanes to switch to adjacent lanes.

We denote the set of vehicles as $\mathbb{C} = \{c_1, c_2, \dots, c_N\}$, where N is the number of vehicles in the scenario. The set of lanes is denoted as $\mathbb{L} = \{l_1, l_2, \dots, l_M\}$, where M is the number of lanes. Then, we can define a road as follows:

Definition 1 (Road). *A road is an ordered arrangement of lanes of the lane set \mathbb{L} , denoted as $\mathcal{R} = \langle l_{i_1}, l_{i_2}, \dots, l_{i_M} \rangle$. A binary relation \prec_{left} is defined on it, where $l_{i_j} \prec_{left} l_{i_{j+1}}$, $1 \leq j \leq M - 1$.*

The lateral position of a vehicle (perpendicular to the direction of the road) is described using the lanes it occupies, while the longitudinal position (along the direction of the road) can be described using the positional relationship with other vehicles. Using the road coordinate system, it can be defined as follows:

Definition 2 (Longitudinal Positional Relationship). *Denote the range occupied by vehicle c as $[s_c^-, s_c^+]$, where s_c^- and s_c^+ represent the s -coordinates of the rear and front ends of the vehicle under Frenet coordinate system (Werling et al. 2010), respectively. Assuming that the S axis direction aligns with the direction of the vehicle, we can categorize the longitudinal positional relationship between two vehicles c_1 and c_2 on the same road as follows:*

1. **Ahead:** *If $s_{c_1}^- > s_{c_2}^+$, then c_1 is ahead c_2 .*
2. **Cover:** *If $s_{c_1}^- \leq s_{c_2}^+$ and $s_{c_1}^+ \geq s_{c_2}^-$, then c_1 covers c_2 .*
3. **Behind:** *If $s_{c_1}^+ < s_{c_2}^-$, then c_1 is behind c_2 .*

The definition is similar when the S axis direction is opposite to that of the vehicles. Obviously, their longitudinal positional relationship falls into one of these three categories for two vehicles on the same road.

Then we can define a *scene*, which is a ‘‘frame’’ of a scenario:

Definition 3 (Scene on Highway). *A scene on a uni-direction multi-lane highway is a pair $\mathfrak{s} = (\mathcal{L}, \mathcal{D})$, where:*

- $\mathcal{L} : \mathbb{C} \mapsto \mathcal{P}(\mathbb{L})$ *is a mapping from the set of vehicles \mathbb{C} to the power set of the set of lanes $\mathcal{P}(\mathbb{L})$, assigning to each vehicle the set of lanes to which it belongs; note that vehicles may cross lane dividers, so a vehicle may belong to more than one lane.*
- $\mathcal{D} : \mathbb{C} \times \mathbb{C} \mapsto \mathbb{D}$ *is a mapping from the set of vehicle pairs to the set of longitudinal positional relationship $\mathbb{D} = \{\text{ahead, cover, behind}\}$.*

A scenario is a sequence of scenes, defined as follows:

Definition 4 (Scenario on Highway). *A scenario on a uni-direction multi-lane highway is a quintuple $\mathcal{S} = (\mathbb{C}, \mathbb{L}, \mathbb{D}, T, \langle \mathfrak{s}_0, \mathfrak{s}_1, \dots, \mathfrak{s}_{T-1} \rangle)$, where:*

- *The meaning of \mathbb{C} , \mathbb{L} and \mathbb{D} is as stated above;*
- $T \geq 1$ *is the number of scenes included in the scenario, i.e. the length of the scenario;*
- $\mathfrak{s}_i = (\mathcal{L}_i, \mathcal{D}_i)$ *is the i -th scene that occurs in the scenario in time sequence.*

For convenience, further define the i -tail of a scenario as:

Definition 5 (i -tail of a Scenario). *The i -tail of scenario \mathcal{S} , denoted by \mathcal{S}_i , is defined as $(\mathbb{C}, \mathbb{L}, \mathbb{D}, T - i, \langle \mathfrak{s}_i, \mathfrak{s}_{i+1}, \dots, \mathfrak{s}_{T-1} \rangle)$ for all $0 \leq i < T$*

Now we provide the syntax and semantics of the TSL for uni-direction multi-lane highway scenario.

Definition 6 (Syntax of Highway TSL). *A formula of highway TSL can be recursively defined as:*

$$\begin{aligned} \phi ::= & \top \mid \text{on}(c, l) \mid \text{left}(l_1, l_2) \mid \text{lonr}(c_1, c_2, d) \mid c_1 = c_2 \\ & \mid l_1 = l_2 \mid d_1 = d_2 \mid \neg\phi \mid \phi_1 \rightarrow \phi_2 \mid \bigcirc\phi \mid \square\phi \\ & \mid \mathbb{F} \mid \forall c\phi \mid \forall l\phi \mid \forall d\phi \end{aligned}$$

where, $c, c_1, c_2 \in \mathbb{C}$, $l, l_1, l_2 \in \mathbb{L}$, $d, d_1, d_2 \in \mathbb{D}$. The modal operators \bigcirc and \square take their meanings ‘‘Next’’ and ‘‘Globally’’ in temporal logic, respectively. \mathbb{F} denotes that the length of this scenario is 1, indicating the final state.

Remark 1. We can further define $\phi \wedge \psi$, $\phi \vee \psi$, $\exists x\phi$ and $\diamond\phi$ as $\neg(\phi \rightarrow (\neg\psi))$, $(\neg\phi) \rightarrow \psi$, $\neg\forall x(\neg\phi)$ and $\neg\square\neg\phi$, respectively.

Please note that the temporal modal operators here represent the transition of scenes, indicating only the chronological order, not the exact time. The interval between steps can last for an arbitrary duration and can be different.

Definition 7 (Semantics of Highway TSL). *Given a road \mathcal{R} and a scenario \mathcal{S} , the satisfaction of formulae in the Definition 6 follows the semantics listed below:*

$$\begin{aligned} \mathcal{R}, \mathcal{S} \models \top & \quad \text{for all } \mathcal{R}, \mathcal{S} \\ \mathcal{R}, \mathcal{S} \models \text{on}(c, l) & \quad \Leftrightarrow l \in \mathcal{L}_0(c) \\ \mathcal{R}, \mathcal{S} \models \text{left}(l_1, l_2) & \quad \Leftrightarrow l_1 \prec_{\text{left}} l_2 \\ \mathcal{R}, \mathcal{S} \models \text{lonr}(c_1, c_2, d) & \quad \Leftrightarrow \mathcal{D}_0(c_1, c_2) = d \\ \mathcal{R}, \mathcal{S} \models \neg\phi & \quad \Leftrightarrow \mathcal{R}, \mathcal{S} \not\models \phi \\ \mathcal{R}, \mathcal{S} \models \phi_1 \rightarrow \phi_2 & \quad \Leftrightarrow \mathcal{R}, \mathcal{S} \models \phi_2 \text{ if } \mathcal{R}, \mathcal{S} \models \phi_1 \\ \mathcal{R}, \mathcal{S} \models \bigcirc\phi & \quad \Leftrightarrow T > 1 \text{ and } \mathcal{R}, \mathcal{S}_1 \models \phi \\ \mathcal{R}, \mathcal{S} \models \square\phi & \quad \Leftrightarrow \forall 0 \leq i < T, \mathcal{R}, \mathcal{S}_i \models \phi \\ \mathcal{R}, \mathcal{S} \models \mathbb{F} & \quad \Leftrightarrow T = 1 \end{aligned}$$

3.2 Rules of Highway TSL

Realistic scenarios obey both physics rules and traffic rules. In this subsection, we outline these rules and auxiliary predicates we use in the form of TSL formulae, as shown in Table 1. Traffic rules can be added or subtracted according to the laws of different regions.

Physics Rules

- PR1 The longitudinal positional relationships are symmetric.
- PR2 The longitudinal positional relationships are transitive.
- PR3 Longitudinal positional relationships must not be contradictory.
- PR4 The longitudinal positional relationships must change continuously.
- PR5 The occupied lanes by a vehicle must be one or more adjacent lanes. We utilize an auxiliary predicate $\text{cleft}/2$ (AP1) here for simplicity, representing one lane being to the left of another lane but not necessarily adjacent.

PR6 Each vehicle always occupies at least one lane.

PR7 The occupied lanes must change continuously. That is, only one lane can be added to or removed from the set of occupied lanes on each step.

Traffic Rules

TR1 Vehicles are not allowed to occupy three or more lanes simultaneously.

TR2 Two vehicles cannot drive side by side in the same lane.

A realistic scenario and its tails (because a tail of a realistic scenario is also a realistic scenario) must satisfy all the aforementioned rules. Consequently, by utilizing these rules, we can validate the plausibility of a scenario, rendering the reasoning of scenarios under TSL executable.

4 Urban Traffic Scenario Logic

In this section, we will introduce the extension of TSL to general urban traffic scenarios, including vehicles traveling on structured roads without pedestrians.

4.1 Formal Representation of Urban Road Network

A structured urban road network consists of lanes. When considering two lanes, their relationships can be categorized as follows:

1. They belong to the same road, in which case the scenario is identical to that described in Section 3;
2. They have an intersection point, but are not connected at this point (Figure 1a);
3. They are connected at a connection point, where vehicles can switch to another lane (Figure 1b);
4. They have an overlap segment in which vehicles travel in the same direction. In this case, we can merge the overlap segment into a single lane;
5. They have an overlap segment in which vehicles travel in opposite directions (Figure 1c);
6. They are unrelated.

All we need is to extend the TSL for intersections, connections, and overlaps in opposite directions. The rest of the cases can reuse Highway TSL.

1. Intersection

Without loss of generality, we only consider the situation where two lanes intersect once. For multiple intersections, we can simply apply it repeatedly to each intersection.

Denote the intersection point as p^x . Then, for any vehicle c on these two intersecting lanes, there are essentially three positional relationships with respect to this intersection point: yet to pass, currently passing, and already passed. We use the predicate $\text{lonr}/3$ to represent these relationships, corresponding to the predicate $\text{lonr}/3$ for vehicles’ relationships. The same predicate can also be used for the connection points and endpoints of overlap segments below.

Rule	TSL Formula
PR1	$\Box(\forall c_1 \forall c_2 \text{lonr}(c_1, c_2, \text{ahead}) \rightarrow \text{lonr}(c_2, c_1, \text{behind})),$ $\Box(\forall c_1 \forall c_2 \text{lonr}(c_1, c_2, \text{behind}) \rightarrow \text{lonr}(c_2, c_1, \text{ahead})),$ $\Box(\forall c_1 \forall c_2 \text{lonr}(c_1, c_2, \text{cover}) \rightarrow \text{lonr}(c_2, c_1, \text{cover})).$
PR2	$\Box(\forall c_1 \forall c_2 \forall c_3 \text{lonr}(c_1, c_2, \text{ahead}) \wedge \text{lonr}(c_2, c_3, \text{ahead}) \rightarrow \text{lonr}(c_1, c_3, \text{ahead})),$ $\Box(\forall c_1 \forall c_2 \forall c_3 \text{lonr}(c_1, c_2, \text{behind}) \wedge \text{lonr}(c_2, c_3, \text{behind}) \rightarrow \text{lonr}(c_1, c_3, \text{behind})).$
PR3	$\Box(\forall c_1 \forall c_2 \forall c_3 \neg(\text{lonr}(c_1, c_2, \text{ahead}) \wedge \text{lonr}(c_2, c_3, \text{cover}) \wedge \text{lonr}(c_3, c_1, \text{ahead}))).$
PR4	$\mathbb{F} \vee \Box(\forall c_1 \forall c_2 \text{lonr}(c_1, c_2, \text{ahead}) \rightarrow \mathbb{O}(\text{lonr}(c_1, c_2, \text{ahead}) \vee \text{lonr}(c_1, c_2, \text{cover}))),$ $\mathbb{F} \vee \Box(\forall c_1 \forall c_2 \text{lonr}(c_1, c_2, \text{behind}) \rightarrow \mathbb{O}(\text{lonr}(c_1, c_2, \text{behind}) \vee \text{lonr}(c_1, c_2, \text{cover}))).$
PR5	$\Box(\forall c \forall l_1 \forall l_2 \forall l_3 \text{on}(c, l_1) \wedge \text{on}(c, l_2) \wedge \text{cleft}(l_1, l_3) \wedge \text{cleft}(l_3, l_2) \rightarrow \text{on}(c, l_3)).$
PR6	$\Box(\forall c \exists l \text{on}(c, l)).$
PR7	$\mathbb{F} \vee \Box(\forall c \forall l_1 \forall l_2 \neg(\text{on}(c, l_1) \wedge \text{on}(c, l_2) \wedge \neg(l_1 = l_2) \wedge \neg \mathbb{O} \text{on}(c, l_1) \wedge \neg \mathbb{O} \text{on}(c, l_2))),$ $\mathbb{F} \vee \Box(\forall c \forall l_1 \forall l_2 \neg(\neg \text{on}(c, l_1) \wedge \neg \text{on}(c, l_2) \wedge \neg(l_1 = l_2) \wedge \mathbb{O} \text{on}(c, l_1) \wedge \mathbb{O} \text{on}(c, l_2))),$ $\mathbb{F} \vee \Box(\forall c \forall l_1 \forall l_2 \neg(\text{on}(c, l_1) \wedge \neg \text{on}(c, l_2) \wedge \neg(l_1 = l_2) \wedge \neg \mathbb{O} \text{on}(c, l_1) \wedge \mathbb{O} \text{on}(c, l_2))).$
TR1	$\Box(\forall c \forall l_1 \forall l_2 \forall l_3 \neg(\text{on}(c, l_1) \wedge \text{on}(c, l_2) \wedge \text{on}(c, l_3) \wedge \neg(l_1 = l_2) \wedge \neg(l_1 = l_3) \wedge \neg(l_2 = l_3))).$
TR2	$\Box(\forall c_1 \forall c_2 \forall l \neg(\text{on}(c_1, l) \wedge \text{on}(c_2, l) \wedge \text{lonr}(c_1, c_2, \text{cover}) \wedge \neg(c_1 = c_2))).$
AP1	$\forall l_1 \forall l_2 \text{left}(l_1, l_2) \rightarrow \text{cleft}(l_1, l_2), \forall l_1 \forall l_2 \forall l_3 \text{left}(l_2, l_3) \wedge \text{cleft}(l_1, l_2) \rightarrow \text{cleft}(l_1, l_3).$

Table 1: The rules and auxiliary predicates used in Highway TSL

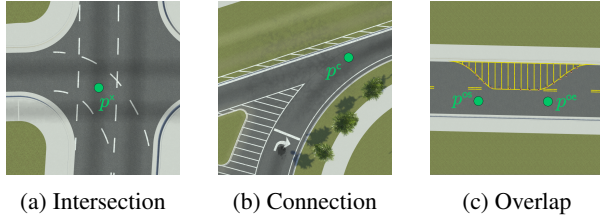


Figure 1: Special points on urban road network

2. Connection

Two or more lanes can connect at their endpoints. We denote the connection points as p^c . Depending on the direction of travel of vehicles on the lanes, there exists a successor relationship \succ_c between the connection points and the lanes.

3. Overlap in opposite directions

When examining the overlap segment between points p^{os} and p^{oe} , as shown in Figure 1c, we can designate one of the two overlapping lanes' direction as the reference direction, i.e., from p^{os} to p^{oe} . Another longitudinal positional relationship, denoted as $\text{lonro}/3$, exists between the vehicles in the overlap segment, and:

- For the lane in the reference direction, the relationships $\text{lonro}/3$ between vehicles are the same as $\text{lonr}/3$;
- For the lane in the opposite direction, the relationships $\text{lonro}/3$ between vehicles are opposite to $\text{lonr}/3$;
- For two vehicles belonging to different directions, the relationship $\text{lonro}/3$ between them can be arbitrary, but only monotonic changes are allowed.

Besides, when there are multiple points (including intersection points, connection points, and endpoints of overlap

segments) on a lane l , there also exists a successor relationship \succ_p^l between points according to the direction of vehicle travel on the lane.

Based on the above analysis, we further provide a formal definition of the road network:

Definition 8 (Road Network). *The road network is a non-tuple $\mathcal{M} = (\mathbb{R}, \mathbb{P}^x, \mathbb{P}^c, \mathbb{P}^{\text{os}}, \mathbb{P}^{\text{oe}}, \mathbb{S}^p, \mathbb{S}^c, \mathbb{O}, \mathbb{T})$, where:*

- $\mathbb{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots\}$ is the set of all roads;
- $\mathbb{P}^x, \mathbb{P}^c, \mathbb{P}^{\text{os}}, \mathbb{P}^{\text{oe}}$ are sets of points p^x, p^c, p^{os} , and p^{oe} respectively;
- $\mathbb{S}^p \subseteq \mathbb{L} \times \mathbb{P} \times \mathbb{P}$ defines the successor relationship \succ_p^l for points on each lane l , where $\mathbb{L} = \bigcup_i \mathbb{L}_i$ is the set of all lanes, and $\mathbb{P} = \mathbb{P}^x \cup \mathbb{P}^c \cup \mathbb{P}^{\text{os}} \cup \mathbb{P}^{\text{oe}}$ is the set of all points;
- $\mathbb{S}^c \subseteq \mathbb{P}^c \times \mathbb{L}$ defines the successor lane for connection points, denoted as the relation \succ_c ;
- $\mathbb{O} \subseteq \mathbb{P}^{\text{os}} \times \mathbb{P}^{\text{oe}}$ represents the pairs $(p^{\text{os}}, p^{\text{oe}})$ for each overlap segment.
- $\mathbb{T} \subseteq \mathbb{L} \times \mathbb{P}$ defines the affiliation relationship between points and lanes.

We proposed an example of this abstraction in the Technical Appendix.

Theorem 1. *For a structured 2D OpenDRIVE road network, there exists a representation in the form defined by Definition 8.*

Proof. See the discussion above. \square

4.2 Abstract Model

In this subsection, we provide the syntax and semantics of TSL for urban traffic scenarios.

Definition 9 (Scene). *A (urban) scene is a quadruple $\mathfrak{s} = (\mathcal{L}, \mathcal{D}, \mathcal{P}, \mathcal{Q})$, where:*

- $\mathcal{L} : \mathbb{C} \mapsto \mathcal{P}(\mathbb{L})$ is a mapping from the set of vehicles to the power set of the set of lanes, assigning to each vehicle a set of lanes to which it occupies;
- $\mathcal{D} : \mathbb{C} \times \mathbb{C} \mapsto \mathbb{D}$ is a mapping from pairs of vehicles to the set of longitudinal positional relationships $\mathbb{D} = \{\text{ahead, cover, behind, none}\}$, where none indicates that there is no relationship between two vehicles, which occurs when the two vehicles belong to different roads;
- $\mathcal{P} : \mathbb{C} \times \mathbb{P} \mapsto \mathbb{D}$ is a mapping from pairs of vehicles and points to the set of longitudinal positional relationships, as explained in Section 4.1;
- $\mathcal{Q} : \mathbb{C} \times \mathbb{C} \mapsto \mathbb{D}$ is a mapping to define the temporary longitudinal positional relationship lonro/3 on the overlap segments.

Definition 10 (Scenario). A (urban) scenario is a sextuple $\mathcal{S} = (\mathbb{C}, \mathbb{L}, \mathbb{D}, \mathbb{P}, T, \langle \mathfrak{s}_0, \mathfrak{s}_1, \dots, \mathfrak{s}_{T-1} \rangle)$, where:

- $\mathbb{C}, \mathbb{L}, \mathbb{D}$, and \mathbb{P} have the meanings as described above;
- $T \geq 1$ is the length of the scenario;
- $\mathfrak{s}_i = (\mathcal{L}_i, \mathcal{D}_i, \mathcal{P}_i, \mathcal{Q}_i)$ is the i -th scene that occurs in the scenario in time sequence.

The definition of the i -tail of a scenario is similar to Definition 5 and will not be repeated.

Theorem 2. For a pedestrian-free traffic scenario on road network \mathcal{M} , there exists a representation in the form defined by Definition 10.

Proof. Since the longitudinal relationship as defined by Definition 2 encompasses all situations, the mappings \mathcal{L} , \mathcal{D} , \mathcal{P} , and \mathcal{Q} are guaranteed to exist and be unique for each scene. Consequently, we establish the existence of a representation \mathcal{S} for each pedestrian-free traffic scenario. \square

Definition 11 (Syntax of Urban TSL). A formula of urban TSL can be recursively defined as follows:

$$\begin{aligned} \phi ::= & \top \mid \text{on}(c, l) \mid \text{pon}(p, l) \mid \text{left}(l_1, l_2) \mid \text{lonr}(c_1, c_2, d) \\ & \mid \text{lonpr}(c, p, d) \mid \text{lonro}(c_1, c_2, d) \mid \text{succp}(l, p_1, p_2) \\ \mid & \text{succl}(p, l) \mid \text{belong}(l, \mathcal{R}) \mid \text{overlap}(p_1, p_2) \mid \text{p}_x(p) \mid \text{p}_c(p) \\ \mid & \text{p}_{\text{os}}(p) \mid \text{p}_{\text{oe}}(p) \mid c_1 = c_2 \mid l_1 = l_2 \mid d_1 = d_2 \mid p_1 = p_2 \\ \mid & \mathcal{R}_1 = \mathcal{R}_2 \mid \neg\phi \mid \phi_1 \rightarrow \phi_2 \mid \bigcirc\phi \mid \square\phi \mid \mathbb{F} \\ \mid & \forall c \phi \mid \forall l \phi \mid \forall d \phi \mid \forall p \phi \mid \forall \mathcal{R} \phi \end{aligned}$$

Definition 12 (The Semantics of Urban TSL). Given the road network \mathcal{M} and the logical scenario \mathcal{S} , the satisfaction of formulas defined in Definition 11 follows the following semantics:

$$\begin{aligned} \mathcal{M}, \mathcal{S} \models \top & \quad \text{for all } \mathcal{M}, \mathcal{S} \\ \mathcal{M}, \mathcal{S} \models \text{on}(c, l) & \Leftrightarrow l \in \mathcal{L}_0(c) \\ \mathcal{M}, \mathcal{S} \models \text{pon}(p, l) & \Leftrightarrow (l, p) \in \mathbb{T} \\ \mathcal{M}, \mathcal{S} \models \text{left}(l_1, l_2) & \Leftrightarrow l_1 \prec_{\text{left}} l_2 \\ \mathcal{M}, \mathcal{S} \models \text{lonr}(c_1, c_2, d) & \Leftrightarrow \mathcal{D}_0(c_1, c_2) = d \\ \mathcal{M}, \mathcal{S} \models \text{lonpr}(c, p, d) & \Leftrightarrow \mathcal{P}_0(c, p) = d \\ \mathcal{M}, \mathcal{S} \models \text{lonro}(c_1, c_2, d) & \Leftrightarrow \mathcal{Q}_0(c_1, c_2) = d \\ \mathcal{M}, \mathcal{S} \models \text{succp}(l, p_1, p_2) & \Leftrightarrow (l, p_1, p_2) \in \mathbb{S}^{\text{P}} \end{aligned}$$

$$\begin{aligned} \mathcal{M}, \mathcal{S} \models \text{succl}(p, l) & \Leftrightarrow (p, l) \in \mathbb{S}^{\text{C}} \\ \mathcal{M}, \mathcal{S} \models \text{belong}(l, \mathcal{R}) & \Leftrightarrow l \in \mathcal{R} \\ \mathcal{M}, \mathcal{S} \models \text{overlap}(p_1, p_2) & \Leftrightarrow (p_1, p_2) \in \mathbb{O} \\ \mathcal{M}, \mathcal{S} \models \text{p}_x(p) & \Leftrightarrow p \in \mathbb{P}^{\text{X}} \\ \mathcal{M}, \mathcal{S} \models \text{p}_c(p) & \Leftrightarrow p \in \mathbb{P}^{\text{C}} \\ \mathcal{M}, \mathcal{S} \models \text{p}_{\text{os}}(p) & \Leftrightarrow p \in \mathbb{P}^{\text{OS}} \\ \mathcal{M}, \mathcal{S} \models \text{p}_{\text{oe}}(p) & \Leftrightarrow p \in \mathbb{P}^{\text{OE}} \\ \mathcal{M}, \mathcal{S} \models \neg\phi & \Leftrightarrow \mathcal{M}, \mathcal{S} \not\models \phi \\ \mathcal{M}, \mathcal{S} \models \phi_1 \rightarrow \phi_2 & \Leftrightarrow \mathcal{M}, \mathcal{S} \models \phi_2 \text{ if } \mathcal{M}, \mathcal{S} \models \phi_1 \\ \mathcal{M}, \mathcal{S} \models \bigcirc\phi & \Leftrightarrow T > 1 \text{ and } \mathcal{M}, \mathcal{S}_1 \models \phi \\ \mathcal{M}, \mathcal{S} \models \square\phi & \Leftrightarrow \forall 0 \leq i < T, \mathcal{M}, \mathcal{S}_i \models \phi \\ \mathcal{M}, \mathcal{S} \models \mathbb{F} & \Leftrightarrow T = 1 \end{aligned}$$

4.3 Rules of Urban TSL

For urban traffic scenarios, the rules described in Section 3.2 regarding vehicles within the same road still apply and will not be reiterated here. Additionally, the following rules are added to handle situations across roads, whose formulae are shown in Table 2.

- PR8 A vehicle can only be located on one road. We introduce an auxiliary predicate cbelong/2 (AP2) to represent the belonging relationship between a vehicle and a road.
- PR9 The longitudinal positional relationship between a vehicle and the points on the same lane can only change monotonically.
- PR10 A unique longitudinal relationship, which is not none, exists between a vehicle and the points on the lane it occupies.
- PR11 Only one vehicle is allowed to cover a point.
- PR12 When a vehicle passes through a connection point, it will enter one successor lane of that connection point.
- PR13 Rules for overlap segments following the discussion in Section 4.1. Auxiliary predicates fwdover/3 (AP3) and rvsover/3 (AP4) are imported for this rule to indicate the vehicle driving along/against the reference direction on an overlap segment, respectively.
- PR14 All longitudinal positional relationships, including those between vehicles and vehicles, vehicles and points, and those on overlap segments, follow similar rules as presented in Section 3.2, which will not be reiterated here.

By adding the above rules, we can validate the plausibility of an urban traffic scenario.

5 Application

TSL is a useful tool for modeling and reasoning about traffic scenarios. It can generate various traffic scenarios, which can be used to create a library of test cases for comprehensive testing of autonomous vehicles. Additionally, TSL can identify potential failure risks while developing autonomous driving strategies. Its unique spatial representation makes it

Rule	TSL Formula
PR8	$\Box(\forall c \forall \mathcal{R}_1 \forall \mathcal{R}_2 \neg(\text{cbelong}(c, \mathcal{R}_1) \wedge \text{cbelong}(c, \mathcal{R}_2) \wedge \neg(\mathcal{R}_1 = \mathcal{R}_2)))$.
PR9	$\mathbb{F} \vee \Box(\forall c \forall p \text{lonpr}(c, p, \text{behind}) \rightarrow \bigcirc(\text{lonpr}(c, p, \text{behind}) \vee \text{lonpr}(c, p, \text{cover}) \vee \text{lonpr}(c, p, \text{none}))),$ $\mathbb{F} \vee \Box(\forall c \forall p \text{lonpr}(c, p, \text{cover}) \rightarrow \bigcirc(\text{lonpr}(c, p, \text{cover}) \vee \text{lonpr}(c, p, \text{ahead}) \vee \text{lonpr}(c, p, \text{none}))),$ $\mathbb{F} \vee \Box(\forall c \forall p \text{lonpr}(c, p, \text{ahead}) \rightarrow \bigcirc(\text{lonpr}(c, p, \text{ahead}) \vee \text{lonpr}(c, p, \text{none}))).$
PR10	$\Box(\forall c \forall p \forall d_1 \forall d_2 \neg(\text{lonpr}(c, p, d_1) \wedge \text{lonpr}(c, p, d_2) \wedge \neg(d_1 = d_2))),$ $\Box(\forall c \forall l \forall p \neg(\text{on}(c, l) \wedge \text{pon}(p, l) \wedge \text{lonpr}(c, p, \text{none}))).$
PR11	$\Box(\forall c_1 \forall c_2 \forall p \neg(\text{lonpr}(c_1, p, \text{cover}) \wedge \text{lonpr}(c_2, p, \text{cover}) \wedge \neg(c_1 = c_2))).$
PR12	$\mathbb{F} \vee \Box(\forall c \forall p \forall l_1 \text{lonpr}(c, p, \text{cover}) \wedge \text{p}_c(p) \wedge \text{on}(c, l_1) \rightarrow$ $\bigcirc((\text{lonpr}(c, p, \text{cover}) \wedge \text{on}(c, l_1)) \vee \exists l_2 (\text{lonpr}(c, p, \text{ahead}) \wedge \text{on}(c, l_2) \wedge \text{succ}(p, l_2))))).$
PR13	$\Box(\forall c_1 \forall c_2 \forall p_1 \forall p_2 \forall d \text{fwdover}(c_1, p_1, p_2) \wedge \text{fwdover}(c_2, p_1, p_2) \wedge \text{lonr}(c_1, c_2, d) \rightarrow \text{lonro}(c_1, c_2, d)),$ $\Box(\forall c_1 \forall c_2 \forall p_1 \forall p_2 \text{rvsover}(c_1, p_1, p_2) \wedge \text{rvsover}(c_2, p_1, p_2) \wedge \text{lonr}(c_1, c_2, \text{ahead}) \rightarrow \text{lonro}(c_1, c_2, \text{behind})),$ $\Box(\forall c_1 \forall c_2 \forall p_1 \forall p_2 \text{rvsover}(c_1, p_1, p_2) \wedge \text{rvsover}(c_2, p_1, p_2) \wedge \text{lonr}(c_1, c_2, \text{cover}) \rightarrow \text{lonro}(c_1, c_2, \text{cover})),$ $\Box(\forall c_1 \forall c_2 \forall p_1 \forall p_2 \text{rvsover}(c_1, p_1, p_2) \wedge \text{rvsover}(c_2, p_1, p_2) \wedge \text{lonr}(c_1, c_2, \text{behind}) \rightarrow \text{lonro}(c_1, c_2, \text{ahead})),$ $\Box(\forall c_1 \forall c_2 \forall p_1 \forall p_2 \text{fwdover}(c_1, p_1, p_2) \wedge \text{rvsover}(c_2, p_1, p_2) \rightarrow (\exists d \text{lonro}(c_1, c_2, d) \wedge \neg(d = \text{none}))).$
AP2	$\forall c \forall l \forall \mathcal{R} \text{belong}(l, \mathcal{R}) \wedge \text{on}(c, l) \rightarrow \text{cbelong}(c, \mathcal{R})$.
AP3	$\forall c \forall p_1 \forall p_2 \forall l \text{lonpr}(c, p_1, \text{ahead}) \wedge \text{lonpr}(c, p_2, \text{behind}) \wedge \text{on}(c, l) \wedge \text{pon}(p_1, l) \wedge \text{pon}(p_2, l) \wedge \text{overlap}(p_1, p_2) \rightarrow \text{fwdover}(c, p_1, p_2)$.
AP4	$\forall c \forall p_1 \forall p_2 \forall l \text{lonpr}(c, p_1, \text{behind}) \wedge \text{lonpr}(c, p_2, \text{ahead}) \wedge \text{on}(c, l) \wedge \text{pon}(p_1, l) \wedge \text{pon}(p_2, l) \wedge \text{overlap}(p_1, p_2) \rightarrow \text{rvsover}(c, p_1, p_2)$.

Table 2: The additional rules and auxiliary predicates used in Urban TSL

well-suited for these applications, which require both the comprehensiveness and generalization of scenarios.

In this section, we start by introducing the reasoning of TSL, which we implemented using Answer Set Programming (ASP) with temporal logic extensions. Following that, we illustrate several examples of TSL for scenario generation to demonstrate its effectiveness in handling miscellaneous urban scenarios.

5.1 Reasoning of TSL

Given an initial scene s_0 , the set of vehicles \mathbb{C} , a finite horizon T and a road network \mathcal{M} , the reasoning of TSL is a kind of model expansion, i.e. to find all logical scenarios $\mathcal{S} = (\mathbb{C}, \mathbb{L}, \mathbb{D}, \mathbb{P}, T, \langle \mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_{T-1} \rangle)$ such that $\mathcal{M}, \mathcal{S} \models \Phi$, where Φ is the set of rules explained in Section 3 and Section 4.

We use Telingo (Cabalar et al. 2019) to implement the reasoning of TSL. Telingo is a solver for temporal logic programs based on ASP and the well-known Clingo (Gebser et al. 2019) input language. Telingo extends the ASP language by importing temporal modal operators under the semantics of the Linear Temporal Logic (\mathcal{LTL}) (Cabalar et al. 2018; Pnueli 1977).

We use aggregates (Son and Pontelli 2007) instead of actions and transitions in the logic program we used for TSL reasoning to derive different scenarios. All rules introduced in Section 3 and Section 4 are rewritten into corresponding ASP rules or constraints. The road network and initial scene are described by facts. For the entire implementation of our reasoning programs, please refer to our source code.

5.2 Examples of Scenario Generation

In this subsection, we provide several examples of TSL on different road layouts to provide an overview of TSL's fea-

tures in scenario reasoning.

Example 1 (Highway overtake). *Figure 2a shows overtake scenarios on a uni-direction two-lane highway. The shortest scenarios we get are shown in Figure 2a. In the first scenario, c_1 (the blue one) overtakes c_2 (the red one) by lane change. In the second and third scenarios, the two cars change lanes simultaneously, and then one of the cars gives up midway through the lane change. In the fourth scenario, c_2 changes lanes to the left to make way for c_1 .*

Example 2 (Intersection). *When two cars pass an intersection point, TSL gives two different scenarios, as shown in Figure 2b. Please note that the two lanes are not connected at the intersection point, so a car cannot switch to the other lane.*

Example 3 (Connecting roads). *Figure 2c illustrates a scenario where a car passes through a connection point and may drive into any of the connecting lanes.*

Example 4 (A vehicle passes two intersection points). *When a vehicle passes through two intersection points successively, there are two possible scenarios since the length of the vehicle and the distance between the intersection points are unknown, as shown in Figure 2d. A long truck may cover both intersection points simultaneously, while a short car can be in between.*

Example 5 (Overtake on the opposite side). *TSL can use overlap to create overtaking on opposite scenarios, as shown in Figure 2e. It contains two roads: one road has two lanes, l_1 and l_2 , running from left to right, and the other road has one lane, l_3 , running from right to left, with l_3 overlapping with l_2 . With no lane changes allowed for c_2 and c_3 , two scenarios are provided by TSL for c_1 to overtake c_2 : waiting for c_3 to pass before overtaking, or overtaking before c_3 passes.*

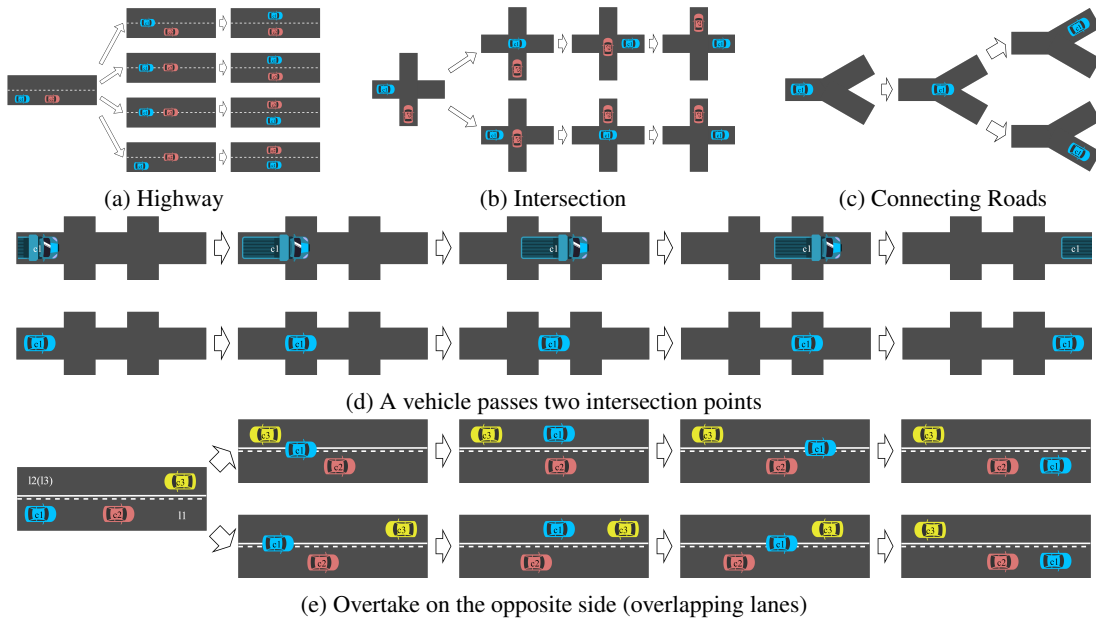


Figure 2: Examples of generated scenarios with TSL

The examples above show how TSL works for scenario generation. By combining these primitives, TSL can manage different urban road networks and create a variety of traffic scenarios. Please refer to our source code for the detailed implementation of these examples. Theorem 2 shows that TSL is not limited to such simple scenarios, but can theoretically handle arbitrarily complex scenarios. More complex examples are provided in the Technical Appendix.

5.3 Use TSL for Test Scenario Generation

The scenarios generated can be described using a scenario description language and simulated as test cases for autonomous vehicles. We use OpenSCENARIO DSL (OSC)¹ as an example. It is a widely accepted industry standard proposed by ASAM for describing abstract scenarios, particularly for autonomous systems testing uses. For the syntax and semantics of OSC, please refer to its document.

A TSL scenario can be easily translated into OSC with the following steps (OSC keywords shown in `typewriter` font):

1. Create a top-level `scenario`;
2. Declare all points introduced in Section 4.1 with the `position_3d` type;
3. Add a `do serial` block to the top-level scenario. Then, for each scene in the TSL scenario, add a corresponding `parallel` sub-block;
4. In each `parallel` sub-block, describe the `drive()` action of every vehicle in this scene. The OSC provides the `position` modifier for the longitudinal position (relative to another vehicle or point declared in step 2) and `lateral` for the lateral position (relative to lane boundaries).

¹<https://www.asam.net/standards/detail/openscenario-dsl/>

This translation is straightforward and can be automatized. A complete example is provided in the Technical Appendix. An OSC file can be further evaluated by a simulator to serve as a test case (Zhang et al. 2021; Bagschik, Menzel, and Maurer 2018) or used to find possible malfunctions (Wang et al. 2023). Specific usages of OSC files are beyond the scope of this paper.

5.4 Discussion

TSL has numerous potential applications beyond test scenario generation. For example, it can accomplish autonomous driving decision-making in a more interpretable manner and seamlessly integrate predictions for other vehicles into its reasoning process. A demonstration is provided in the Technical Appendix. It can also be used to verify the safety of control sequences for autonomous vehicles, which is a focus of our future research.

New predicates and constraints can also be introduced to extend the semantics of TSL for broader applications. For example, predicates representing vehicle types can be added, and additional traffic rules specific to certain vehicle types can be incorporated.

6 Conclusion

In this paper, we propose a novel spatial-temporal logic, Traffic Scenario Logic, for modeling and reasoning of urban pedestrian-free traffic scenarios. TSL is compatible with OpenDRIVE and has an efficient and complete spatial representation, making it well-suited for developing a human-interpretable test scenario library for autonomous vehicle safety assessment. Our future work will involve incorporating pedestrians into scenarios and exploring additional applications of TSL.

Acknowledgments

This work was partially supported by National Key R&D Program of China No. 2023YFB4704500, Hunan Province Major Scientific and Technological Project No. 2024QK2001, and was partially supported by Sangfor Technologies Co., Ltd.

References

- Bagschik, G.; Menzel, T.; and Maurer, M. 2018. Ontology based scene creation for the development of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 1813–1820. IEEE.
- Bishopink, C.; and Olderog, E.-R. 2022. Spatial and timing properties in highway traffic. In *International Colloquium on Theoretical Aspects of Computing*, 114–131. Springer.
- Cabalar, P.; Kaminski, R.; Morkisch, P.; and Schaub, T. 2019. *telingo=ASP+time*. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, 256–269. Springer.
- Cabalar, P.; Kaminski, R.; Schaub, T.; and Schuhmann, A. 2018. Temporal answer set programming on finite traces. *Theory and Practice of Logic Programming*, 18(3-4): 406–420.
- Dupuis, M.; Strobl, M.; and Grezlikowski, H. 2010. Open-drive 2010 and beyond—status and future of the de facto standard for the description of road networks. In *Proc. of the Driving Simulation Conference Europe*, 231–242.
- Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2019. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming*, 19(1): 27–82.
- Hilscher, M.; Linker, S.; and Olderog, E.-R. 2013. Proving safety of traffic manoeuvres on country roads. *Theories of Programming and Formal Methods: Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*, 196–212.
- Hilscher, M.; Linker, S.; Olderog, E.-R.; and Ravn, A. P. 2011. An abstract model for proving safety of multi-lane traffic manoeuvres. In *International Conference on Formal Engineering Methods*, 404–419. Springer.
- Hilscher, M.; and Schwammberger, M. 2016. An abstract model for proving safety of autonomous urban traffic. In *International Colloquium on Theoretical Aspects of Computing*, 274–292. Springer.
- Kontchakov, R.; Kurucz, A.; Wolter, F.; and Zakharyashev, M. 2007. Spatial logic+ temporal logic=? *Handbook of spatial logics*, 497–564.
- Li, Y.; Tao, J.; and Wotawa, F. 2020. Ontology-based test generation for automated and autonomous driving functions. *Information and software technology*, 117: 106200.
- Loos, S. M.; and Platzer, A. 2011. Safe intersections: At the crossing of hybrid systems and verification. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1181–1186. IEEE.
- Moszkowski, B. 1982. *A temporal logic for multi-level reasoning about hardware*. Department of Computer Science, Stanford University.
- Pnueli, A. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, 46–57. iee.
- Schwammberger, M.; and Alves, G. V. 2021. Extending urban multi-lane spatial logic to formalise road junction rules. *arXiv preprint arXiv:2110.12583*.
- Son, T. C.; and Pontelli, E. 2007. A constructive semantic characterization of aggregates in answer set programming. *Theory and Practice of Logic Programming*, 7(3): 355–375.
- Wang, R.; Xu, Y.; Peng, J.; and Ji, J. 2023. A2CoST: An ASP-based Avoidable Collision Scenario Testbench for Autonomous Vehicles. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning*, 690–699.
- Werling, M.; Ziegler, J.; Kammel, S.; and Thrun, S. 2010. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE international conference on robotics and automation*, 987–993. IEEE.
- Xu, B.; and Li, Q. 2016. A spatial logic for modeling and verification of collision-free control of vehicles. In *2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS)*, 33–42. IEEE.
- Xu, B.; Li, Q.; Guo, T.; and Du, D. 2019. A scenario-based approach for formal modelling and verification of safety properties in automated driving. *IEEE Access*, 7: 140566–140587.
- Yao, S.; and Li, Q. 2023. A Hierarchical Spatial Logic for Knowledge Sharing and Fusion in Intelligent Connected Vehicle Cooperation. In *International Symposium on Theoretical Aspects of Software Engineering*, 183–201. Springer.
- Zhang, X.; Khastgir, S.; Asgari, H.; and Jennings, P. 2021. Test framework for automatic test case generation and execution aimed at developing trustworthy avs from both verifiability and certifiability aspects. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 312–319. IEEE.