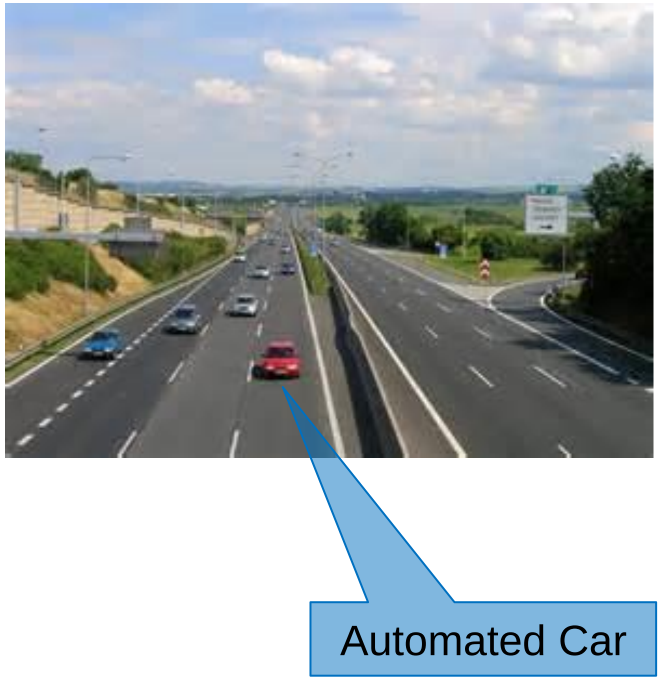
Formalizing scenarios for safety testing of automated driving functions

Hardi Hungar
Institute of Transportation Systems
German Aerospace Center (DLR)

This research was partially funded by the German Federal Ministry for Economic Affairs and Energy, Grant No. 19A15012F (PEGASUS), based on a decision by the Parliament of the Federal Republic of Germany. The responsibility for the content lies with the author.

Automated Driving System (ADS)
Example: Highway Pilot

- **Highly automated driving** on a highway under regular conditions
  - Passenger car
  - Highway or similarly equipped road
  - Speed limited to 130 km/h
  - Ordinary weather conditions



Included

- Stop & Go
- Changing lanes
- Overtaking
- Emergency manoeuvers
  - Braking
  - Evasive actions
- Fallback when reaching system boundaries:
  - Driver (with sufficient takeover time)
  - Risk minimizing maneuver (if driver does not respond)

Excluded

- Entering the highway
- Exiting the highway
- Bad weather
  - (very) Slippery surface
  - Heavy rain, snow, fog

SAE, J3016 2018-06

SAE: Levels of automation

SAE Society of Automotive Engineers

Driver responsibility

System responsibility

Highway Pilot

Highly automated driving

DDT dynamic driving task
OEDR object and event detection and response
ODD operational driving domain

Level	Name	Narrative definition	DDT		DDT fallback	ODD
			Sustained lateral and longitudinal vehicle motion control	OEDR		
Driver performs part or all of the *DDT*						
0	No Driving Automation	The performance by the *driver* of the entire *DDT*, even when enhanced by *active safety systems*.	*Driver*	*Driver*	*Driver*	n/a
1	Driver Assistance	The *sustained* and *ODD*-specific execution by a *driving automation system* of either the *lateral* or the *longitudinal vehicle motion control* subtask of the DDT (but not both simultaneously) with the expectation that the *driver* performs the remainder of the *DDT*.	*Driver* and *System*	*Driver*	*Driver*	Limited
2	Partial Driving Automation	The *sustained* and *ODD*-specific execution by a *driving automation system* of both the *lateral* and *longitudinal vehicle motion control* subtasks of the *DDT* with the expectation that the *driver* completes the *OEDR* subtask and *supervises* the *driving automation system*.	***System***	*Driver*	*Driver*	Limited
ADS ("*System*") performs the entire *DDT* (while engaged)						
3	Conditional Driving Automation	The *sustained* and *ODD*-specific performance by an *ADS* of the entire DDT with the expectation that the *DDT fallback-ready user* is *receptive* to *ADS*-issued *requests to intervene*, as well as to *DDT performance-relevant system failures* in other *vehicle* systems, and will respond appropriately.	*System*	***System***	*Fallback-ready user (becomes the driver during fallback)*	Limited
4	High Driving Automation	The *sustained* and *ODD*-specific performance by an *ADS* of the entire *DDT* and *DDT fallback* without any expectation that a *user* will respond to a *request to intervene*.	*System*	*System*	***System***	Limited
5	Full Driving Automation	The *sustained* and unconditional (i.e., not *ODD*-specific) performance by an *ADS* of the entire *DDT* and *DDT fallback* without any expectation that a *user* will respond to a *request to intervene*.	*System*	*System*	*System*	**Unlimited**

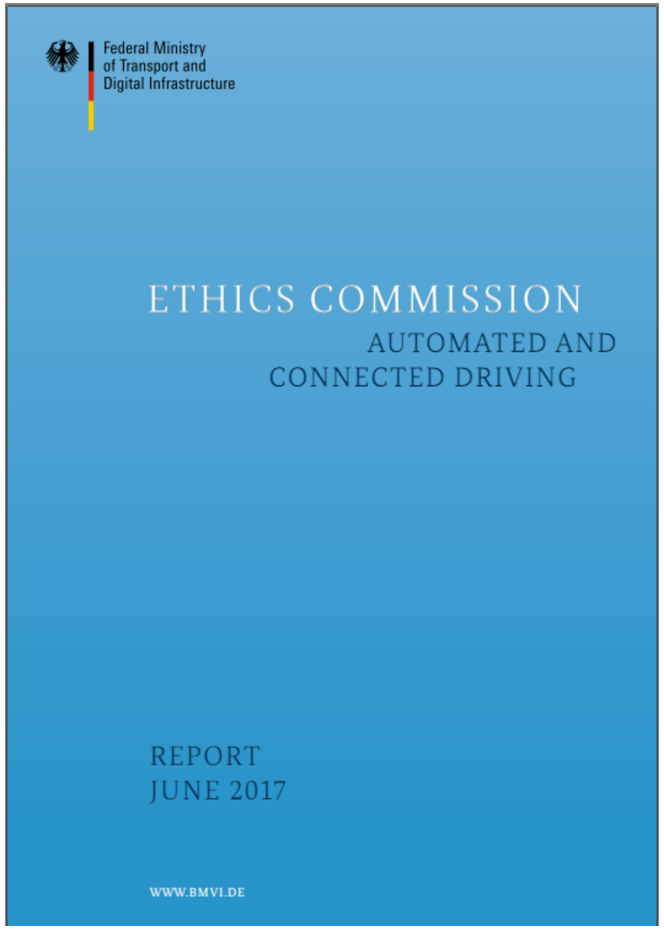
Safety target for automated driving

Ethics Commission on Automated Driving set up by the German Federal Ministry of Transport and Digital Infrastructure (BMVI)

Fully automated driving systems:

1. […] [Their] primary purpose […] is to **improve safety** for all road users.

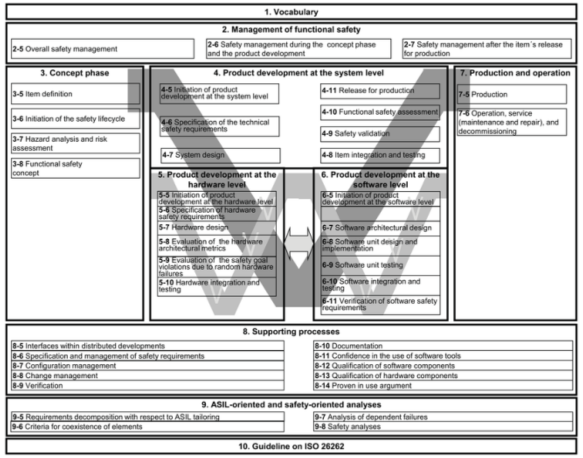
2. […] produce at least a diminution in harm compared with human driving, in other words a **positive balance of risks**.



The “standard” approach – ISO 26262

- **ISO 26262**: Standard „Road Vehicles – Functional Safety“ for developing systems with electronic elements (additional considerations: SOTIF ISO/WD PAS 21448)
  - Risk-based approach to safety
    - Risk $\approx \sum_{h \in H} E_h * C_h * S_h$
      - H: Set of harmful events h
      - E: probability of occurrence (precisely: expected number per time unit)
      - C: controllability (here: probability of *not* avoiding an accident)
      - S: severity of event (injuries, fatalities)

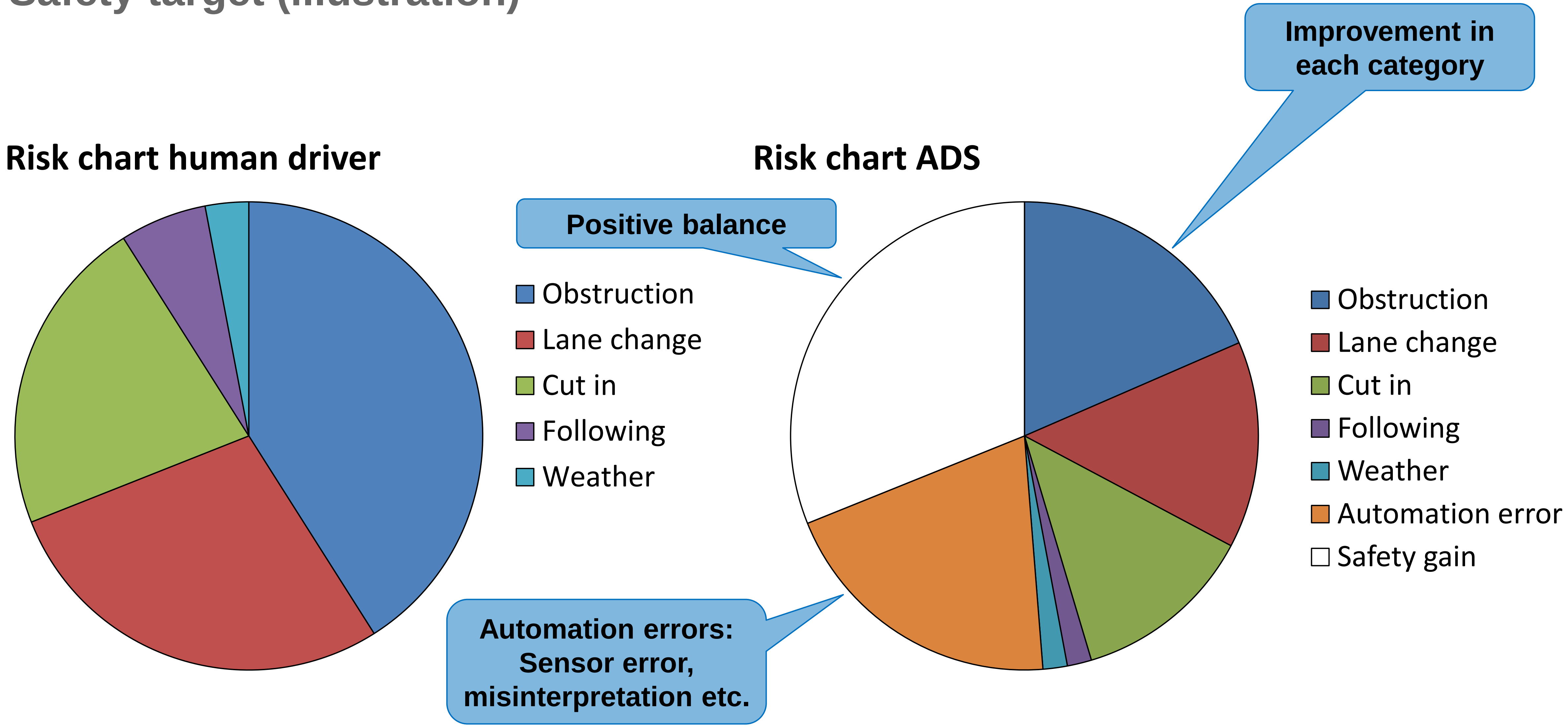
Similar to insurance risk calculation



ISO 26262, Overview figure

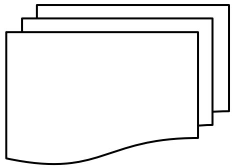
SOTIF: Road vehicles – Safety of the intended functionality

Safety target (illustration)



Risk assessment (commonly applied procedure)

- List all hazards
- Determine
  - Exposure
  - Criticality
  - Severity
- Sum up for overall risk



Hazard	E	C	S	Risk
Obstruction				
Lane change				
Cut in				
Cut through				
Overtaking				
Lane violation				
…				
…				
Sum				

Systematic computation of risk chart

1. Derive all potentially critical evolutions

2. Formalize the evolutions in precise descriptions of classes of evolutions

3. Exhaustive testing of evolution classes
   1. Derive concrete instantiations of a class
   2. Test concrete instances
   3. Identify critical instances

4. Analyze the critical instances
   1. Detailed evaluation
   2. Aggregate in risk chart

Hazard	E	C	S	Risk
Obstruction				
Lane change				
Cut in				
Cut through				
Overtaking				
Lane violation				
...				
...				
Sum				

Functional scenarios

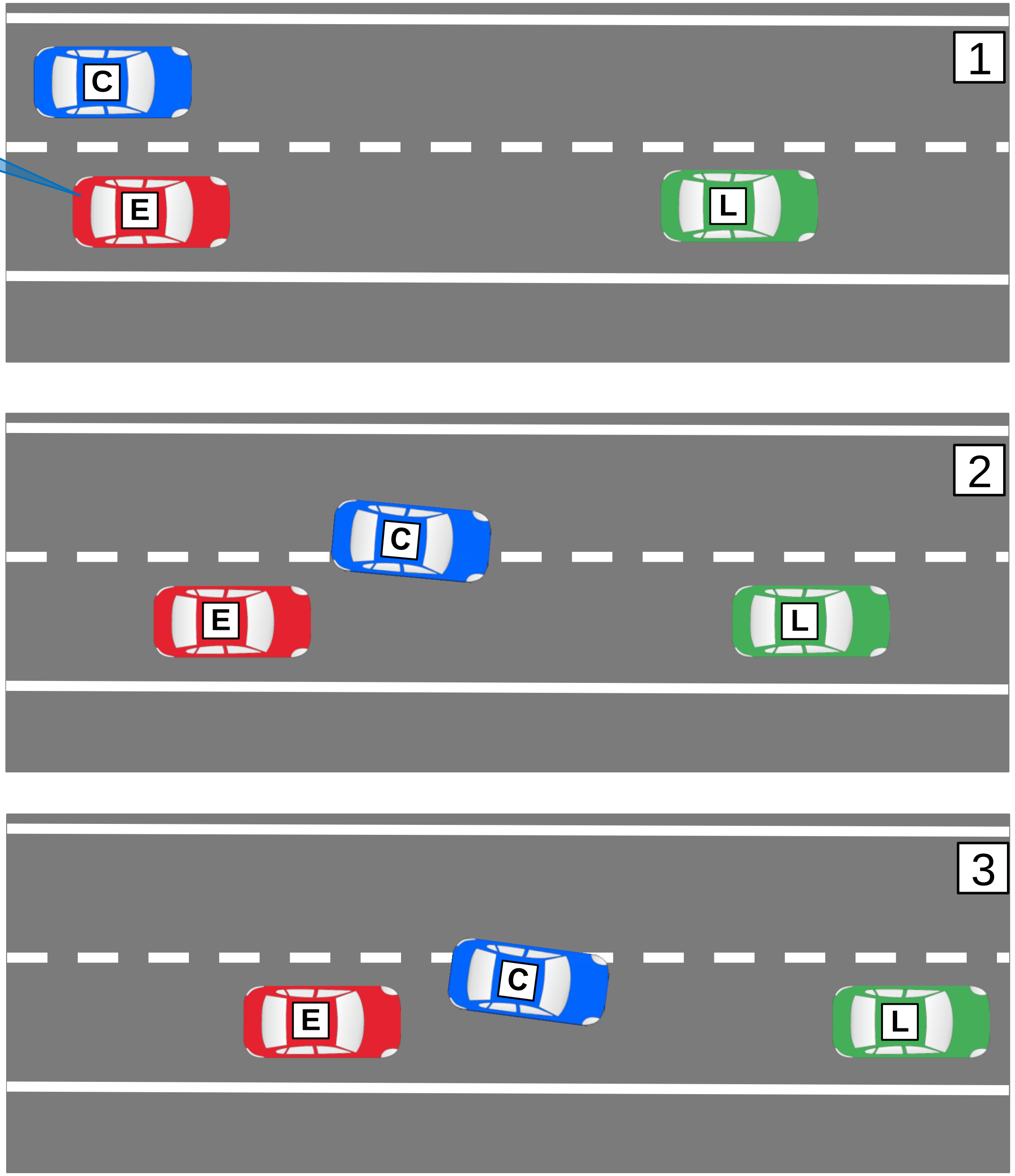
Logical scenarios

Concrete critical scenarios

Risk chart

Functional scenario “cut in“

Ego vehicle



- Rough storyboard of a cut-in evolution

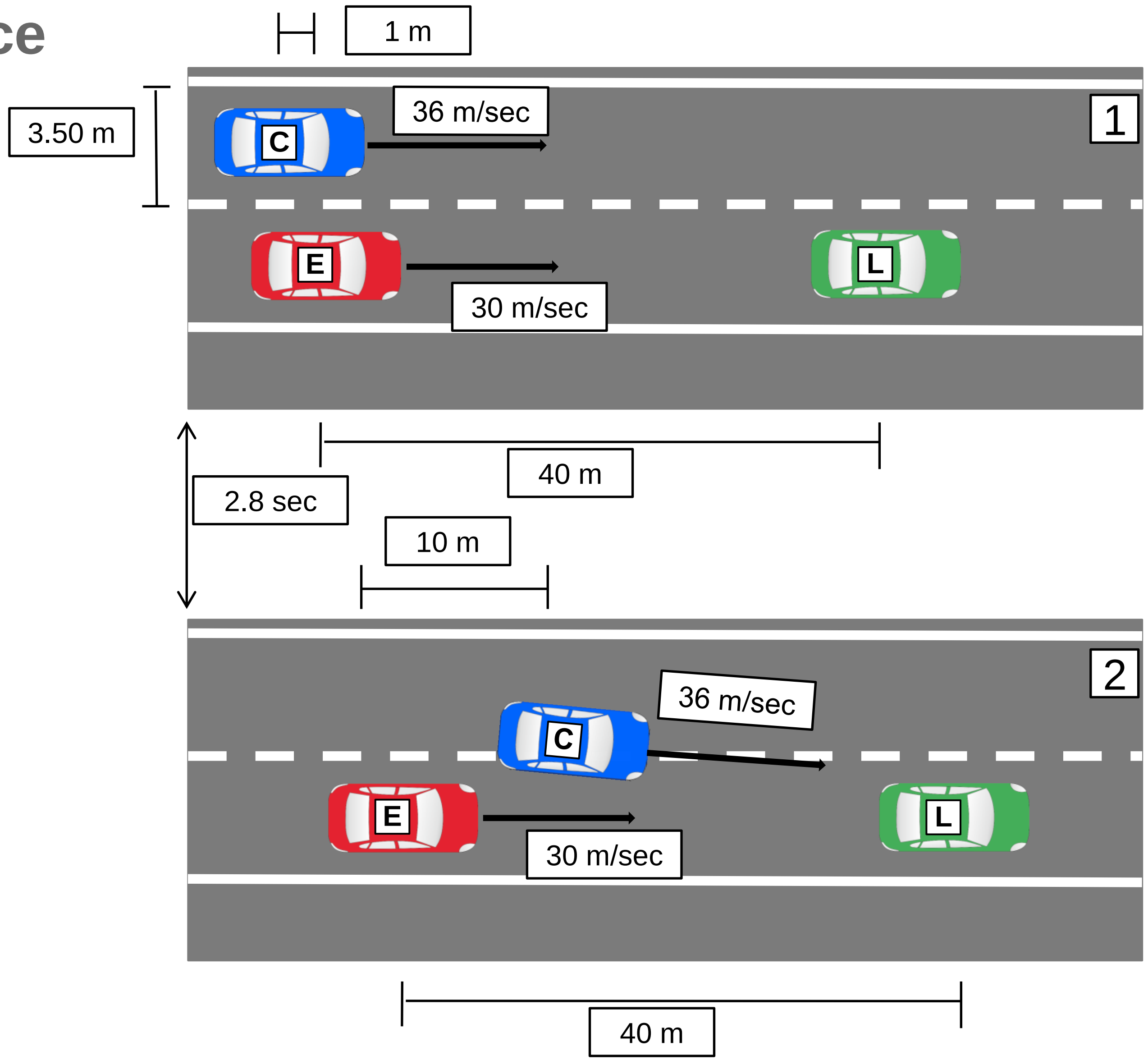
- Sequence of events
  - C is approaching on left lane
  - C overtakes E
  - C changes to right lane in front of E

- Parametrizing and varying over discrete variants yields the concrete instantiations of a “cut-in“

E Ego vehicle
C Cut-in vehicle
L Leading vehicle

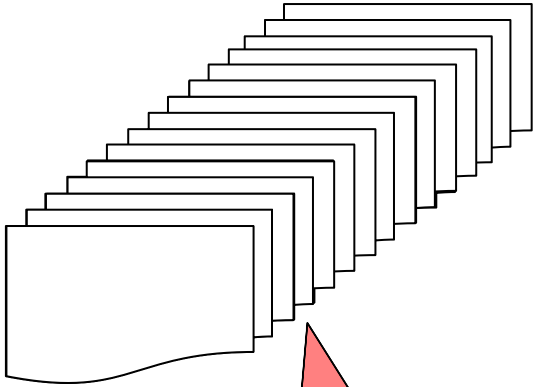
Cut in: Example of a concrete instance

- Deriving a concrete test scenario
  - Street dimensions
  - Relative positions of vehicles (road and other vehicles)
  - Velocities of vehicles
  - Changes of the dynamic parameters over time
- The derivation process should be systematic
  - This necessitates a formal description of scenarios



Standard risk computation

- List all hazards
- Derive all concrete instances
- Determine
  - Exposure
  - Criticality
  - Severity



A very long list!

- Sum up for overall risk

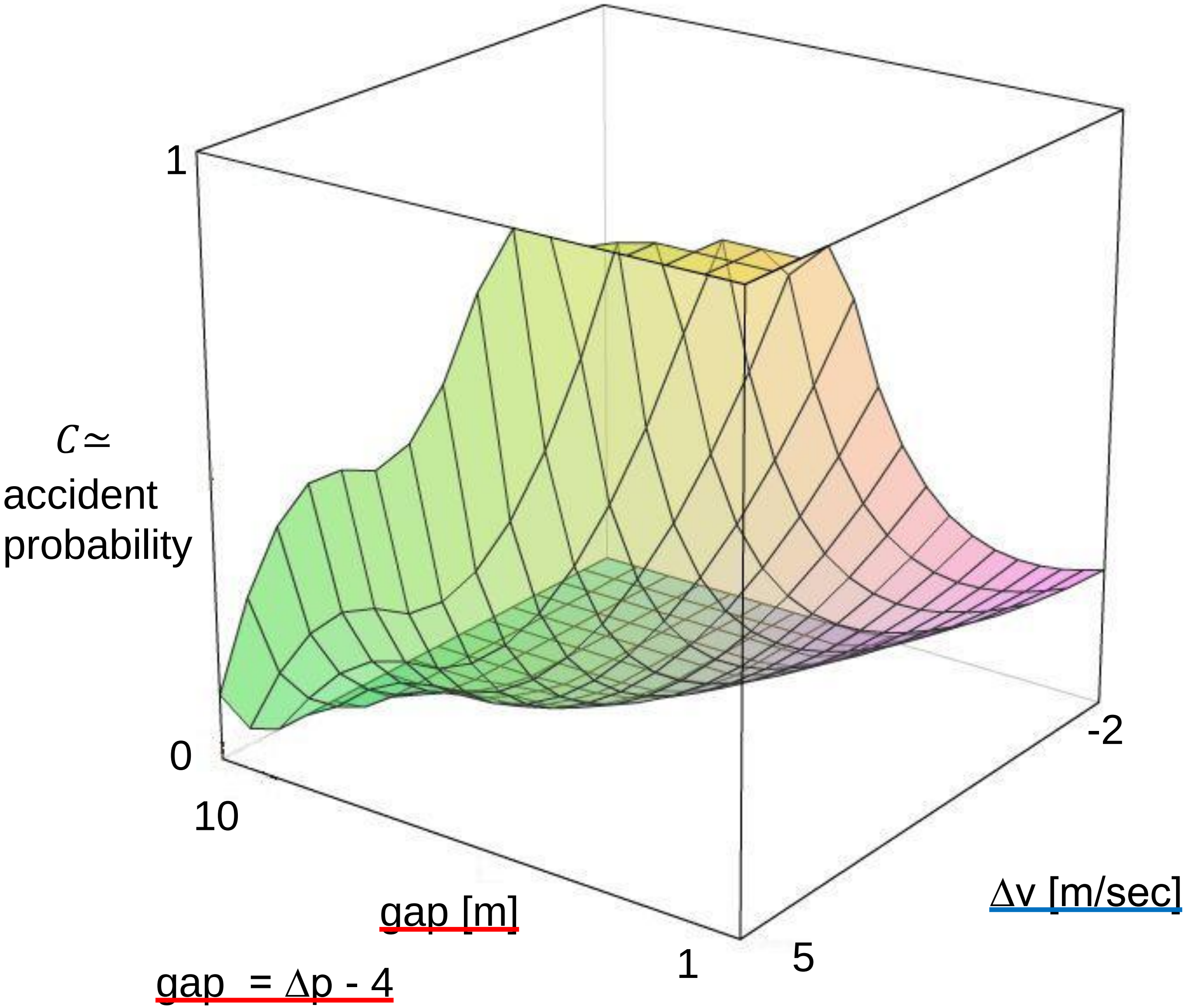
Automation needed

Hazard	E	C	S	Risk
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 100 km/h				
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h				
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 130 km/h				
…				
…				
…				
Sum				

Risk computation illustration Scenario “Cut-in“: Accident probability (“C“)

Cut-in (left, from behind)

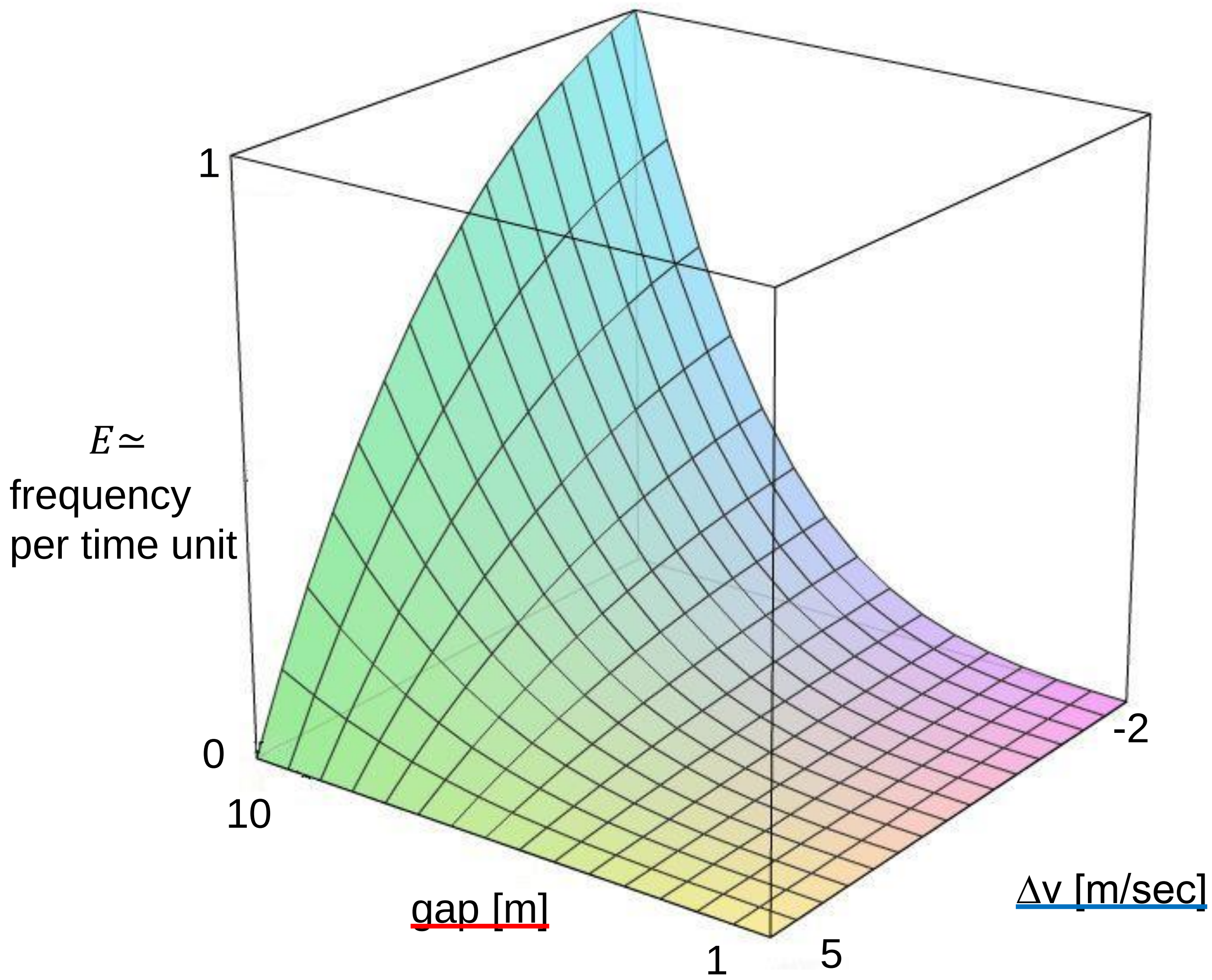
- Step 1:
  - Velocity [m/sec]: E , L: [22]; C-E: [1,45];
  - Position [m]: L-E: [33,100]; E-C: [0,30];
  - …
- Step 2: Cut-in starts (C crosses lane marking) Δt: [2,20]
  - Velocity [m/sec]: Δ L: [-7,+7]; Δ C: [-40,+4]; C-E: [-5,2]; C-L:[-9,12]
  - Position [m]: L-E: [25,110]; C-E: [3,12]; L-E: [15,100]
  - …
- Step 3: Cut-in completed (C has crossed lane marking halfway) Δt: [0.5,4]
  - Velocity [Δ m/sec]: …
  - …



Risk computation illustration Scenario “Cut-in“: Exposure (“E”)

Visualization of frequency of cut-in depending on

- Δv [m/sec]: velocity difference between Ego vehicle and Cut-in vehicle
  - The frequency *decreases* for relatively slower Cut-in vehicle
  - Usually, the Cut-in vehicle is faster than the Ego vehicle (negative values of Δv)
- gap [m]: gap between Cut-in and Ego vehicle:
  - The frequency *increases* with gap size
  - Usually, the gap is reasonably large

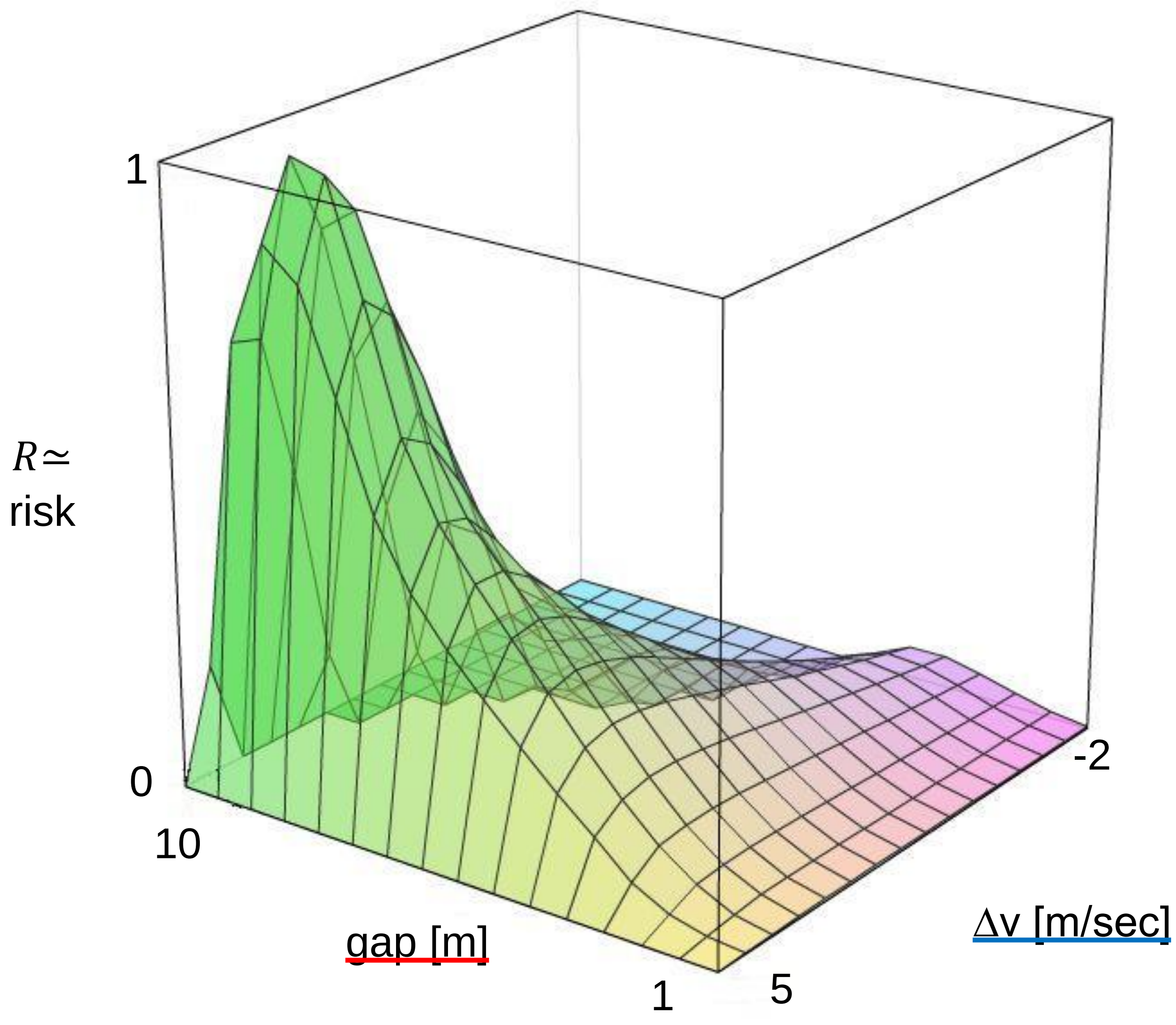


Risk computation illustration Scenario “Cut-in“: Risk

Visualization of risk* of cut-in

- Risk is highest for
  - a rather high velocity difference
    $\Delta v \approx 4$ [m/sec]
  - A narrow (but not minimal) gap
    gap ≈ 9 [m]
  - The highly dangerous situations occur less often
- The numeric risk is to be computed as the integral of the risk function

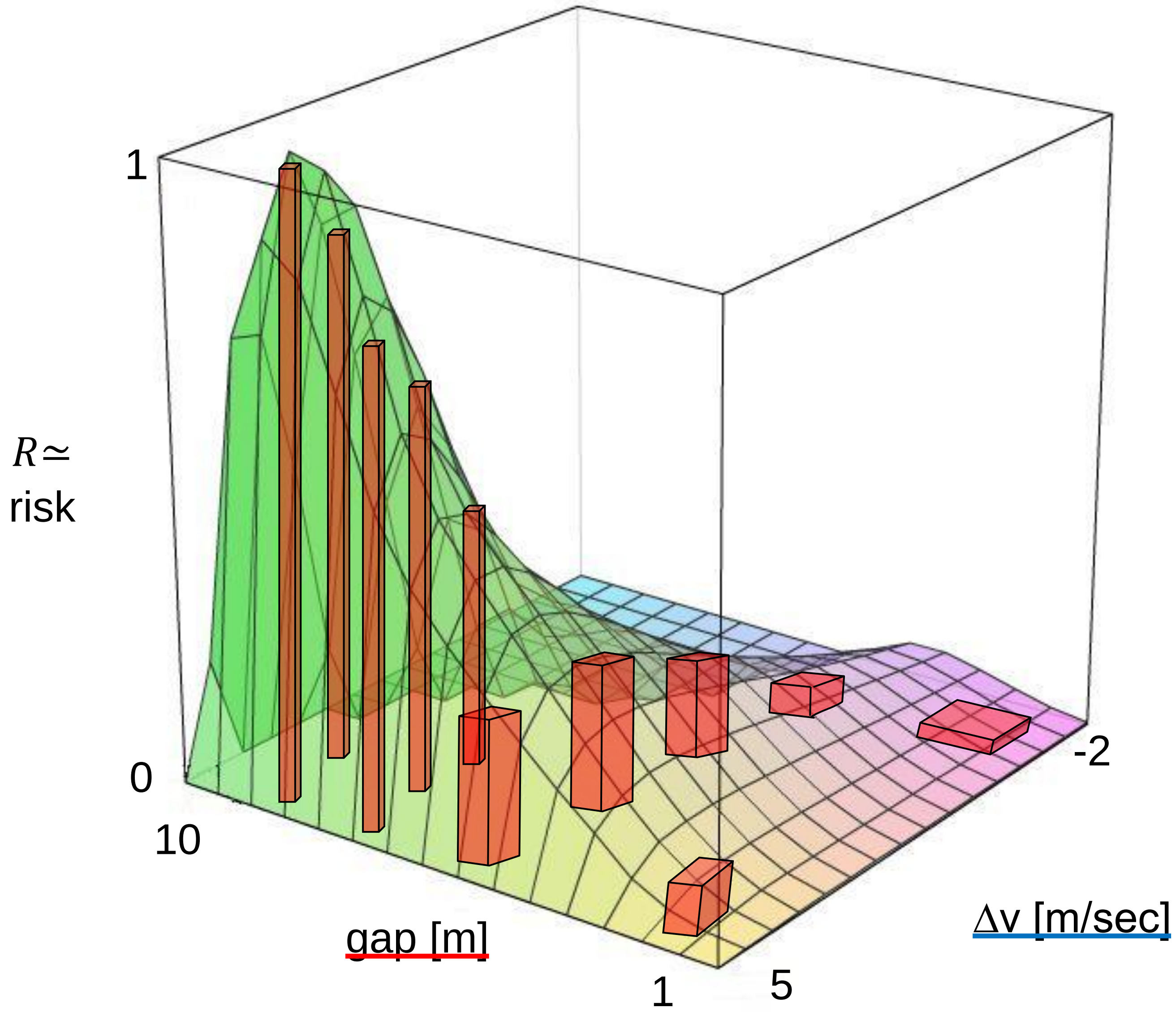
* The severity is assumed to be constant, here



Risk computation illustration Scenario „Cut-in“: Risk integration by simulation

Computation by approximate discrete summation

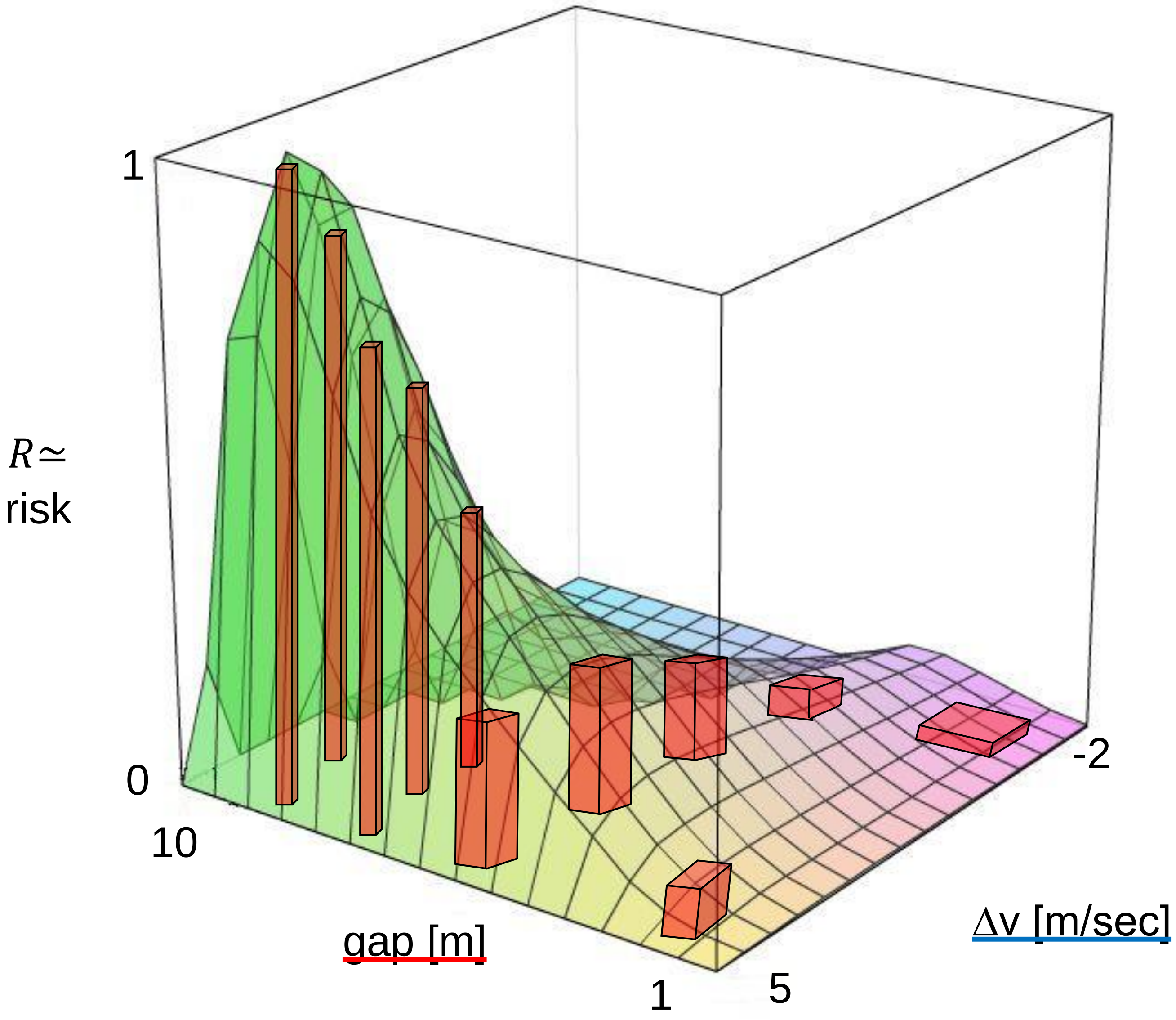
- Like Riemann integral approximation
- Each column represents the result of a test run (simulation / proving ground / field)
- Lower test density in regions with low accident probability



Risk computation illustration Scenario „Cut-in“: Risk integration by simulation

This would work, if

- we had a reliable simulation tool
- we had a complete test specification
- we could estimate the accident probability (“**C**”) of each simulated scenario
- we knew the frequency of each scenario (“**E**”)
- we could judge the accident severity (“**S**”)



Risk computation illustration Scenario „Cut-in“: Risk integration by simulation

This would work, if

- we had a reliable **simulation tool**
- we had a **complete test specification**
- we could estimate the **accident probability** (“**C**”) of each simulated scenario
- we knew the **frequency** of each scenario (“**E**”)
- we could judge the accident **severity** (“**S**”)

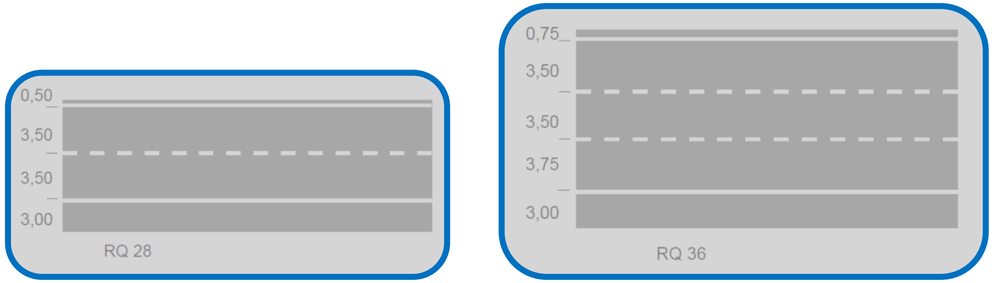
To be constructed

Can be measured by testing

Few valid data available

Only rough models available

Formalization of scenarios: Description layers



- L1: Street layer:
  - Geometry, topology, material
- L2: infrastructure :
  - Boundaries, traffic signs, markings

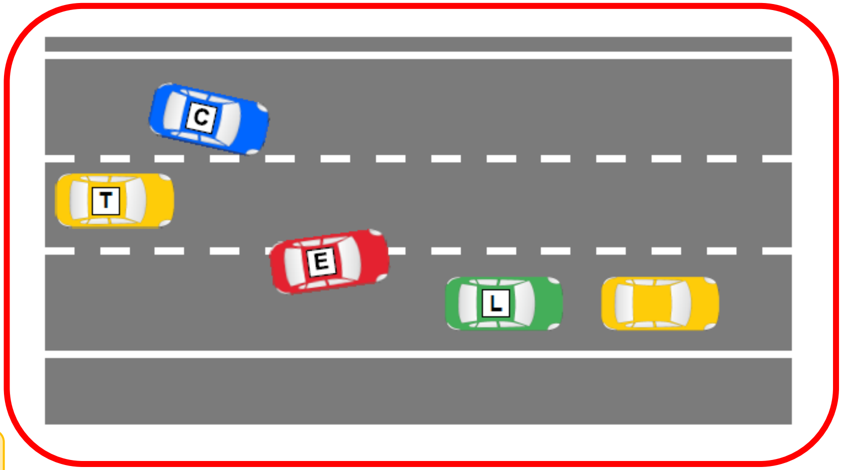
manageable

Irregular variations

- L3: Temporary modification of elements of L1 and L2 (example: installations of construction sites)

Focus

- L4: Moving objects:
  - Types and specificies, dynamics



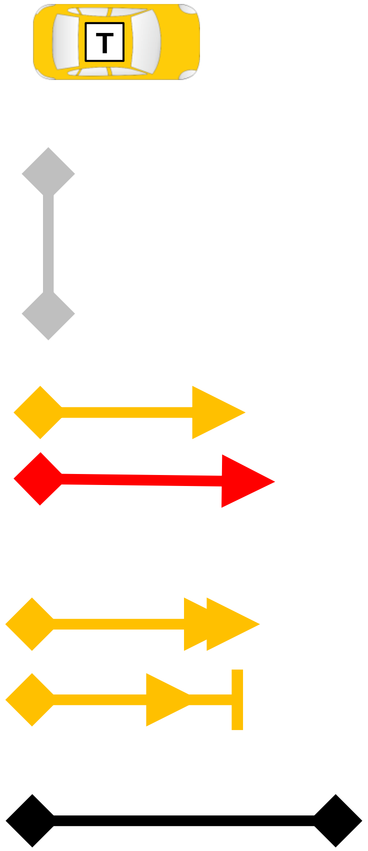
- L5: Environment conditions:
  - Weather, light

Very diverse

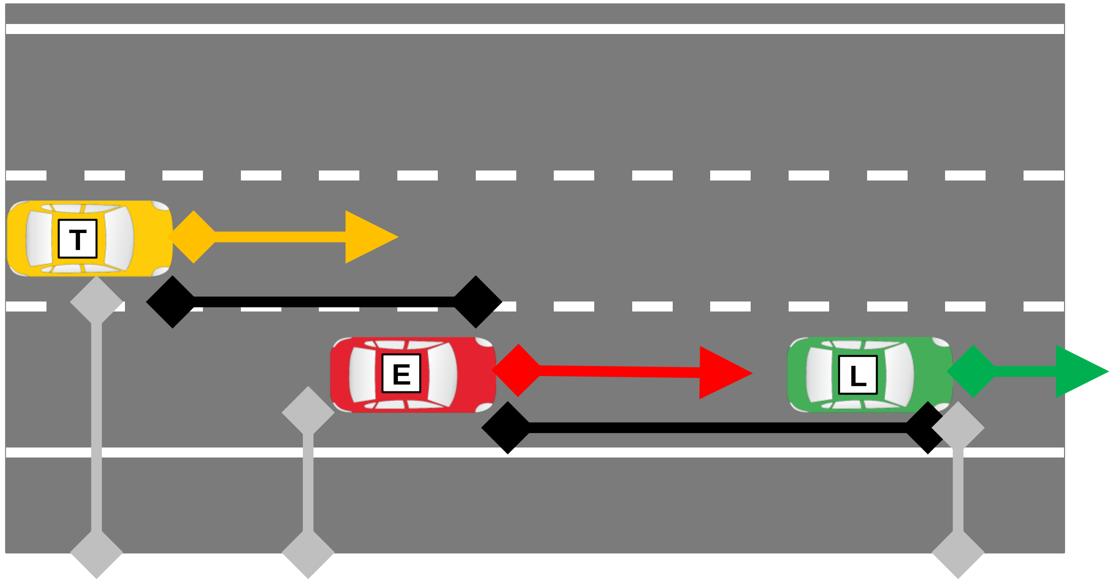
Layer definition after: Schuldt et al.
Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen, *AAET* 2013. (further developed in PEGASUS)

Scene: snapshot of evolution

- **Traffic participants**
  - **T, E, L**
- **Positions on the street**
  - Distance from road edge
- **Velocities**
  - Acceleration
  - Deceleration
- **Positions**
  - (here: relative to **E**)

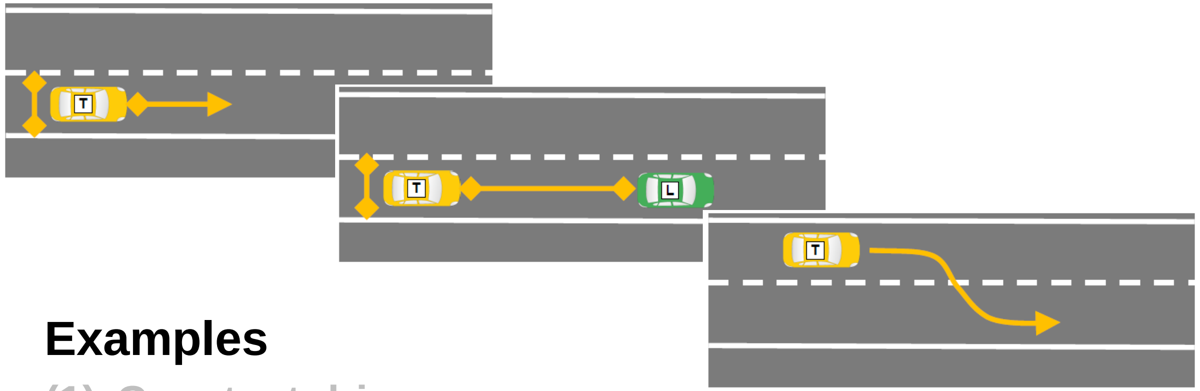


Graphical representation



More complex: links between scenes

Maneuver macros: Linking scenes to evolutions



Program-like descriptions of vehicle behavior

a. **Geometry**:
   - Lateral position
   - Discrete shape type: straight, sinusoidal, etc.
   - Modifiers: distortions, deviations

b. **Execution**:
   - time profile
   - Completion condition (e.g.: time slot, space limitations)
   - Absolute or relative to other traffic particpants

c. **End and exit conditions**

Examples

(1) Constant drive

a. Lane 1, straight, low lateral deviations
b. constant velocity, low deviation
c. --

(2) Following

a. Lane 1, straight, low lateral deviations
b. Velocity adjusted on distance to lead vehicle
c. Lane change of lead vehicle

(3) Lane change

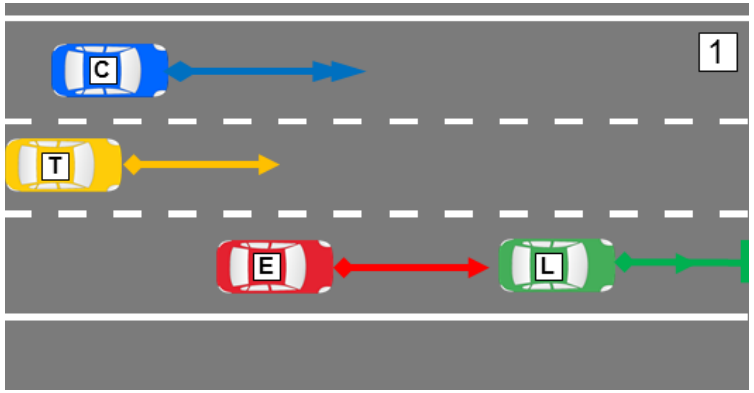
a. Lane 2, sinusoidal negative, low lateral deviations
b. constant velocity, low deviation
c. Completion of trajectory

discrete parameter

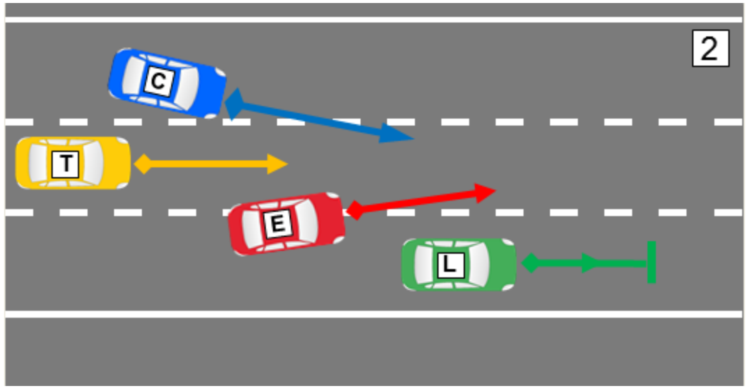
numerical parameter

Example scenario: conficting lane changes

0. The ego vehicle E follows L on the right lane
   T is driving on the middle lane with the same velocity



1. C overtakes T,
   L decelerates, which might provoke E to change lanes



2. C and E both move towards the middle lane

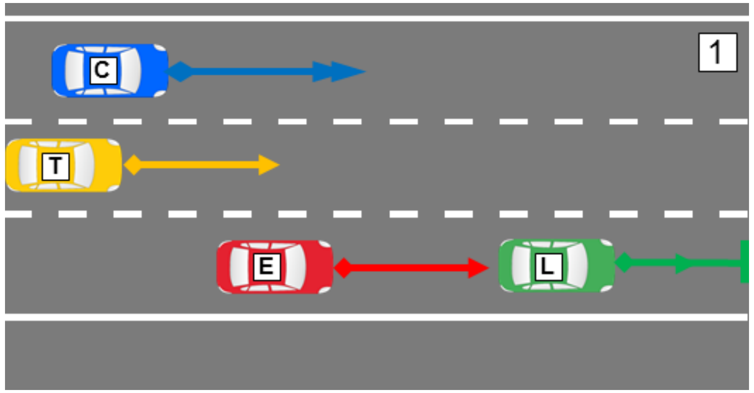
Example scenario: conficting lane changes
Programming the scenario with maneuver macros

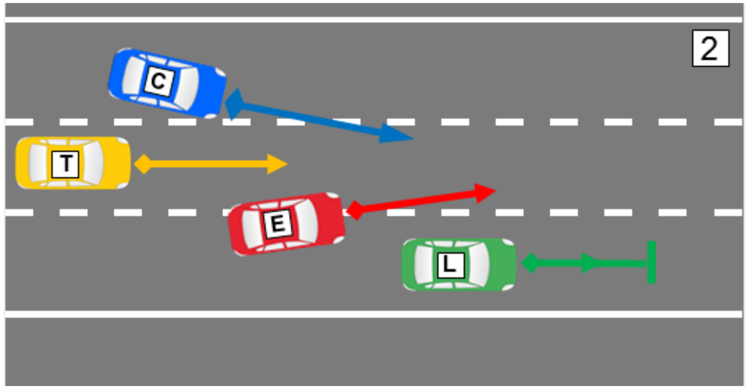
0. L: constant drive
   T: constant drive
   C: lane following with goal constellation
   depending on (C, T, E)

1. L: lane following, decelerating
   T: constant drive
   C: lane follwing with goal constellation
   depending on (C, T, E)

C reaches goal constellation / E veers out

1. L: lane following, decelerating
   T: constant drive
   C: lane change





Precisely specifying the test space with logical scenarios

Shown

- **Building blocks of logical scenarios**
  - Maneuver macros as elementary constituents
  - Scenario definition by composing maneuver macros
- **Logical scenarios** are similar to programs
  - Defining logical scenarios needs testing them (no reasonably complex program will be correct on first writing)

Comments

- The formalization may be seen as a domain-specific language
- The use of macros results in comprehensible definitions
- That maneuver macros capture real behaviors realistically can be validated on a reasonably small set of observation data.

Precisely specifying the test space with logical scenarios

Shown

- **Building blocks of logical scenarios**
  - Maneuver macros as elementary constituents
  - Scenario definition by composing maneuver macros
- **Logical scenarios** are similar to programs
  - Defining logical scenarios needs testing them (no reasonably complex program will be correct on first writing)
- **Coverage of the test space** by complementary scenario spaces
  - Manually manageable set of logical scenarios (though certainly large)

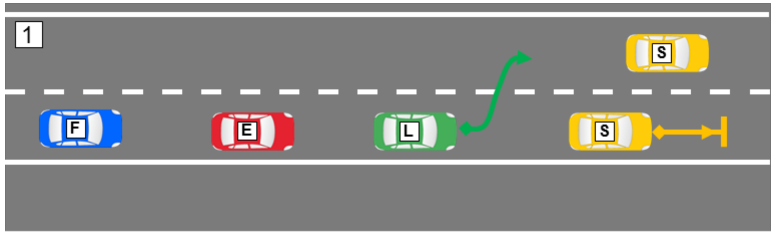
Next

Comments

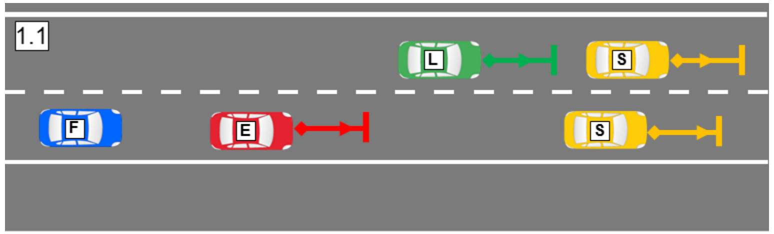
- The formalization may be seen as a domain-specific language
- The use of macros results in comprehensible definitions
- That maneuver macros capture real behaviors realistically can be validated on a reasonably small set of observation data.

Scenario branching: Example

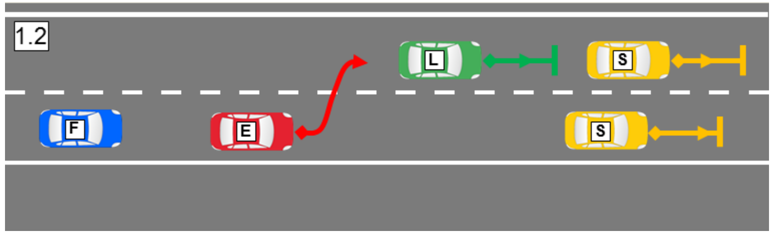
1. E follows L on the right lane
   S decelerates
   L changes lanes



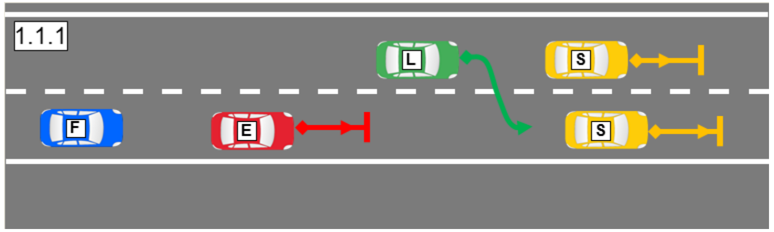
1.1 E decelarates
L decelerates



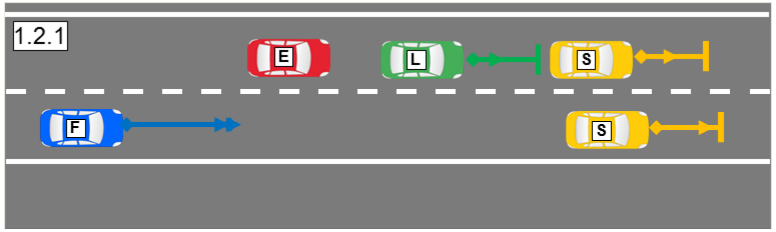
1.2 E changes lanes
L decelerates



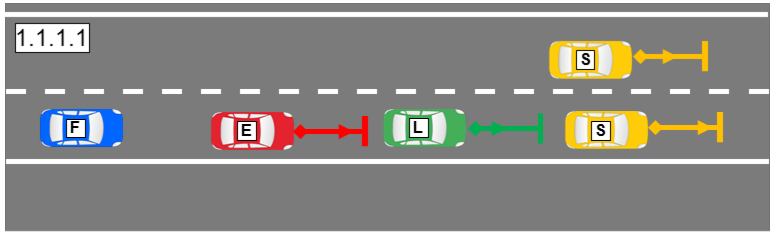
1.1.1 L changes back



1.2.1 L decelerates hard
F accelerates on right lane (closing gap)

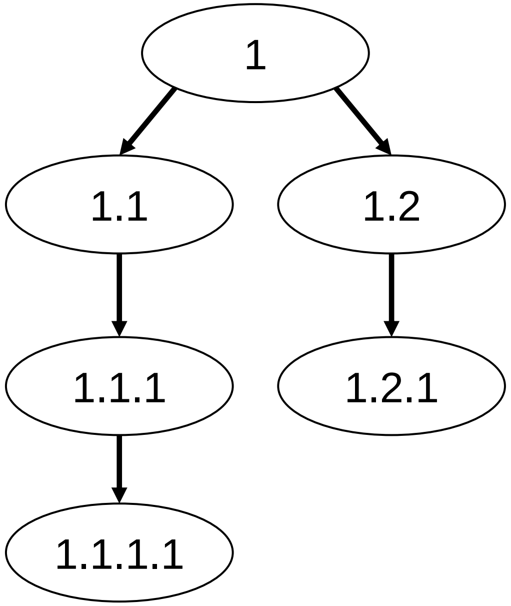


1.1.1.1 L decelerates



Scenario branching: Tree structure

1. E follows L on the right lane
   S decelerates
   L changes lanes



1.1 E decelerates
L decelerates

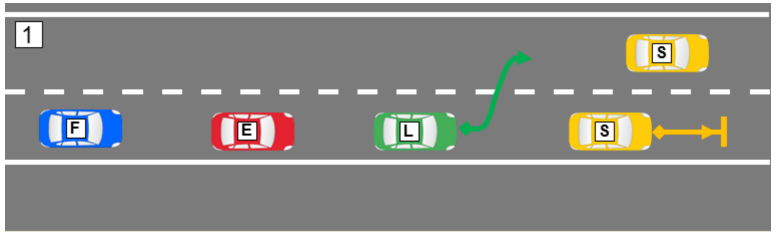
1.2 E changes lanes
L decelerates

1.1.1 L changes back

1.2.1 L decelerates hard
F accelerates on right lane (closing gap)

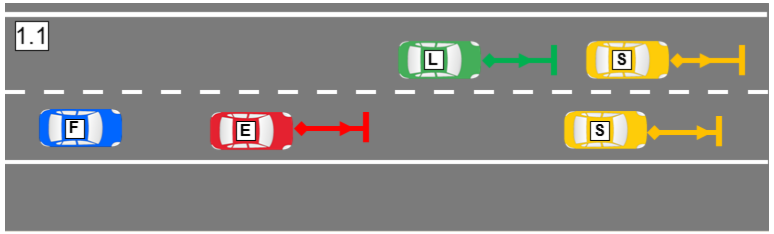
1.1.1.1 L decelerates

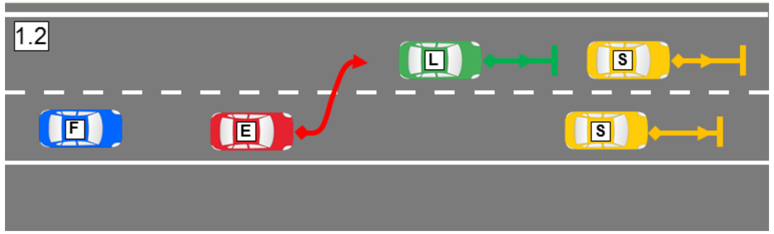
Scenario branching: Specification by two scenarios



1. E follows L on the right lane
   S decelerates
   L changes lanes

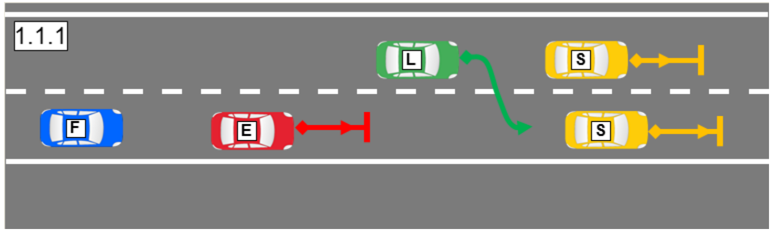
1.1 E decelarates
L decelerates

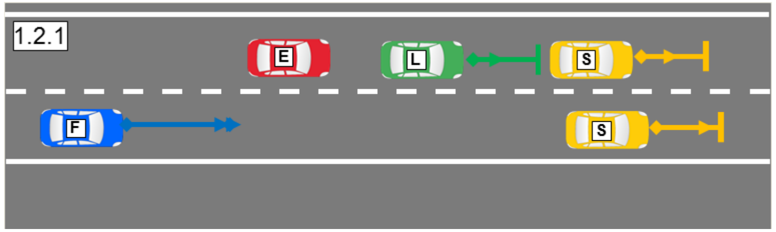




1.2 E changes lanes
L decelerates

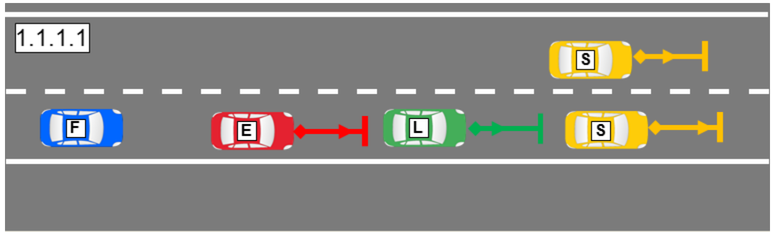
1.1.1 L changes back



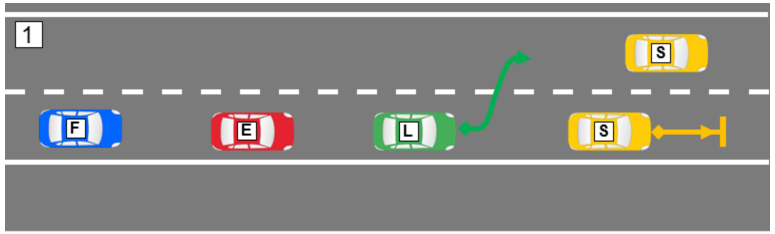


1.2.1 L decelerates hard
F accelerates on right lane (closing gap)

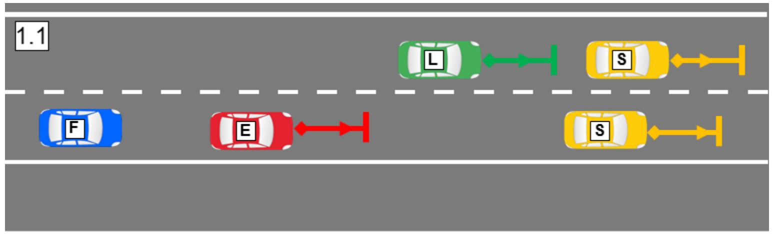
1.1.1.1 L decelerates



Scenario branching: Specification by two scenarios

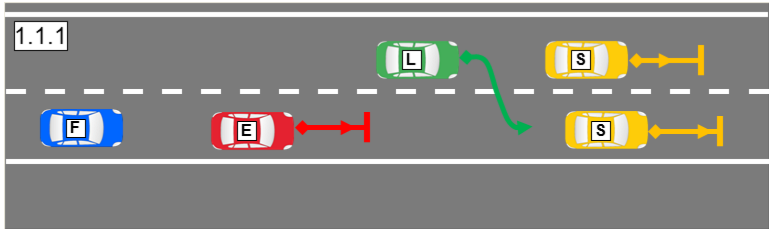


1.1 E decelarates
L decelerates

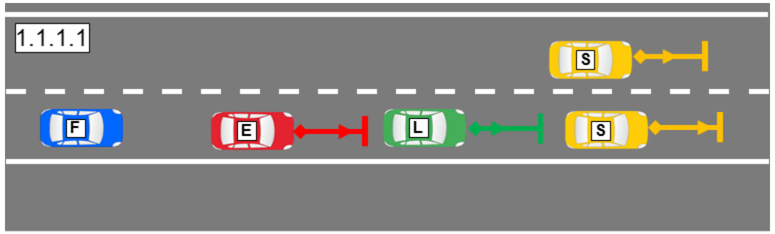


IF [E changes lanes] THEN BREAK

1.1.1 L changes back

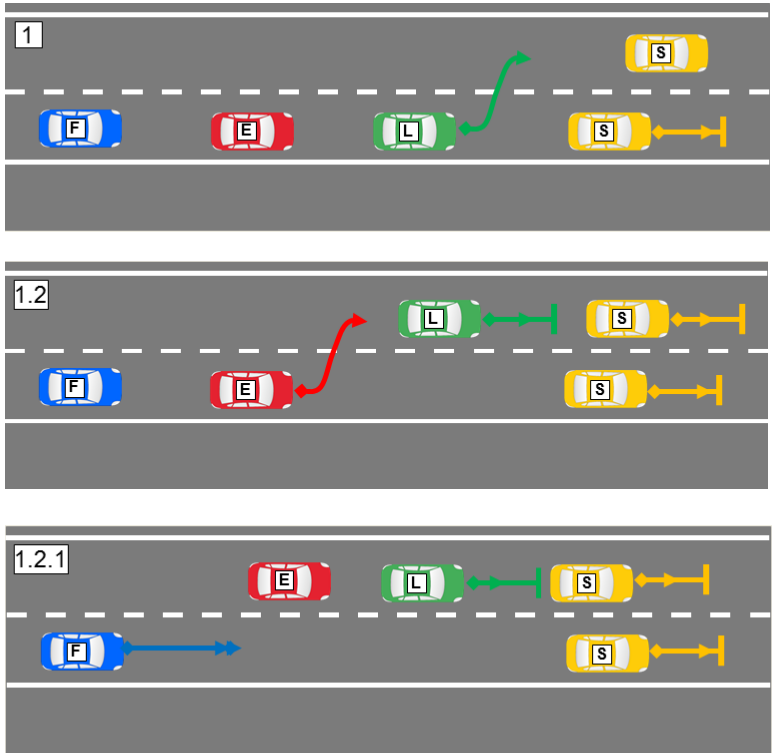


1.1.1.1 L decelerates



Scenario branching: Specification by two scenarios

IF not([E changes lanes]) THEN BREAK



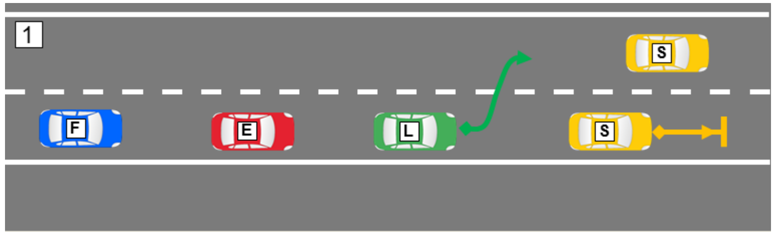
1. E follows L
   S decelerates
   L changes lanes

1.2 E changes lanes
L decelerates

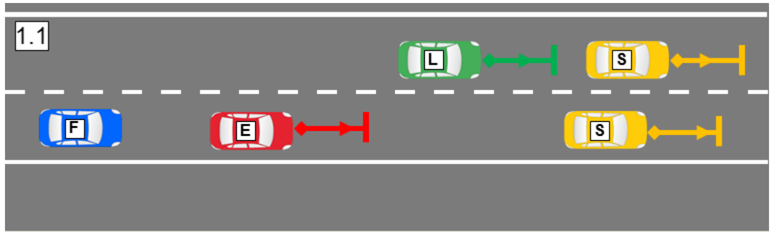
1.2.1 L decelerates hard
F accelerates on right lane (closing gap)

Scenario branching: Specification by two scenarios

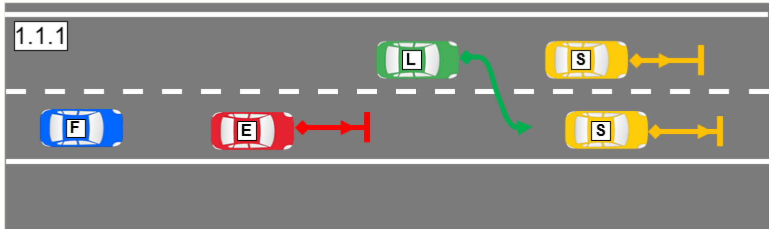
1. E follows L
   S decelerates
   L changes lanes



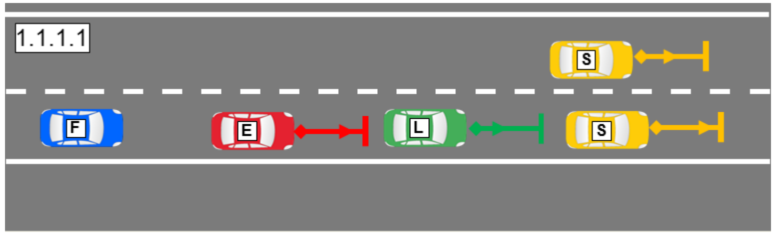
1.1 E decelerates
L decelerates

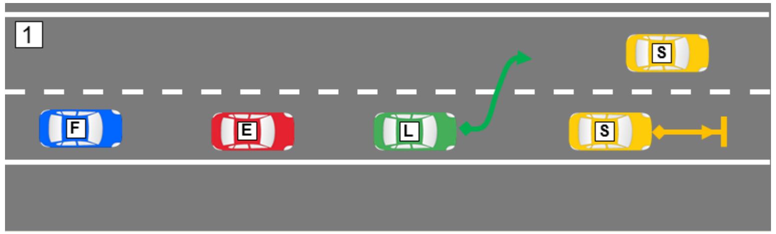


1.1.1 L changes back

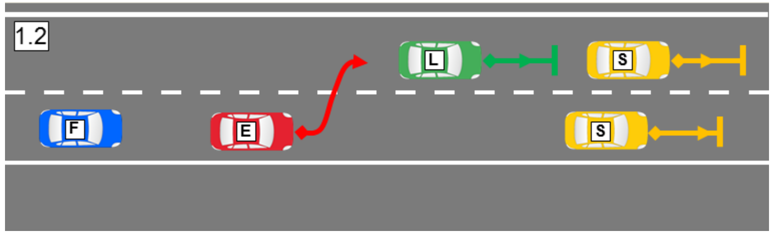


1.1.1.1 L decelerates

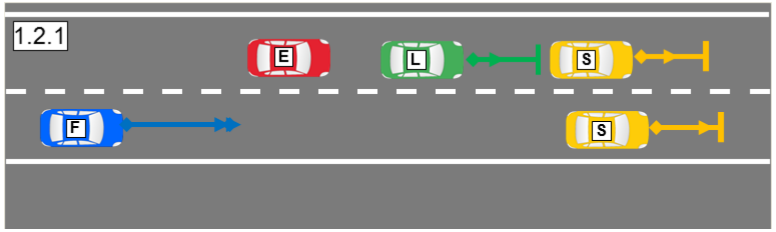




1. E follows L
   S decelerates
   L changes lanes



1.2 E changes lanes
L decelerates



1.2.1 L decelerates hard
F accelerates on right lane (closing gap)

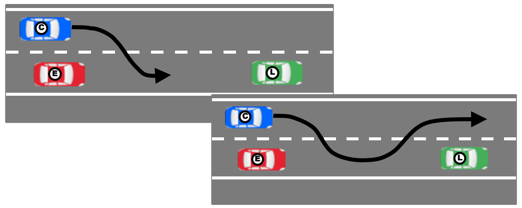
Different logical scenarios are distinguished by **different discrete actions** of E (and the other vehicles, of course).

Not a formal definition - yet

DLR

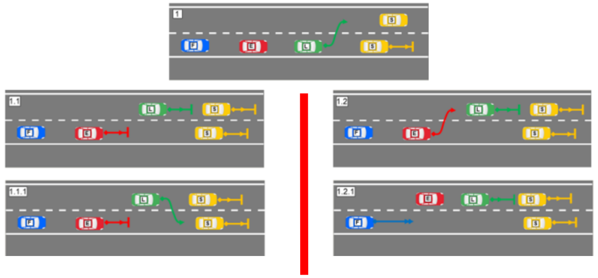
Logical scenarios as test specification

1. Capture all dynamic evolutions in discrete event structures (functional scenarios)



Functional scenarios

2. Extract linear evolutions by splitting branches



Linear scenarios

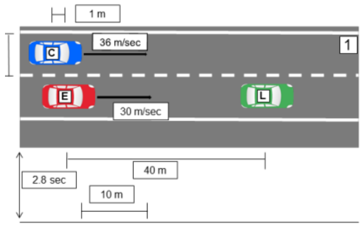
3. Formalize linear evolutions in parameterized programs (logical scenarios)

0. L: constant drive
   T: constant drive
   C: lane following with goal constellation depending on (C, T, E)

1. L: lane following, decelerating
   T: constant drive
   C: lane follwing with goal constellation depending on (C, T, E)

Logical scenarios

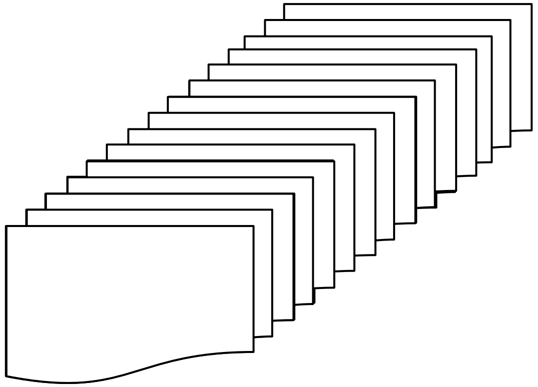
4. Instantiate scenarios for complete set of test cases



Concrete instances

Computing the risk

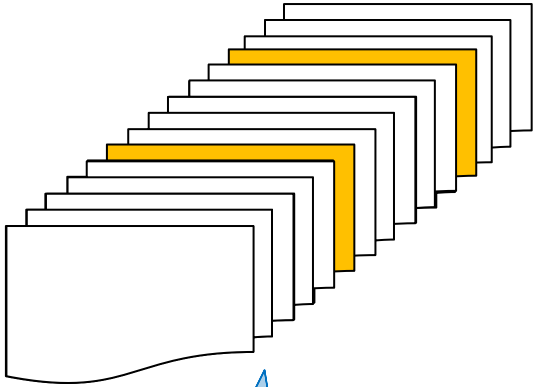
- List all hazards
- Determine
  - Exposure
  - Criticality
  - Severity



Hazard	E	C	S	Risk
…				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h				
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h				
…				
…				
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h				
…				
Sum				

Computing the risk

- List all hazards
- Determine
  - Exposure
  - **Criticality**
  - Severity



Formalized scenario descriptions enable automated test case generation

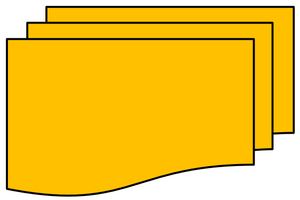
Splitting scenarios helps in keeping test cases disjoint

Determine values by automated simulation

Hazard	E	C	S	Risk
…				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h		0.23		
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h		0.12		
…				
…				
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h		0.15		
…				
Sum				

Computing the risk

- List all hazards
- Determine
  - Exposure
  - **Criticality**
  - Severity



- Extract **relevant row sets**

...

Hazard	E	C	S	Risk
…				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h		0.23		
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h		0.12		
…				
…				
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h		0.15		
…				
Sum				

Computing the risk

- List all hazards
- Determine
  - **Exposure**
  - Criticality
  - **Severity**

- Extract relevant row sets

- Detailed **analysis of risk** in critical scenarios

…

Hazard	E	C	S	Risk
…				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h	0.13	0.23	0.8	**0.239**
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h	0.02	0.12	1.3	**0.003**
…				
…				
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h	0.01	0.15	1.4	**0.002**
…				
Sum				

Computing the risk

- List all hazards
- Determine
  - Exposure
  - Criticality
  - Severity

- Extract relevant rows

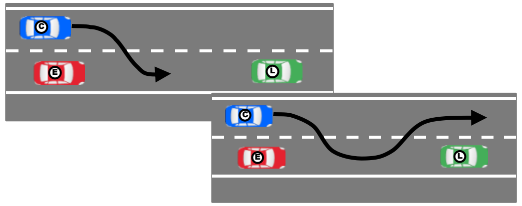
- Detailed analysis of risk in critical scenarios

- Sum up for **aggregated risk chart**

Hazard	E	C	S	Risk
…				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h	0.13	0.23	0.8	**0.239**
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h	0.02	0.12	1.3	**0.003**
…				
…				
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h	0.01	0.15	1.4	**0.002**
…				
Sum				

Conclusion

1. Capture all potentially critical evolutions in functional scenarios



Functional scenarios

2. Formalization of functional scenarios in precisely defined logical scenarios using maneuver macros

0. L: constant drive
   T: constant drive
   C: lane following with goal constellation depending on (C, T, E)

1. L: lane following, decelerating
   T: constant drive
   C: lane follwing with goal constellation depending on (C, T, E)

Split scenarios

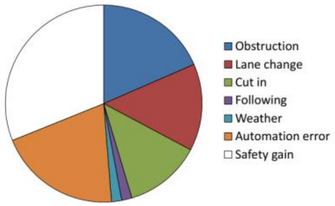
Logical scenarios

3. Identify all critical scenarios by systematic testing

Hazard	E	C	S	Risk
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h	0.13	0.23	0.8	**0.239**
...				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h	0.02	0.12	1.3	**0.003**
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h	0.01	0.15	1.4	**0.002**

Concrete critical scenarios

4. Build the risk chart by analyzing and rating the critical scenarios



Risk chart

Contact info

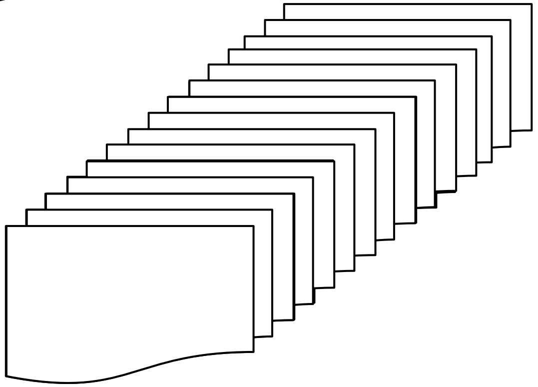
PD Dr. Hardi Hungar
German Aerospace Center
Institute of Transportation Systems
hardi.hungar@dlr.de

Risk computation

Few valid data available

- List all hazards
- Determine
  - Exposure
  - Criticality
  - Severity

Can be measured by testing



Only rough models available

- Sum up for overall risk

Hazard	E	C	S	Risk
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 100 km/h		0.00		
…				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h		0.12		
…				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 130 km/h		0.00		
…				
…				
…				
Sum				