# ABSTRACT SCENE GRAPHS

## SPECIFYING AND MONITORING SPATIAL PROPERTIES OF AUTOMATED DRIVING FUNCTIONS

**Ishan Saxena**[a]**,** Bernd Westphal[a], Martin Fränzle[b]

a: German Aerospace Center (DLR) e.V., Oldenburg, Germany b: Carl von Ossietzky University of Oldenburg, Germany

Seventh International Workshop on Formal Methods for Autonomous Systems (FMAS 2025)

EdgeAI-Trust

Carl von Ossietzky Universität Oldenburg

DLR

# Outline

- Motivation
- Research Questions
- Abstract Scene Graphs (ASGs)
  - Syntax
  - Formalization of spatial properties
- Runtime Monitoring using ASGs
- Future Work

# Motivation

- Automatic validation of Automated Driving Functions is desired to ensure safety during their operation
- Requires formalized, machine-readable system properties

Credit: https://www.futureelectronics.com/blog/article/understanding-autonomous-vehicle-safety/

# Motivation

## Example spatial property

- Automated vehicle should stop at a safe distance if there is an obstacle in front.

# Motivation

## Example spatial property

- Automated vehicle should stop at a safe distance if there is an obstacle in front. But here?

# Motivation

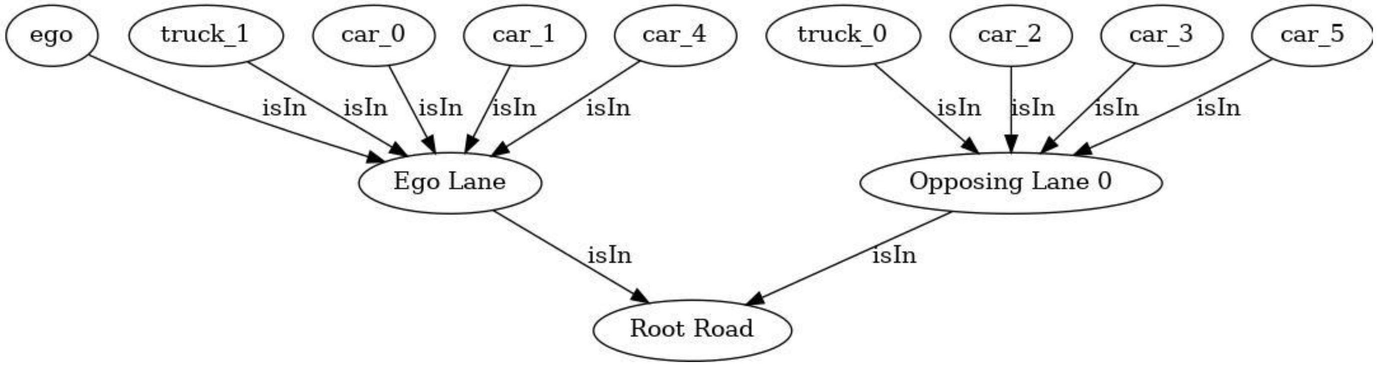
- Formalization of spatial properties in the automotive domain is challenging
- Because, such spatial properties are:
  - Textual
    - Traffic Rules and Regulations
  - Complex
    - Legal terminology
    - Multi-stakeholder knowledge
  - Vague
    - Rely on human intuition

# Motivation

- Certain visual formalisms in literature implicitly formalize spatial properties
  - E.g., using comparison of geometric coordinates of the objects
    - "vehicle *ego* is in lane *l* " is represented as: $l.yleft < ego.y < l.yright$
  - Limits their usage for use cases such as runtime monitoring
    - To check if vehicle is in lane, y-coordinates of lane are required

# Scene Graphs

- Scene Graphs: represent entities present in a scene and their spatial relationships explicitly




Source: https://github.com/less-lab-uva/carla_scene_graphs

- In the automotive domain, Scene Graphs have been used for:
  - Monitoring of safety properties [12]
  - Scene Understanding and Risk Assessment [8,9,10]

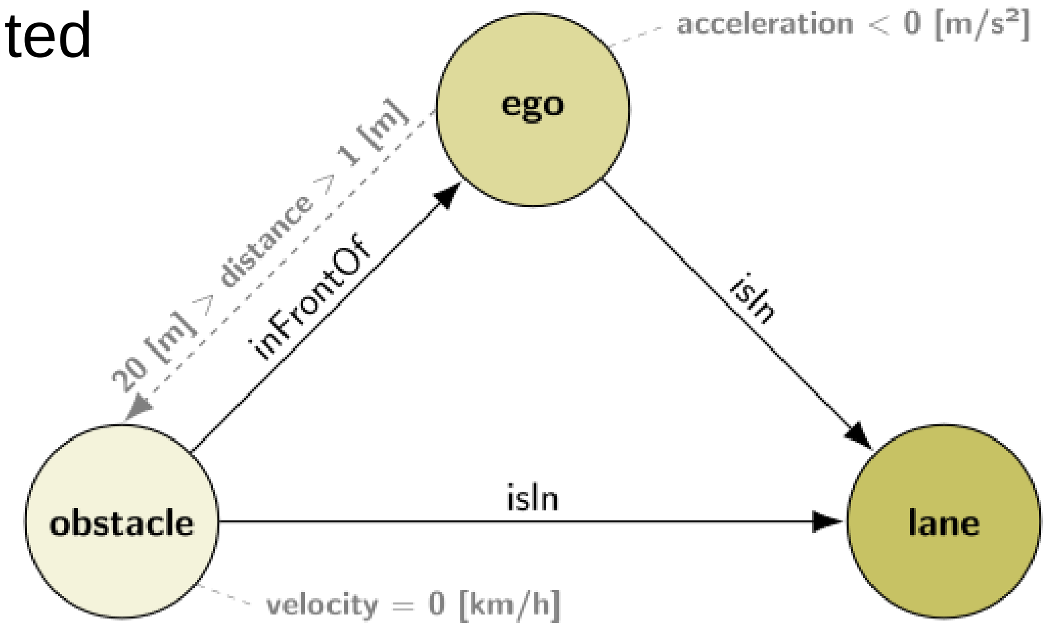
# Research Questions

**Q1:** Can **Scene Graphs** be used to formalize spatial system properties occuring in automotive domain?

**Q2:** How can **Runtime Monitoring** be performed for spatial system properties formalized using **Scene Graphs**?

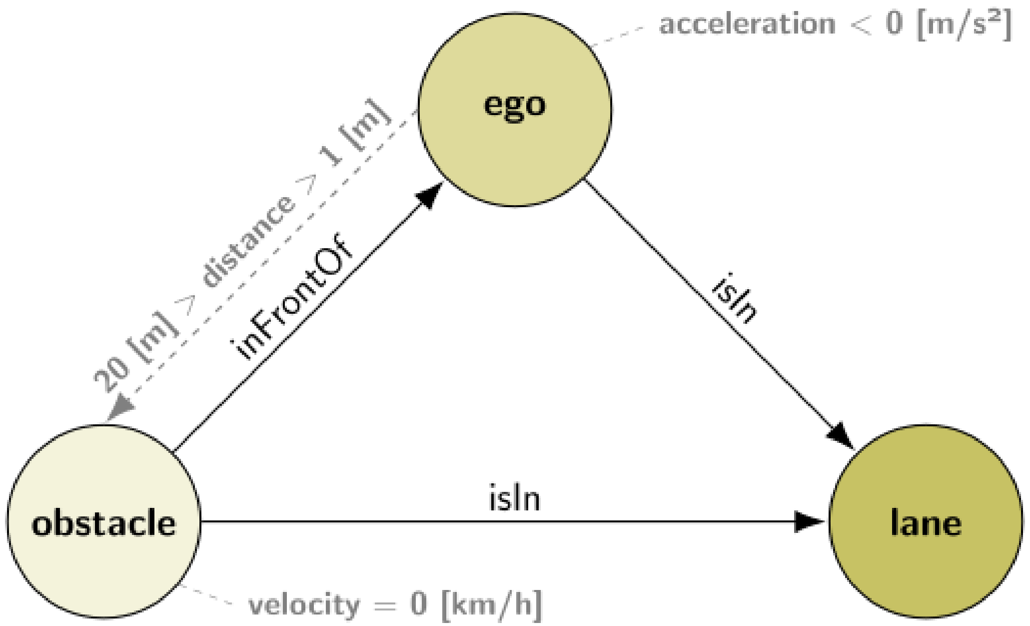
# Abstract Scene Graph (ASG) formalism

- ASGs build upon Scene Graphs
- Describe desired spatial relations between entities in a traffic scene
- Restrict the set of traffic scenes represented by a Scene Graph



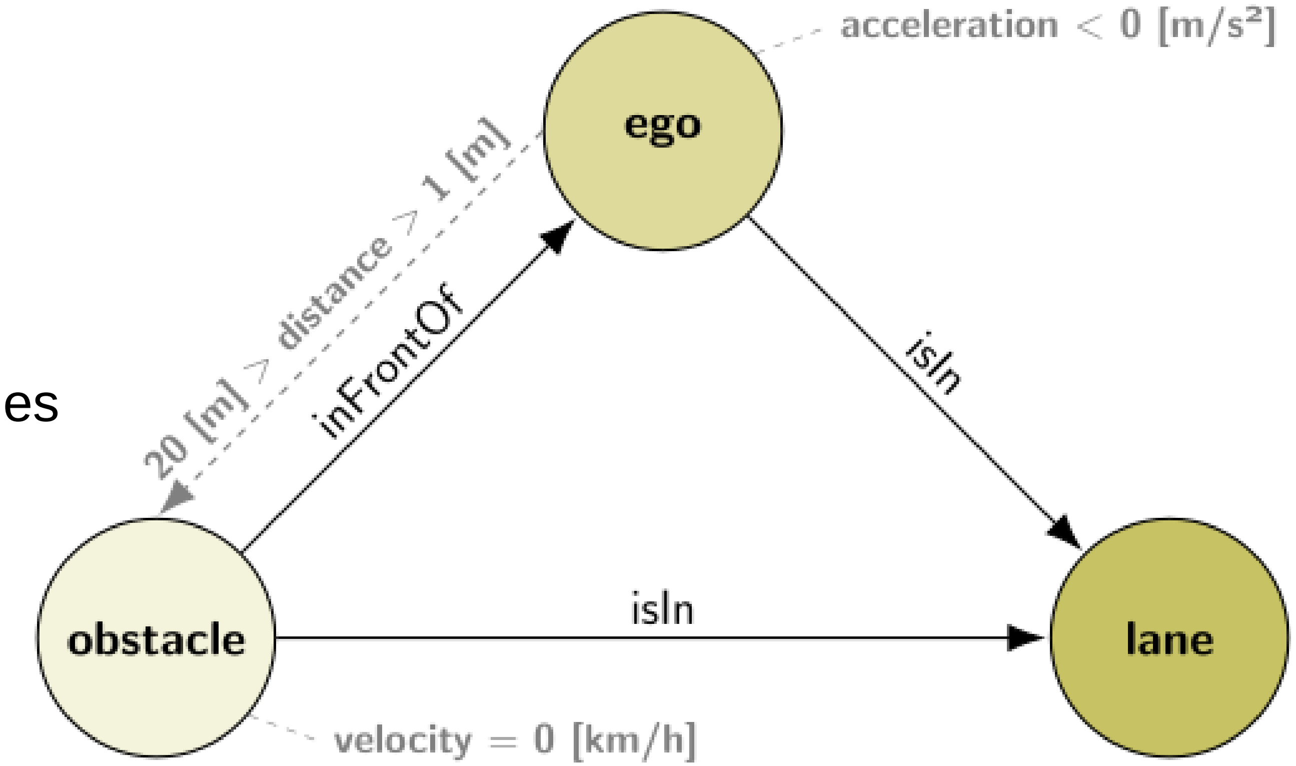
# Abstract Scene Graph (ASG)

- ASG is defined over an Object Model
  - Object Model consists of a finite set of
    - Basic types,
    - Object classes, and
    - Relation types
- ASG = $(G_A, D)$
  - $G_A$ is a directed heterogeneous graph
  - $D$ is a set of predicates



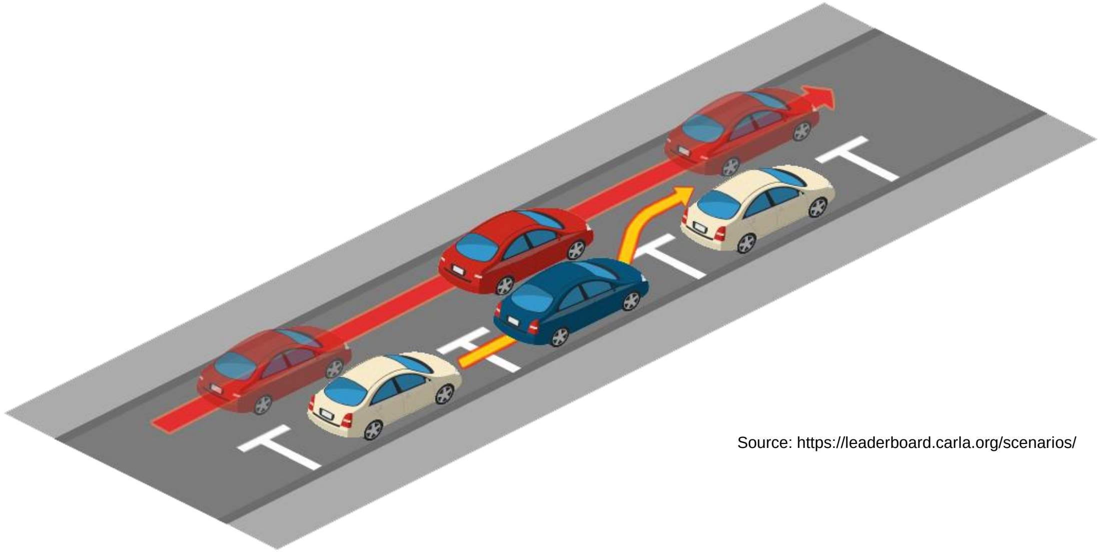
# Syntax of ASGs

- ASG = $(G_A, D)$
- $G_A = (V, E)$
  - $V$ : Finite set of typed nodes
    - Ego node must be present
  - $E$ : Finite set of typed directed edges
- $D$ : Finite set of predicates
  - Attribute and distance expressions



# Motivating Example Scenario: Parking Exit

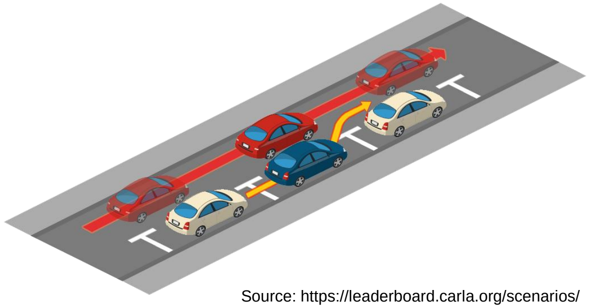
- *“The ego-vehicle must exit a parallel parking bay into a flow of traffic safely”*
  - Inspired from the NHTSA pre-crash scenario catalogue
  - Scenario = sequence of scenes



Source: https://leaderboard.carla.org/scenarios/

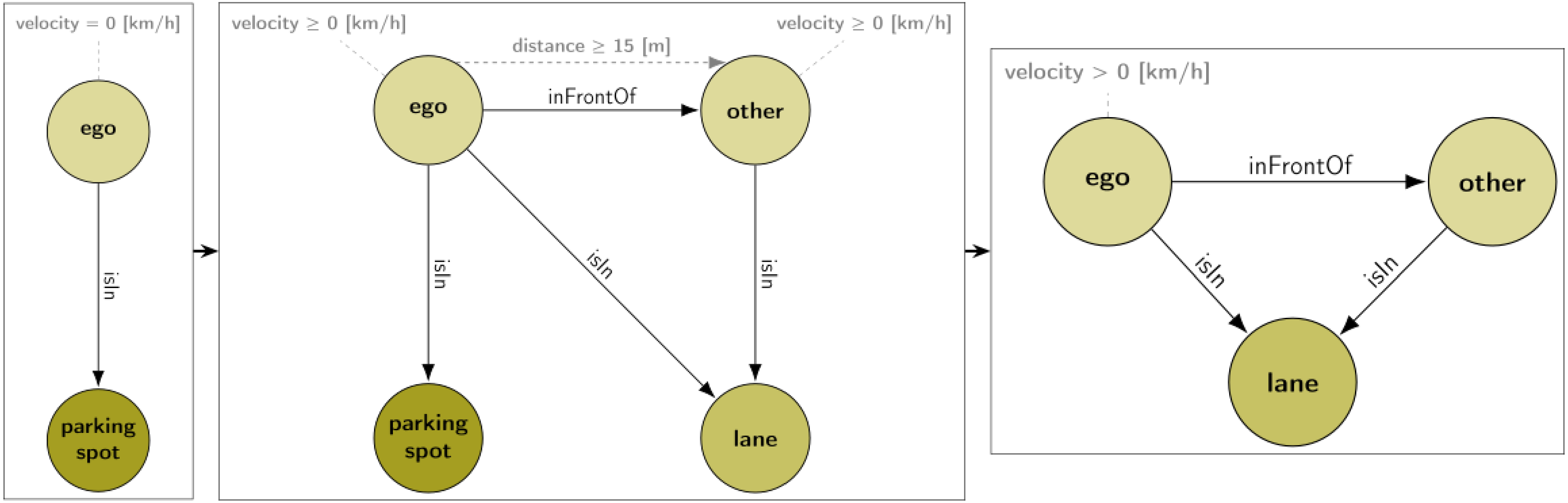
# Formalization of spatial properties using ASGs

- *“The ego-vehicle must exit a parallel parking bay into a flow of traffic”*
- To formalize this property, we need to identify
  - Entities: Vehicle, Lane, Parking Spot
  - Scenes present in the above scenario:
    - P1-1 ego is standing still in the parking spot
    - P1-2 ego is turning into the driving lane next to the parking spot while maintaining a safe distance (at least 15 m) to the next vehicle coming from behind
    - P1-3 ego has joined the lane completely and is driving ahead

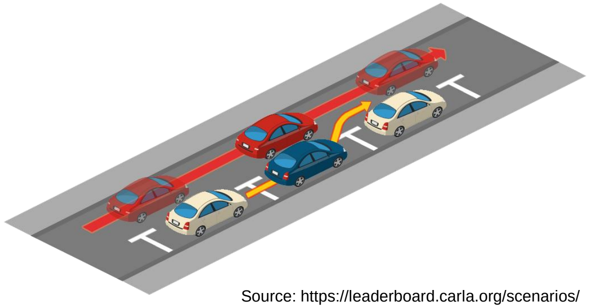


Source: https://leaderboard.carla.org/scenarios/

# Formalization of spatial properties using ASGs

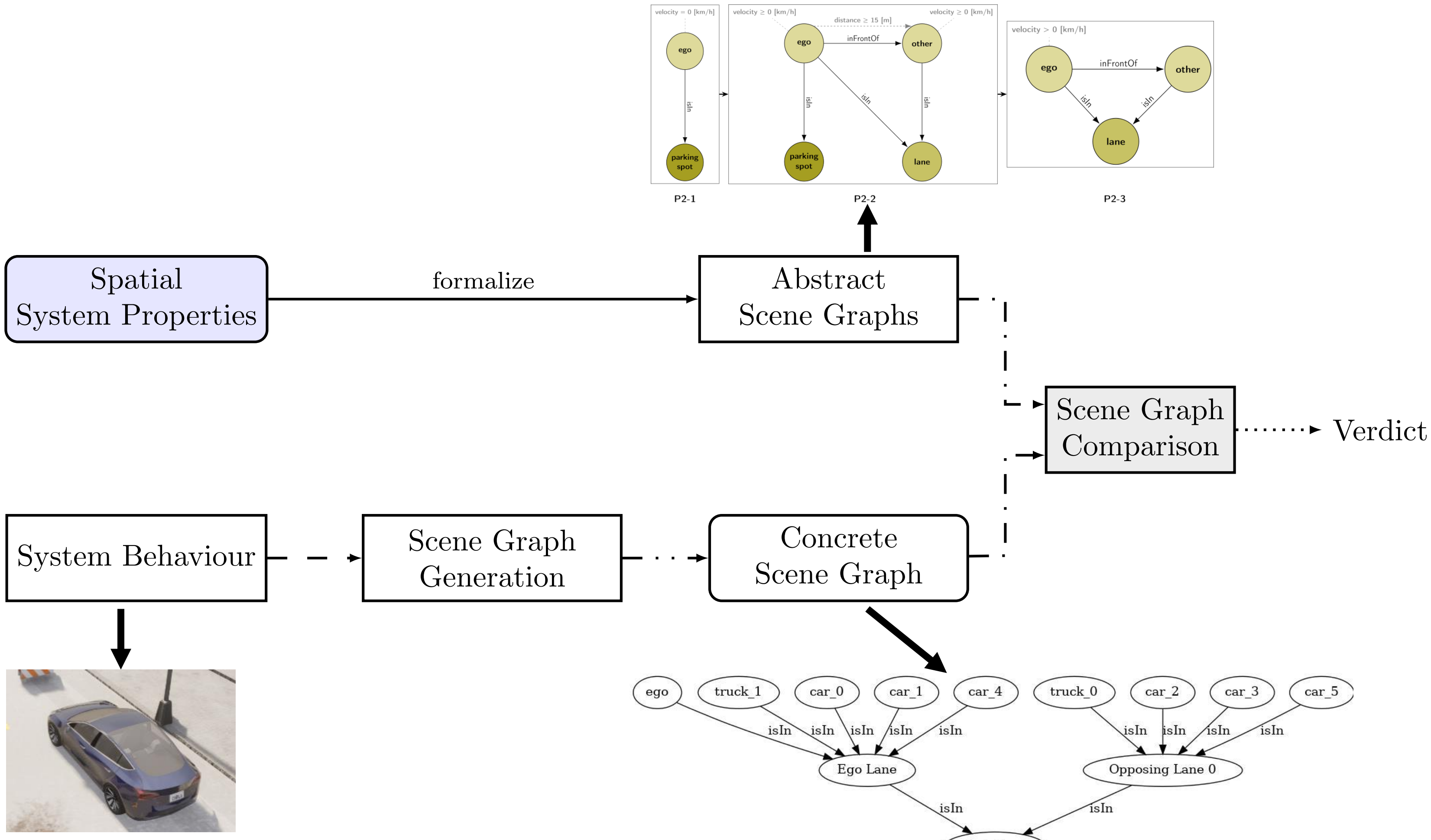



P2-1 P2-2 P2-3



Source: https://leaderboard.carla.org/scenarios/

# Runtime Monitoring using ASGs




Source: CARLA Simulator

Source: https://github.com/less-lab-uva/carla_scene_graphs

# Scene Graph Comparison : Current approach

- Let ASG = $(G_A, D)$ be an Abstract Scene Graph, CSG = $G_C$ be a Concrete Scene Graph and $G'_C$ be its subgraph
- We define CSG$\vDash$ ASG $\Leftrightarrow$ $(\exists G'_C\colon G'_C \cong G_A) \land (\rho_C \vDash D)$, where
  - $G'_C$ is isomorphic to $G_A$
  - $\rho_C$ contains the attribute values present in $G'_C$
- Use of subgraph isomorphism to extract suitable subgraph of $G_C$
  - Computationally expensive!
  - But, here could be fine since we know at least one (ego) node

# Scene Graph Comparison: Implemented algorithm

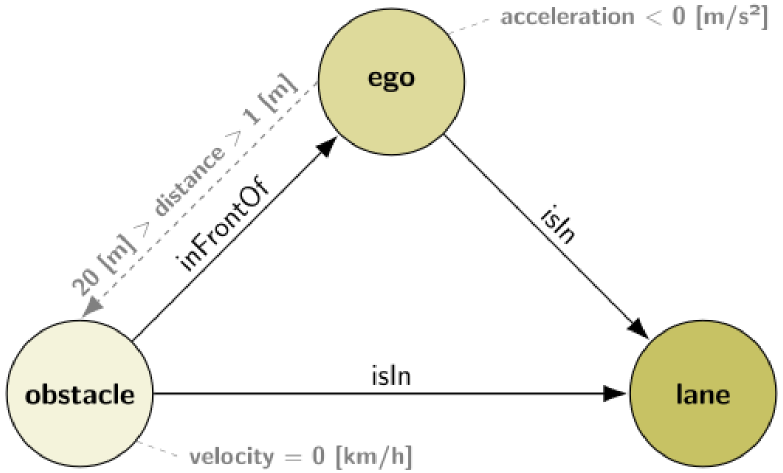
```
Input: ASG : (G_A, D) and CSG : G_C Output: Boolean verdict
procedure SG_COMPARISON(ASG, CSG)
    (exists, map) ← CHECK_ISOMORPHISM_VF2(G_C, G_A)
    if exists is True then
        attr ←GET_MAPPED_ATTRIBUTE_VALUES(map, ASG, G_C)
        for all predicate ∈ D do
            pred_verdict ←CHECK_IF_PREDICATE_SATISFIED(attr, predicate)
            if pred_verdict is False then
                verdict ← False ▷ If a predicate is not satisfied, CSG does not satisfy ASG
                end procedure
        verdict ← True            ▷ If all predicates are satisfied, CSG satisfies ASG
    else
        verdict ← False                              ▷ No isomorphic subgraph exists
    return verdict
```

# Future Work

- Extension of ASG syntax to formalize more complex spatial properties
- Mapping of used Object Model to a suitable ontology
- Alternative Scene Graph comparison approach
- Applying ASGs in other transportation domains such as maritime

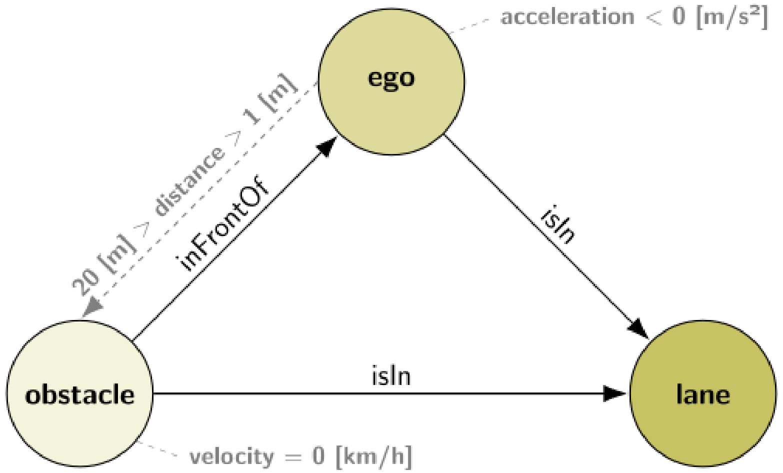
# Thanks for your attention!

# Questions?



# Takeaways

- Abstract Scene Graphs enable formalization of spatial properties of Automated Driving Functions in the automotive domain



# References

[1] Grundt, Dominik, et al. “Towards Runtime Monitoring of Complex System Requirements for Autonomous Driving Functions”

[2] Neurohr, Christian, et al. “Fundamental Considerations around Scenario-Based Testing for Automated Driving”

[3] Malawade, Arnav, et al. “roadscene2vec: A Tool for Extracting and Embedding Road Scene-Graphs”

[4] Neurohr, Christian, et al. “Criticality Analysis for the Verification and Validation of Automated Vehicles”

[5] Zipfl, Maximilian, et al. “Towards traffic scene description: The semantic scene graph.”

[6] Zipfl, Maximilian, et al. “Relation-based motion prediction using traffic scene graphs.”

[7] Zipfl, Maximilian, et al. “Traffic Scene Similarity: a Graph-based Contrastive Learning Approach.”

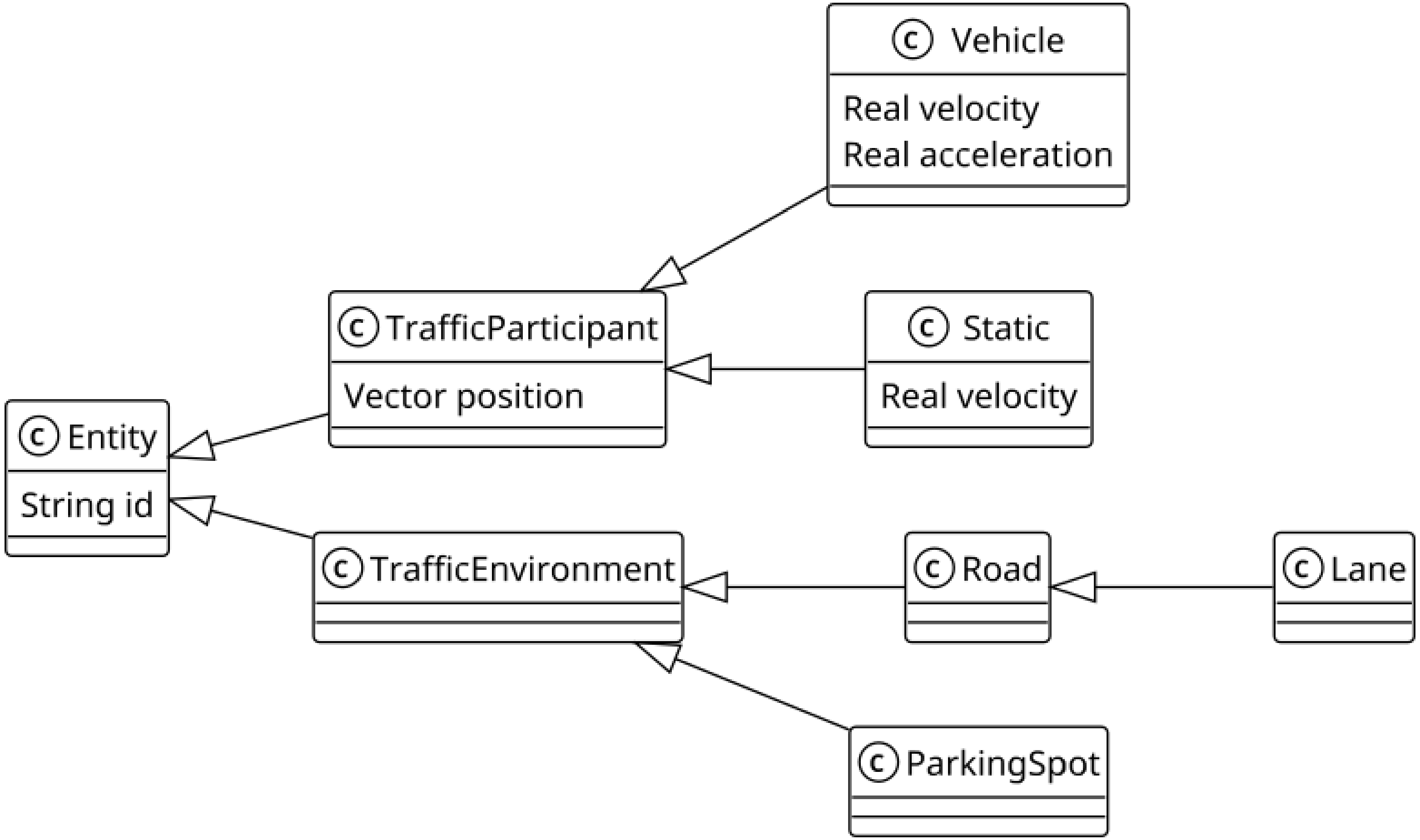
[8] Monninger, Thomas, et al. “Scene: Reasoning about traffic scenes using heterogeneous graph neural networks.”

[9] Yu, Shih-Yuan, et al. “Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions.”

[10] Malawade, Arnav Vaibhav, et al. “RS2G: Data-Driven Scene-Graph Extraction and Embedding for Robust Autonomous Perception and Scenario Understanding.”

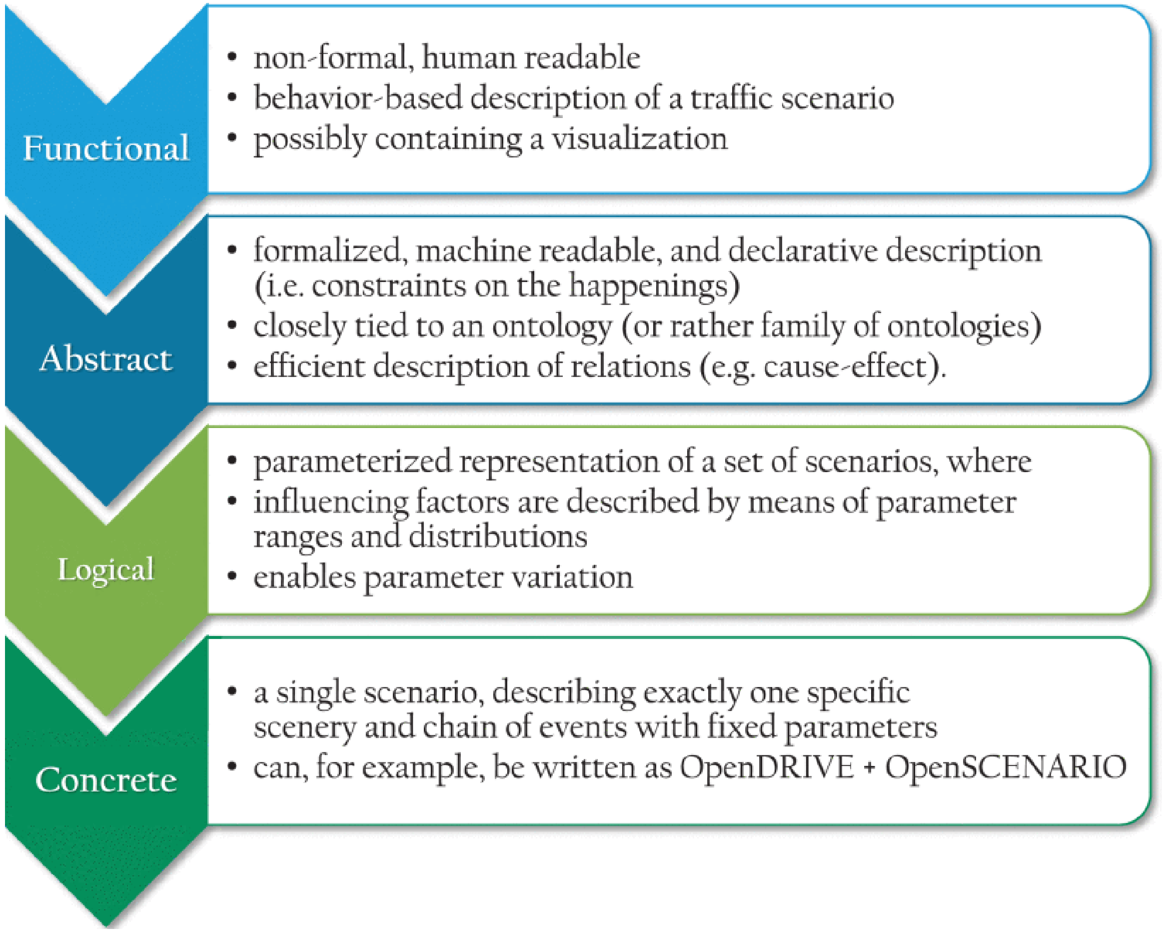
[11] Knyazev, Boris, et al. “Understanding attention and generalization in graph neural networks.”

# Object Model Class Diagram



# Scenario descriptions in Scenario-based Testing of Automated Vehicles

**Neurohr et al, Criticality Analysis for the Verification and Validation of Automated Vehicles, IEEE Access**



# Imprint

| | |
|---|---|
| Topic: | **Abstract Scene Graphs**: Specifying and Monitoring Spatial Properties of Automated Driving Functions |
| Date: | 2025-11-17 (YYYY-MM-DD) |
| Author: | Ishan Saxena |
| Institute: | Systems Engineering for Future Mobility (SE) |
| Image sources: | All images “DLR (CC BY-NC-ND 3.0)” unless otherwise stated |