FRAV - Annex
An Approach to Defining Codified Rules of the Road

This Annex presents a framework for codifying the rules of the road to govern the behaviour of ADSs. The approach may be used to define *"good behaviour"* to inform validation and verification processes (including for scenario-based testing) for nominal scenarios.

Current rules of the road (for human drivers) have three components:

Rule of road (for human drivers) = *Operating condition + Behaviour competency + Assumptions (implicit)*

Operating conditions include both ODD aspects and vehicle states (e.g., system failures, hardware failures etc.). Every set of traffic laws or behaviour rules (for human drivers) defined in any country are based on an understanding of the expected behaviours of human drivers. As a result they do not explicitly define all aspects of the expected driving behaviour but can be argued to include "implicit assumptions" based on this understanding.

Following the process (illustrated in section 4), **a "codified" rule of the road for an automated driving system, will also have three components:**

Codified Rule of road = *Operating condition + Behaviour competency + Driving decisions*

The process of codification helps identify where "implicit assumptions" about driving behaviour are present in the rules for human drivers. **The codified rules of the road help to turn "undefined" attributes in the rules of the road (for human drivers) to "defined" attributes in the codified "rules of the road".**

Taking an example of the UK road rules where behaviour (for human drivers) is governed by the Highway Code (HC)[1], the methodology is further explained. UK's Highway Code Rule 195 states (Zebra crossing):

Rule 195: "As you approach a zebra crossing: look out for pedestrians waiting to cross and be ready to slow down or stop to let them cross; you MUST give way when a pedestrian has moved onto a crossing"

[1] UK Highway Code: https://www.gov.uk/guidance/the-highway-code

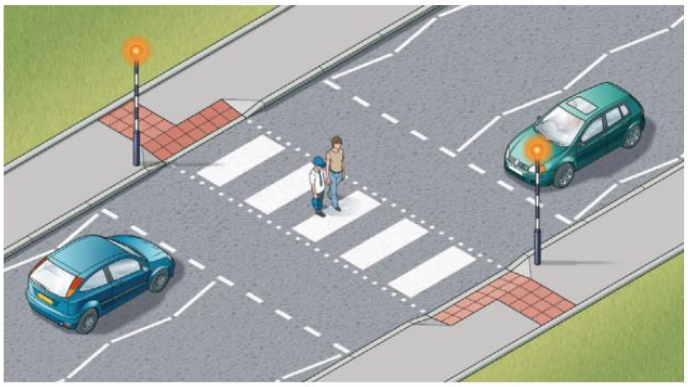

Figure 1: Example of zebra crossing from UK's Highway Code:
Source: https://www.gov.uk/guidance/the-highway-code/rules-for-pedestrians-1-to-35#rule19

From this rule, one can extract the "operating condition or ODD" variables, as well as the behaviour competencies. ***"Zebra crossing"*** and ***"pedestrian"*** define the ***operating condition***; and ***"slow down or stop"*** defines the ***behaviour competency***. However, the rule doesn't mention for how long the vehicle should be stopped, or when it is considered safe to proceed again. There is an "implicit assumption" made based on typical human (the driver behaviour), and it is not considered necessary for the rule to define this. However, for an ADS, such assumptions how long the vehicle is stopped for, and when it moves off again will be determined by the automated driving system and its analysis of the relevant parameters specific to that situation and will need to be specified. For every concrete scenario being tested, the driving decisions exhibited by ADS will need to be explainable.

Figure 2 illustrates this process. After following the codification process of defining the "rules of the road", there will be no underlying "assumptions" (see section 4). Furthermore, for all areas or jurisdiction or country, there will be a minimum set of behaviour code rules which will have consistent *"driving characteristics"* – the base or common set of rules of the road (for ADS).

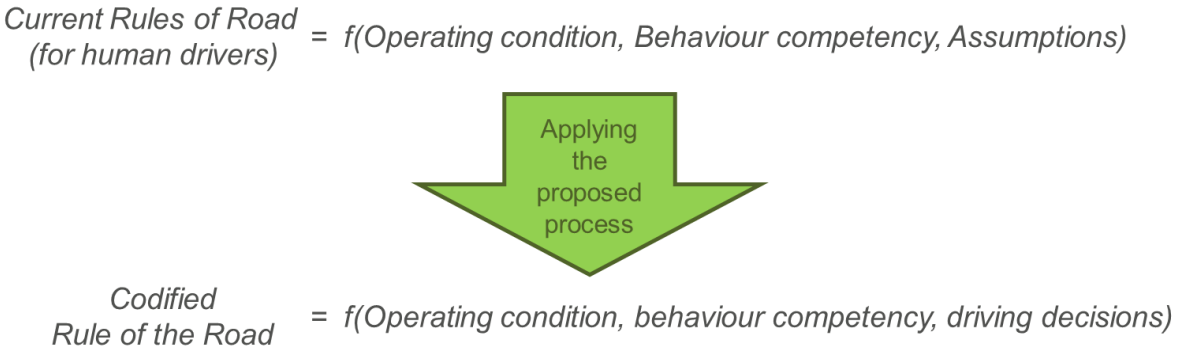

Figure 2: Converting current rules of the road (for human drivers) to codified rules for ADS

1. Codification methodology

The codification methodology is a four-step process:

- **Step 1: Identify terms and construct a vocabulary:** The natural language text of the rule is analysed and words that are associated with the ODD or behaviour of actors in the rule are identified. These terms taken together are used to identify the component of the rule that can be codified

- **Phase 2: Identify unspecified terms:** Some terms are unclear because they are not unequivocal or absolute and therefore require clarification. In some cases, these terms are codified as is, when a meaning can be inferred, while in others, comments are provided to highlight why the terms are not defined, and how they may be elaborated.

- **Phase 3: Query / Update/ Add ODD and Behaviour terms:** Terms defining predicates (representing facts whose truth may be evaluated) and functions (representing non-Boolean properties – such as ADS attributes, action labels) are identified. The codified rule will consist of these predicates and functions. The outcome of Phase 3 is an intermediate rule that is in its minimal form.

- **Phase 4: Express rule in first order logic:** For each rule of the road, a single codified rule, or a set of rules are written. The predicates and functions identified in Phase 3, together with the structure of constraints from Phase 1 are used to construct the rule(s). The output of Phase 2 provides insights concerning the rule and gaps that exist in its codification. Phase 4 uses the vocabulary to identify which sub-rules are to be converted to First Order Logic and then perform the conversion.

2. Codification Example: Rule 162 (of the UK's Highway Code)

Rule 162 of the UK's Highway Code is used to illustrate the four phases of the codification process. The rule is stated below.

Rule Text
Before overtaking you should make sure *- the road is sufficiently clear ahead* *- road users are not beginning to overtake you* *- there is a suitable gap in front of the road user you plan to overtake.*

The following sections take this rule through each phase, explaining how each component of the codification process works.

2.1. Phase 1: Identify Terms and Construct a Vocabulary

In this phase, terms are identified to generate a vocabulary of predicates. The terms extracted from the ruleset are those relevant to:

- ODD (Scenery, Actor, Environment) & Behaviour

- Rule/Parameter qualifiers: such as '*when', 'limit', 'does not mean',* etc. which affect the meaning of the statement
- Other important terms that need to be reviewed and clarified in Phase 2

Sub-rules that do not contain rules that are actionable for an ADS are not codified.

Example: Rule 162 (Phase 1: Identify Terms)

The rule is re-stated below highlighting important terms.

Before <u>overtaking</u> you should **make sure**

- the **<u>road</u>** is **<u>sufficiently clear ahead</u>**
- **<u>road users</u>** are **not beginning** to **<u>overtake</u>** **<u>you</u>**
- there is a **<u>suitable gap</u>** **in** **<u>front</u>** **of the** **<u>road user</u>** **you plan to** **<u>overtake</u>**.

Terms that are ODD and behaviour related are in bold and underline, while other terms that are relevant to giving the rule meaning are in bold.

2.2. Phase 2: Identify Unspecified Terms

Using domain specific concepts, each minimal statement is fleshed out to clarify any underspecified (unquantified) terms, ambiguous or abstract terms. For instance, if a broad statement is made requiring further qualification, such as, "unsafe road layouts or junctions", it must be identified that a further qualification is necessary and what this may look like. In this case, it is important to specify which road layouts or junctions are unsafe. This may be done using relative terms – for instance, with respect to the ODD of the vehicle; or in absolute terms – enumerating a list of unsafe road layouts and junctions. This should however not be confused with a rule that expresses a general requirement, where the absence of specification of an ODD concept makes the rule applicable to all instances of that concept. For instance, if road type or weather condition is not qualified in the rule's text, then the rule is applicable to all roads and weather conditions. It is only the vague components of a rule that must be fleshed out to make the rule complete from the perspective of an ADS.

Example: Rule 162 (Phase 2: Identify Unspecified Terms)

This phase involves the identification of the terms that are unclear and that need to be clarified. These are the terms that are absolute so make the rule subjective and hence need to be investigated and resolved.

From the example above, the terms that do not remain fully specified are as follows:

Term	Specification Required
Sufficiently clear ahead	How is sufficiently clear ahead defined? *Time To Collision (TTC) of any oncoming vehicle evaluated against time for maneouvre*
Suitable gap	What is a suitable gap? Twice the *stopping distance* may be a good definition to consider.
Overtaking is an action that is applicable to vehicles that are ahead of the ego	This is an assumption that is understood by a human reader.

2.3. Phase 3: Identify Predicates and Functions

In this phase, each rule is reduced to its minimal form by identifying predicates and functions that form the core facts of the rule. These are the terms that provide meaning to the rule. Once terms are identified, it is important to establish which terms are synonyms or antonyms. For terms that are synonymous, a single term is chosen to be used in place of all terms that are equivalent in meaning to it. In this manner a normalized vocabulary may be constructed.

This exercise focuses on the key aspects of the rule of the road and eliminates the unimportant phrases or terms that cannot be actioned as part of this process.

Example: Rule 162 (Phase 3: Identify Predicates and Functions)

The non-highlighted terms are removed and only terms that are important to the meaning of the rule are kept.

Before overtaking make sure

- **road sufficiently clear ahead**
- **road users not beginning to overtake you**
- **suitable gap in front of the road user you plan to overtake**.

The terms identified are converted into predicates. For Rule 162, we construct the following predicates:

Predicate	Description
isEgo(x)	x is the Ego
isAhead(x,y)	x is ahead of y
isOtherRoadUser(x)	x is a non-Ego object
isSufficientlyClearAhead(x)	x is sufficiently clear ahead
isOvertaking(x,y)	x is overtaking y
hasSuitableGapAhead(x)	There is a suitable gap ahead of x
canOvertake(x,y)	x can overtake y

isOnRoadLane(x,y)	x is on road-lane y

2.4. Phase 4: Express Rule in First Order Logic

Each rule is then expressed using the normalized vocabulary in first-order logic. The normalized vocabulary is a collection of predicates and predicate parameters, representing concepts in the rule of the road, that are re-used across the codified ruleset.

Phase 4: Express Rule in First-Order Logic

Rule 162 is a rule that identified whether an overtake manoeuvre can be performed. If the conditions of the rule are true, then an overtake manoeuvre can be acted upon, otherwise it must be abandoned. Further, this rule implicitly also identifies which actor the ego can overtake.

For ease of understanding, the rule may be broken down into four logical statements, that are logically related, with the relationship being stated as the last rule. The predicates that were produced as an outcome of Phase 1 are used to construct the logic specification for the rule.

The parameters for the rules: the ego vehicle (x), the lane (y), other actor (w), and actor being overtaken (z).

The rules are as follows:

*Rule (**a**):*	**isEgo**(*x*)	x is the ego
*Rule (**b**):*	**isOnRoadLane**(*x,y*) ∧ **isClearAhead**(y)	x is on road-lane y and y is clear ahead
*Rule (**c**):*	**isOtherRoadUser**(*w*) ∧ **isOvertaking**(*w,x*)	w is overtaking x
*Rule (**d**):*	**isAhead**(z,*x*) ∧ **hasSuitableGapAhead**(*z*)	suitable gap in front of the road user you plan to overtake.
The Rule	(**a**) ∧ (**b**) ∧ (¬**c**) ∧ (**d**) → **canOvertake**(*x,z*)	

The symbol "¬" when used as a prefix to a logic sentence (such as "c" which denotes Rule (c)) indicates the negation of the logic sentence. In this context, in English, the rule may be read as: If "a" is true, and "b" is true, and "c" is false, and "d" is true, then x can overtake z. The truth asserted is hierarchically asserted within the sub-rules.

3. Codification Example: Rule from the Vienna Convention

The rule is stated below (Chapter 2 – Rules of the Road – Article 11 (Overtaking – 11)).

VC Rule Text

A vehicle shall not overtake another vehicle which is approaching a pedestrian crossing marked on the carriageway or signposted as such, or which is stopped immediately before the crossing, otherwise than at a speed low enough to enable it to stop immediately if a pedestrian is on the crossing.

The following sections take this rule through each phase, explaining how each component of the codification process works.

3.1. Phase 1: Identify Terms and Construct a Vocabulary

Example: VC Rule (Phase 1: Identify Terms)

The rule is re-stated below highlighting important terms.

> A vehicle **shall not** **overtake** **another vehicle** which is **approaching** a **pedestrian crossing** marked **on** the **carriageway** **or** **signposted** as such, **or** which is **stopped** **immediately** **before** the **crossing**, **otherwise** than at a **speed** **low enough** to **enable** it to **stop** **immediately** **if** a **pedestrian** is **on** the **crossing**.

Terms that are ODD and behaviour related are in bold and underline, while other terms that are relevant to giving the rule meaning are in bold.

3.2. Phase 2: Identify Unspecified Terms

Example: VC Rule (Phase 2: Identify Unspecified Terms)

From the example above, the terms that remain underspecified are as follows:

Term	Specification Required
Immediately	How is immediately defined? *A distance* may be used to define this.
Low enough	What speed is considered low enough? This could be a function of distance to the pedestrian, or an absolute threshold.
Overtaking is an action that is applicable to vehicles that are ahead of the ego	This is an assumption that is understood by a human reader.

3.3. Phase 3: Identify Predicates and Functions

Example: VC Rule (Phase 3: Identify Predicates and Functions)

The non-highlighted terms are removed and only terms that are important to the meaning of the rule are kept.

Shall not overtake another vehicle

- **approaching pedestrian crossing on carriageway or signposted,**
- **or stopped immediately before crossing**,

otherwise speed low enough enable stop immediately if pedestrian on crossing.

The terms identified are converted into predicates. For the VC Rule, we construct the following predicates:

Predicate	Description
isEgo(x)	x is the Ego
canOvertake(x,y)	x can overtake y
isApproaching(x,y)	x is approaching y
isPedestrianCrossing(x)	x is a pedestrian crossing
isCarriageway(x)	x is a carriageway
isSignposted(x)	x is signposted
isStopped(x)	x is stopped
isAhead(x,y)	x is ahead of y
hasSpeed(x,y)	x has speed y
isLowEnoughSpeed(x,y)	x is a low enough speed for action y

3.4. Phase 4: Express Rule in First Order Logic

Phase 4: Express Rule in First-Order Logic

The rule determines overtaking behaviour for a vehicle that is close to a pedestrian crossing. The rule contains conditions that would prevent a vehicle from overtaking another, but simultaneously provides an exception, that of being slow enough to stop. Further, the ability of the vehicle to stop is independent of whether there is an actor (such as a pedestrian) on the crossing. The rule makes references to the vehicle having a slow enough speed to stop immediately, which has been identified as an ambiguous phrase and represented as a predicate in Phase 3. To represent the action of stopping immediately, we use the constant “STOP_IMM”.

For ease of understanding, the rule may be broken down into four logical statements, that are logically related, with the relationship being stated as the last rule. The predicates that were produced as an outcome of Phase 1 are used to construct the logic specification for the rule.

The parameters for the rules: the ego vehicle (x), the other actor (y), the pedestrian crossing (w), the carriageway (c), the speed of the ego (s).

The rules are as follows:

*Rule (**a**):*	**isEgo**(*x*) ∧ **isOtherRoadUser**(y)	x is the ego and y is the other vehicle
*Rule (**b**):*	**isPedestrianCrossing(**w**)** ∧ (**isCarriageway(**c**)** ∨ **isSignposted**(w))	w is a pedestrian crossing and (c is a carriageway or w is signposted)
*Rule (**c**):*	**isApproaching(**y,w**)** ∨ **isAhead(**w,y**)**	y is approaching w, or w is ahead of y
*Rule (**d**):*	**hasSpeed(**x,s**)** ∧ **¬isLowEnoughSpeed(**s,STOP_IMM)	x has speed s, and s is not a low enough speed to stop immediately.
The Rule	(**a**) ∧ (**b**) ∧ (**c**) ∧ (**d**) → ¬**canOvertake**(*x,z*)	

The symbol "¬" when used as a prefix to a predicate indicates the negation of the predicate. In this context, in English, the rule may be read as: If "a" is true, and "b" is true, and "c" is true, and "d" is true, then x cannot overtake z. Note that the exception condition, that of being slow, is used in its negative form to assert that the vehicle cannot overtake, since this is explicit in the rule. It is left to interpretation if a positive rule, specifically allowing the vehicle to overtake is necessary. If so, a new rule that allows a vehicle to overtake must be written. This would depend on the interpretation of the rule.

Bibliography:

1. NHTSA - A Framework for Automated Driving System Testable Cases and Scenarios: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13882-automateddrivingsystems_092618_v1a_tag.pdf
2. Waymo's Safety Report (see behaviour competencies): https://storage.googleapis.com/waymo-uploads/files/documents/safety/2021-03-waymo-safety-report.pdf
3. CETRAN - Scenario Categories for the Assessment of Automated Vehicles : https://cetran.sg/wp-content/uploads/2020/01/REP200121_Scenario_Categories_v1.7.pdf