# Semi-Automated Synthesis of Driving Rules

Diego Ortiz Barbosa [a,*], Leilani Gilpin [b] and Alvaro A. Cardenas [b]
[a] *Computer Science and Engineering, University of California, Santa Cruz, CA, USA*
*E-mail: dortizba@ucsc.edu*
[b] *Computer Science and Engineering, University of California, Santa Cruz, CA, USA*
*E-mails: lgilpin@ucsc.edu, alacarde@ucsc.edu*

**Abstract.** Autonomous vehicles must operate in a complex environment with various social norms and expectations. While most of the work on securing autonomous vehicles has focused on safety, we argue that we also need to monitor for deviations from various societal "common sense" rules to identify attacks against autonomous systems. In this paper, we provide a preliminary approach to encoding and understanding these common-sense driving behaviors by semi-automatically extracting rules from driving manuals. We encode our driving rules in a formal specification and make our rules available online for other researchers.

Keywords: Autonomous Vehicles, Rulebooks, Automation

## 1. Introduction

Neuro-symbolic AI, as a relatively new approach within AI, seeks to build bridges between symbolic reasoning and the power of neural networks, the two main approaches that traditionally encompass the field. The potential of reconciling these approaches brings the possibility of building systems capable of learning from data while inferring desired behaviors due to the presence of a symbolic reasoning layer capable of giving decisions explainability that neural networks usually struggle. This approach is helpful for cyber-physical systems such as autonomous vehicles that can struggle in real-world scenarios that conflict with human actors[1].

Autonomous vehicles operate in a safety-critical, complex, open-world environment. Human drivers can abstract and apply common sense rules in this ever-changing environment, from avoiding traffic accidents to driving according to societal expectations. Due to this abstract reasoning ability, humans can reason in situations where we can effectively balance the need to reach our objective while successfully considering their impact on a broader context. For example, even if you see a green light, you should not proceed if your action will result in a blocked intersection. These rules tend to be well-understood by human drivers. In fact, these rules are documented in the form of "rule books" or driving handbooks, which prepare new drivers for driving tests in different states, territories, and countries.

Although exploration of rules of the road have previously been examined [1], previous work has mainly focused on the extraction of rules regarding solely safety concerns [2] . However, there is more to driving than safety assurances; as context changes, so do driving patterns; thus, autonomous vehicles might want to behave differently depending on the region in which the vehicle operates. For example, some regions might allow a turn-right driving maneuver at an intersection with a red light, while others might prohibit this. Therefore, the types of driving rules

*Corresponding author. E-mail: dortizba@ucsc.edu.
[1]An example of inconsistent behavior is the Tesla stopping behavior in front of a tunnel: https://www.businessinsider.com/tesla-stops-tunnel-pileup-accidents-driver-says-fsd-enabled-video-2023-1.

are location dependent, and each state has its own "driving handbook" or "rule book". In this work, we develop a framework to extract driving rules from driving manuals in an effort to understand these rules better and categorize them to begin a discussion of how certain rules need to be monitored to detect attacks and how some of the abstractions can help improve autonomous driving systems beyond safety.

In particular, developing this rule set can impact current perception or planning modules used to learn and calculate paths along the AV's route. Using these rules on top of the modules of the AV stack, such as perception and planning models, using this neuro-symbolic approach, vehicles can *monitor* for abiding by safety rules, laws, and best practices and correct their behavior while having explainability. For example, if a self-driving car is trained to drive in California, our system can monitor that the self-driving car abides by California-specific rules. The vehicle's location can also change any autonomous agent's prior beliefs. For example, drivers in Arizona have to be aware of dust storms–e.g., the Arizona handbook has the rule, "if you encounter a severe dust storm, reduce your speed immediately." We also find that some of the rules in driving manuals are recommendations due to different conditions of the car; for example, the rule "if you are pulling a trailer, wind currents can cause your vehicle to jackknife." These rules can be used to adjust a prior belief of any autonomous driving vehicle so it drives more cautiously. In the event of an unexplained anomaly (e.g., a jackknife), the anomaly detection agent can check if there is an explanation for it (the car is pulling a trailer) or not (a failure or potential attack). Finally, we find that certain driving rules also depend on additional equipment in the vehicle; for example, "if you have a transponder as you approach a toll plaza, look for and follow signs with the purple logo of E-ZPass."

To better understand these rules, we present an automated rule extraction framework that encodes driving rules in a formal specification that allows for inferences, portability, and adaptability. Our system automatically extracts 709 rules, relevant to the behavior of several actors on the road. As part of the contributions of this paper, we make these extracted rules publicly available online[2]. We also perform *meta analyses* with the rule set to cluster driving rules according to different types of desired features.

## 2. Motivation

Deployment of AVs in urban areas continues to develop. Recently the California Public Utilities Commission allowed AV companies Cruise and Waymo to expand their operation of autonomous taxis through the streets of San Francisco [3], allowing the companies to offer their services 24/7 through the city. This expansion means that more portions of the public can interact with this new driving technology, and an increasing number of incidents involving AVs have been reported. Just a couple of days after the expansion, Cruise vehicles found themselves stuck seemingly without a reason in a narrow street, bringing all traffic to a standstill. [4] In another incident, a Cruise AV blocked the road for an ambulance trying to get the victim of an accident to a hospital, and according to the San Francisco Fire Department, it is not the first time that this kind of incident occurred.

Another issue with the integration of AVs is how society perceives and responds to the introduction of this technology into their daily life. Resistance to the idea of autonomous taxis from groups of individuals has become concrete actions that intend to sabotage or even damage the operation of AVs. Activist groups like Safe Street Rebel [5] use traffic cones to disable AVs throughout the city showing their disapproval of the expansion of the vehicles. However, there are reports of people misusing AVs [4] and certain actions from opponents of the AVs have turned violent for example a man attacking an Avs camera with a hammer. [6]

As the community pushes back against AVs and the problem with interactions between them and other non-robotic users on the road become more frequent, it is important to consider the implications that an AV has on the community it is being introduced into. Adapting AVs to operate alongside humans with different backgrounds implies that the vehicle planner should follow the rules established by each territory. Adapting these rules is a complex challenge as there is an abstract representation: it is not straightforward to translate them to deterministic values or behaviors that we as humans may call "common sense." Several efforts exist to try to reconcile this barrier with industry working with regulators to generate specifications, however, any specification cannot be universally applied as communities differ from one another with discrepancies arising from interpretations [7]

[2]https://github.com/RollingBeatle/rule-analyzer/tree/main/results

A successful adaptation of the rules of the road must address challenges on the politics of AVs such as exploring their impact on users' lives on different fronts like their safety, how infrastructural changes for AV adoption impact communities, and their impact on the labor market. [8] Considering safety as the most important part of AVs objectives, AVs have to deal with handling ethical issues and trade-offs that may put objectives at odds with each other (liability vs harm) [9] making the interpretation of the rules of the road central in the development of safe and efficient AVs.

## 3. Related Work

Our rule extraction system extracts driving rules from natural language text. Language explanations have been shown to help in classification tasks [10] by allowing annotators to provide supervision to a classifier via natural language explanations. Previous work on learning based on explanations [11] was demonstrated in two planning domains. While this work focuses on updating environment models, our contribution is to extract rules directly in a real-world environment. To the best of our best knowledge, this is a new contribution of using NLP and rule-based systems to learn new rules in the context of autonomous driving.

Our rule extraction system relies on a set of rules and a representation of primitive actions in autonomous vehicles. These primitive actions are outlined in prior work [12]. We extend this work by automatically extracting and analyzing safety rules from existing driving manuals. Our work is also related to the concept of "rule books" for formalizing AV behavior as a hierarchy of specifications [13]. While our goal to generate sets of rules is similar, our main objective in this paper is to facilitate the extraction of these driving rules from the driving authority in each region, make them publicly available, and design a comprehensive classification, as they represent the best practices each state has identified over several decades.

Our rule learning system is a complement to existing interlock systems [14] or formal logic [15, 16]. The key difference is that our rule language is interpretable and abstract. Our rules are represented in a simplified language (symbolic triples), and the language is abstract, using qualitative descriptions in lieu of numerical constraints. Our extracted rules can be used in these systems, and we will explore this direction in future work.

The format and representation of our extracted rules are inspired by Web standard rules and written as an object-oriented Python class. The `IF(?x), THEN(?y)` format is similar to the abstract syntax of Rule Interchange Format (RIF) [17], which is a W3C standard. Initial iterations of the rule-learning system [18] were based on Prolog and Python syntax for Prolog[3], but these languages are not adaptable and require a Prolog-style reasoner. Therefore, we wrote our new rule language in Python so that we could utilize our own NLP parsing. Other suitable rule formats include constraint logic programming languages like Datalog [19]. However, this is most applicable to database systems, which is outside of the scope of our work.

Finally, one goal of this work is to strengthen anomaly detection in autonomous vehicles by adding common sense rules. Anomaly detection is a well-studied field in the realm of machine learning [20], although learning from those anomalies is an open area. Learning from errors is promising for transition repair [21], but this is specific to the domain of robotics. Other work has strengthened anomaly detection for overcoming blind spots in autonomous vehicles [22], but this approach relies on a human in the loop, where control can be transferred to a human operator if the autonomous operator needs help. Our goal is to create the first step towards creating a set of rules that can later be used (in future work) as a specification that we can then monitor with robust anomaly detection.

## 4. Proposed Framework

We propose a semi-automated methodology to extract and encode driving rules from the driving handbooks from different states and regions. We analyze the content of the manuals using natural language processing to automate the extraction of rules and then encode them in an abstract syntax for first-order logic. We then manually check the rules and refine a subset to provide a better interpretation of them. See Figure 1 for a summary of our pipeline.

[3]https://pypi.org/project/pyprolog/

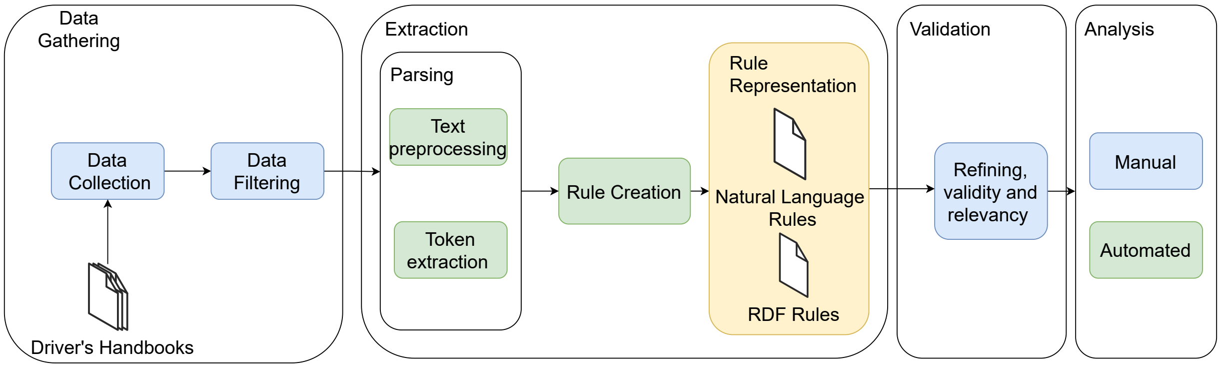


Fig. 1. Overview of our analysis methodology. We use NLP to extract rules into a formal representation, and then manually refine a subset of the rules.

## 5. Data Gathering

### 5.1. Data Collection

We processed the driving manuals of the 15th most populated states in the U.S., which correlate to the states with the most active drivers. In addition, to obtain a global perspective on driving rules, we also selected the six states of Australia.

We collected the driving manuals by downloading them from each state's department of motor vehicles website. Each driver handbook is downloaded locally as a PDF file. We collected our dataset between June 20th and July 24th 2022. Two representative driving manuals (from Massachusetts and California) are available in our GitHub repository[4] so other researchers can test our tools.

### 5.2. Data Filtering

To analyze the PDF files, we transformed the file into a text format. We used PyPDF2[5], a free and open-source Python library that can retrieve text from PDF files.

PyPDF2 was suitable for processing 16 of our selected manuals , but we found that some of the manuals were formatted in ways that PyPDF2 did not parse effectively. We found three sources of error: (1) Orientation (when the driving manuals are in landscape mode), (2) Protection (the driving manuals prevent text scraping), and (3) Images and diagrams (when the driving manual is mostly described in diagrams and flow charts). As a result, we could not process the handbooks of two U.S. states (Georgia, and Michigan) and three Australian states (New South Wales, Victoria, and Northern Australia).

Furthermore, driving manuals cover a wide range of topics, such as the process for obtaining a driving license or regulations for responding to traffic infraction tickets. Since this information is not related to driving rules, we excluded several pages of the driving handbooks from our process; however, there were instances in which these rules emerged due to the particular structure of some manuals. In particular, we focus on the specific chapters or sections of the driver handbook that correspond to safety. These pages are manually selected. A summary of the pages that we feed our tools for each manual is shown in Table 1.

## 6. Rules Extraction

After specifying the pages of a driving manual, we use natural language processing and a novel rule extraction method to encode conditional propositions in an RDF triple-store format[6]. The output of the rule extraction method

[4]https://github.com/lgilpin/driving-rules
[5]https://pypi.org/project/PyPDF2/
[6]RDF: https://www.w3.org/RDF/

Table 1

Pages processed for each manual.

| Manual name | start page | end page | total # of pages |
|---|---|---|---|
| California | 33 | 102 | 69 |
| Texas | 29 | 63 | 34 |
| Florida | 34 | 64 | 30 |
| New York | 27 | 70 | 43 |
| Pennsylvania | 40 | 70 | 30 |
| Illinois | 30 | 90 | 60 |
| Ohio | 37 | 70 | 33 |
| New Jersey | 64 | 157 | 93 |
| Virginia | 7 | 27 | 20 |
| Oregon | 30 | 80 | 50 |
| Arizona | 27 | 70 | 37 |
| Massachusetts | 82 | 124 | 42 |
| North Carolina | 21 | 50 | 29 |
| Queensland | 64 | 153 | 89 |
| West Australia | 49 | 96 | 47 |
| Tasmania | 13 | 74 | 61 |

is a text file of rules with one rule per line.

### 6.1. Rule Format

Our rules are represented as strings in the format: `"IF(?x), THEN(?y)"`, where `(?x)` is the antecedent of the rule, and `(?y)` is the consequent of the rule. We represent the antecedent and consequent in two ways: (1) in natural language and (2) in a conjunction of a triple store representation.

This format allows the rules to be standardized, portable, and adaptable. The standardized format allows transformations, matching, and inference on the rules can be done systematically. For example, our rule list can be combined with a modus ponens inference method, and the antecedent and consequent of the rules can be matched without complicated parsing processes. Rules can also be translated into equivalent logical sentences.

Secondly, the portability of the rules allows a systematic encoding of driving safety expertise. This expertise is programming language agnostic: the rule list is usable whether the underlying system is in Python, C, JavaScript, etc. The rules do not have to be translated into different languages. And finally, the rules are adaptable; they can be altered, changed, and removed without affecting the underlying system. Whereas, if the rules were written in a software system, there would be dependencies between rules that would be to be altered.

### 6.2. Natural Language Processing

Rules are created by parsing the natural language text. Our system parses each page of text into a list of strings. Each string is one sentence. Then, the sentence is parsed into word tokens. If any of the rule keywords: `IF, THEN, BECAUSE` are in the sentence, then a rule is flagged to be made.

If any keywords are found in a sentence, the sentence is flagged as a potential rule, and we proceed to the next step of our analysis. Our rationale for using these keywords is the following:

- Syntax: Since we are creating if-then rules, the natural language rule description must contain an antecedent (an if clause) and a consequent (a then clause). If either the antecedent or the consequent is not found, then the rule extraction fails for the input.
- Soundness: Conditional propositions may be encoded in different natural language expressions, but looking for these might produce several false rules. Because our goal in this work is to get an initial sense of the diversity of driving rules, we focus on keywords that will produce If-Then rules with high specificity, even if we miss some.

We split each identified sentence into the *if-phrase* and the *then-phrase* corresponding to the antecedent and the consequent of the rule. First, the sentence is split by non-alphanumeric characters: `:,.` into a list of clauses. Then the *if-phrase* and *then-phrase* are set with the following rule-based method:

– If there are exactly two clauses, return the if-clause as the first item and the then-clause as the second item if `IF` is in the first clause. Otherwise, reverse the order of the clauses.
– If there is one clause, split the clause on `IF` and set the if-clause to the first part of the split and the then-clause to the last part of the split.
– If there are more than two clauses, then the if-clause is the first clause, and the then clause is the composition of the rest of the clauses.

After the *if-phrase* and the *then-phrase* are set, then the natural language representation is completed. The rule is returned as `IF(if-phrase), THEN(then-phrase)`. If the triple-store representation is flagged, then the `make_triples_from_phrase` function is called. This function makes a set of triples for the `if-phrase` and the `then-phrase`. The `make_triples_from_phrase` function receives an input phrase: a natural language phrase corresponding to the `if-phrase` or the `then-phrase`. The function searches the input phrase for a conjunction keyword: `and, that, not` or `or`.

The `make_triples_from_phrase` function is recursively called until no conjunction keywords are found. The function searches for conjunction keywords: `and, that, not` or `or` and performs the following:

1. If a conjunction keyword "and" or "that" is found in the input phrase, then an AND phrase is created: `AND(?x, ?y)`
2. If instead "or" is found in the phrase, then an OR phrase is created: `OR(?x, ?y)`
3. Otherwise if `NOT` or `NEVER` is found in the phrase, then a NOT phrase is created: `NOT(?x)`

The `(?x)` and `(?y)` parameters are the two phrases of the conjunction. For example, in the sentence: "I yield at the stop sign and I continue through the intersection," the two phrases are "I yield at the stop sign" and "I continue through the intersection." In the `make_triples_from_phrase` function, if a complex conjunction, e.g., AND or OR is found in the input phrase, then the phrases are split on the conjunction keyword. In the NOT case, the negation keyword is removed from the phrase. At this point, each phrase is converted into a set of triples.

### *6.3. Triple-Store Rule Creation*

Each of the `(?x)` is a list or singleton of RDF-style triples. The triples are made by using a part of speech (POS) tagger from the NTLK[7] library, especially `nltk.tokenize.word_tokenize`.

#### *6.3.1. Generating Triples*

Triples are generated by using a part-of-speech (POS) tagger and extracting the subject, verb, and object. The nltk POS tagger returns the list of tokens as a list of tuples of tokens, where each token is associated with a POS tag indicating whether the word is a noun, verb, adverb, adjective, etc.[8]. In creating triples, we proceed in the following steps to make a triple consisting of a `(subject, verb, object)`:

1. Set the verb by finding the first POS tag that starts with a "V". The verb is set by setting the verb. There are list of "special verbs," e.g., to be and to have which are represented with `IsA` and `HasA` which are semantic relations used in the semantic web, knowledge base, and symbolic AI community [23].
2. Set the subject by finding the first POS tag that starts with a "N" before the verb. If no subject is found, then the keyword `self` is used. This assigns the subject of the sentence to the ego vehicle.
3. Set the object by finding the first POS tag that starts with "N" after the verb.

[7] nltk library: https://www.nltk.org/
[8] The full list of POS tags are available here: https://www.nltk.org/book/ch05.html

### *6.4. First Iteration limitations*

In our preliminary investigation, our main goal was to demonstrate a proof of concept for extracting driving rules, identify interesting common-sense rules, and share them with other researchers.

Many of the rules that needed refinement were initially parsed incorrectly, mostly by the parser chopping words into letters separated by spaces. We focused on extracting driving rules from PDF manuals, as PDF files were available from all our target states.

After obtaining text, the next room for improvement is to guarantee good levels of soundness and completeness for the rules generated. Ideally, we want that all extracted rules are relevant to safety and accurate. Our tokenization method looks for if-then rules guarantees that we will get a conditional statement, but the expressiveness of this rule might not capture more complex situations.

### *6.5. Improvements*

Considering the initial limitations of the rule extractor, we identified several patterns that significantly enhanced the functionality of our prototype. The original extraction of 278 rules from 15 manuals represents only a fraction of the potential information available in these documents. We addressed and rectified parsing issues, thereby improving the quality and scope of the rule set.

The first enhancement involved expanding the script's keyword repository to include terms like *crosswalk, white line,* and *stop line.* This selection was informed by our analysis of the existing rule set, which frequently referenced these terms in relation to road features and specific signs. By focusing on these keywords, we were able to more effectively isolate previously overlooked rules.

Additionally, we refined the extraction process by analyzing the verb usage within rule sentences. We developed a methodology to select the verb that most accurately represents the intended action, based on its position relative to the subject. This involved typically considering the verb preceding the subject as part of the subject clause, while the subsequent verb was selected as the action descriptor. Moreover, we enhanced the parsing accuracy by addressing issues with non-ASCII characters and line break-induced word fragmentation.

These improvements led to the extraction of over 180 additional rules from the same manuals, enlarging our dataset by approximately 21% overall. However, challenges persist, as some of the new rules were derived from sample tests found in various driving manuals, such as the Illinois Driving Manual. Table 2 illustrates the comparison between the original and enhanced rule extraction. With the revised methodology, our dataset now encompasses 709 rules, indicating a substantial increase in extraction efficiency compared to the initial iteration.

## 7. Validation

We validate the rules manually. Overall, in our first iteration, we obtained 67 rules that did not require manual refinement at all while in our second we obtained 170. The following example shows one of the rules extracted by our method, informing drivers that if they go through an intersection but cannot advance (i.e., if they will block the intersection), they may get a ticket:

```
IF (self, block, intersection), THEN (self, get, ticket)
```

Our automated tool can extract rules that contain logical operators such as `AND` `OR` and `NOT`. For example, the following rule tells the driver that if they see a train, they should wait until it passes and then cross the tracks:

```
IF (self, see, train), THEN AND((wait, until, passes), (self, cross, tracks))
```

Our validation process focuses on taking the extracted rules automatically and checking for their consistency with the corresponding natural language. Specifically, we ensure that our structure is consistent with the rules and that the correct words are positioned throughout our structure. Likewise, in the case of missing numbers or chopped words, we track the sentence back to its manual and add it to the rule to complete it comprehensively.

Table 2
Number of extracted rules by state, original prototype and enhancement.

| State/Territory | First Extractor | Enhanced Extractor |
|---|---|---|
| California | 66 | 90 |
| Texas | 31 | 41 |
| Florida | 24 | 34 |
| New York | 44 | 57 |
| Pennsylvania | 43 | 57 |
| Ohio | 18 | 19 |
| Illinois | 42 | 52 |
| New Jersey | 44 | 45 |
| Virginia | 27 | 39 |
| Oregon | 0 | 46 |
| Arizona | 39 | 50 |
| Massachusetts | 77 | 48 |
| North Carolina | 0 | 42 |
| Queensland | 44 | 60 |
| Tasmania | 2 | 14 |
| West Australia | 12 | 14 |
| Total | 525 | 708 |

### 7.1. *Refinement*

ABout 75% of our automatically extracted rules require modifications before they can be used. Although we managed to improve the number of extracted rules that did not need any refinement, there are still many cases in which manual intervention is necessary. For example, the following automatically-extracted rule is incomplete:

`IF (Dont, isA, ashing), THEN (Drivers, yield, pedestrians)`

This rule has two problems, first, the subject is incomplete (it should be the `Don't walk` sign), and second, our parser had problems with some words (when we looked at the extracted sentence, the original sentence had the word `flashing`, but our PDF parser translated it as `fl ashing`). Therefore, we manually refined the extracted rule to the following proposition.

`IF (Don't walk, isA, flashing), THEN (Drivers, yield, pedestrians)`

To illustrate another example of the types of rules that needed manual refinements, consider the following rule:

`IF (self, isA, intersection), THEN (continue, through, intersection)`

While the above rule appears to be semantically correct, when we looked at the original sentence from which this rule was created, we discovered it was missing the `AND` logical operator in the antecedent. The sentence is "if you are in an intersection when you see an emergency vehicle, continue through the intersection." We, therefore, refined the rule as:

`IF AND((self, isA, intersection),(self, see, emergency vehicle)) THEN (continue, through, intersection)`

Overall we used 708 automatically-extracted rules as a starting point and refined them to create valid rules. We emphasize that we did not read the manuals, nor created the rules completely manually. We only looked at the extracted sentence from our parser (sentences satisfying if-then conditions), and the associated rule created from that sentence. If the semantics of both statements did not match, we refined the extracted rule to match the intended semantics of the original sentence.

### 7.2. *Relevancy*

Finally, while we focused on parsing the relevant parts of the driving manuals, we still obtained several rules that are not directly related to safe driving. We still keep these extracted rules because they may be used to expand the functionality of autonomous driving agents. For example, an autonomous agent might inform the human driver that it can place a call to an emergency line: “if a dangerous condition exists at a rail crossing call the number listed on the emergency sign”

```
IF (dangerous condition, at, rail), THEN (self, call, emergency number)
```

Autonomous agents can also remind human drivers of potential penalties:

```
IF (self, isA, hit-and-run), THEN (punishment, isA, severe)
```

Finally, some of these non-driving rules can be helpful for autonomous driving agents to inform their prior beliefs on how other objects near the car may behave. For example, we extracted rules regarding motorized bicycles, such as the following rule telling motorized bicyclists to use caution with others,

```
IF (self, drive, motorized bicycle), THEN (use caution, avoid, other bicyclists)
```

## 8. Results

In summary, our approach automatically identified 169 semantically-correct rules, with an extra 539 rules obtained by manual refinement. We summarize these results in Table 2. The results are compiled in a CSV file available in our GitHub repository[9].

Table 3
Total number of rules extracted automatically, refined.

| State/Territory | Automated | Refined | Total |
|---|---|---|---|
| California | 26 | 64 | 90 |
| Texas | 11 | 30 | 41 |
| Florida | 7 | 27 | 34 |
| New York | 16 | 41 | 57 |
| Pennsylvania | 12 | 45 | 57 |
| Ohio | 4 | 15 | 19 |
| Illinois | 24 | 28 | 52 |
| New Jersey | 14 | 31 | 45 |
| Virginia | 1 | 38 | 39 |
| Arizona | 5 | 45 | 50 |
| Massachusetts | 15 | 33 | 48 |
| Queensland | 16 | 44 | 60 |
| Tasmania | 2 | 12 | 14 |
| West Australia | 2 | 12 | 14 |
| Total | 169 | 539 | 708 |

### *8.1. Rule Coverage: Why Are Rules Hard to Extract in Some Manuals?*

As we can see in the table, we were able to extract a significant number of rules (90 in some states) while in some other regions, we got few rules–e.g., Tasmania with 14, West Australia also with 14, and Ohio with 19. In this section we study why we had few rules for these states.

We found that the challenges for extracting rules originated from three main problems:

1. Our parser had difficulties with page jumps and design choices regarding columns being inconsistent through the pages
2. Some manuals make heavy use of images and diagrams to express driving rules, and therefore, our parser cannot extract them.
3. The writing style of some manuals did not match the tokens we were looking for.

[9]https://github.com/RollingBeatle/rule-analyzer/tree/main/results

For example, in some manuals, PyPDF2 had issues extracting the text, as in the case of Tasmania. For example, the following rule,

`IF (travel, oo, ride), THEN (self, co, b)`

corresponds to the following extracted text warning the driver that a rider might come off their bicycle:

`if yo u travel t oo close to a ride r they co uld come off their b icyc le`

Our automated tool was unable to extract rules from this broken English. After seeing these problems with the parsing from PyPDF2, we attempted to use other tools for extracting text, but we got similar results.

With West Australia, we found that the format of the manual did not allow PyPDF2 to find text. While the manual does have some text that can be parsed, most of the driving rules are within figure boxes, and the text in the figure boxes was not parsed by PyPDF2, as the graphics interrupted the reading process.

Finally, 3 of the driving manuals have different styles that affect our parser. For example, the driving manual of Ohio has several statements written as cases, such as "Drive on the right half of the roadway except:" followed by a list of cases when the driver should avoid this behavior as:

- When overtaking and passing another vehicle proceeding in the same direction.
- When driving on a road divided into three or more marked lanes.
- When driving on a road designed and posted with signs for one-way traffic.
- When otherwise directed by a police officer or traffic control device
- When an obstruction makes it necessary for you to drive left-of-center.

This style is not appropriately read by our parser, thus preventing us from extracting these rules.

Similarly, when trying to parse the driving manual of Washington it does not follow the standard if-then form that our parser works on. Important information that would generate rules is posted in a similar listing format that is not easily read by our parser. As an example take this list of speeds allowed on different roads from the Washington Driver Guide:

"Obey speed limit signs. They are there for your safety. Speed limits, unless otherwise posted, are:"

- 20 mph in school zones.
- 25 mph on streets of cities and towns.
- 50 mph on county roads.
- 60 mph on state highways.
- Parts of interstate highways may be posted with higher maximum speeds.

### *8.2. Manuals with a Large Number of Driving Rules*

We now turn our attention to the state with the largest number of driving rules: California. This outlier result is interesting because we feed our tool more pages to analyze from the driving manuals of New Jersey and Queensland, but we end up with half of the rules compared to the number we obtained from California.

In analyzing why we got these results, we found that the writing style of California's driving manual was particularly well-suited for our rule extraction method. Most of the rules were expressed as if-then statements in sentences after section subtitles. Take as an example, the following text was successfully extracted: "if you cannot see farther than 100 feet, it is not safe to drive faster than 30 mph." The text follows our antecedent consequence form, and given the manual structure, it had no conflict with figures or other statements. Although California was the manual with the most rules, others, such as Queensland, New York, and Pennsylvania, had many extracted rules. These manuals, although some rely on diagrams, are complemented with significant text that helps our extractor get the important rules

In addition, Ohio and West Australia explain different behaviors using diagrams, lists and tables, which our parser cannot process. Likewise, the presence of the figures with descriptions can conflict with the text extraction process, making it more difficult for our parser to get the proper text.

### *8.3. Classification of Rules*

With our enhanced rule set, we managed to extract more diverse rules, this meant that we needed classification to have a much more precise categorization of the rules. This categorization allowed us to frame each rule in a specific context with the precision of the situation and actors involved that can be used in determining their importance when being applied to an AV driving context. A summary of our rules per category is displayed on Table 4.

Table 4

Table of Categories and their Corresponding Number of Rules

| **Category** | **Number of Rules** |
|---|---|
| Other Driving Indications | 197 |
| Safety | 184 |
| Signs and Road Markings | 151 |
| Interactions Between Vehicles | 120 |
| Emergency Driving Response | 69 |
| Accident Response | 64 |
| Pedestrian and Cyclist Safety | 49 |
| Law Enforcement Interactions | 46 |
| Vehicle Peripherals and Accessories | 44 |
| Weather | 41 |
| Mechanical Issues | 40 |
| DMV | 31 |
| Sanctions | 29 |
| Substances | 20 |
| Wildlife | 11 |
| Emergency Vehicles Interactions | 9 |

As seen in Table 4, our categorization shows a wide range of rules in vastly different subjects, most related to safety or generally acceptable behavior from the drivers on the road. This shows an interesting contrast with the number of rules extracted that involve emergency vehicles; most rules that could potentially involve an emergency vehicle center on how the driver should handle the situation directly and mainly report to emergency services, perhaps implying that the involved vehicle will get out of the road when possible and not interact with them. Rules in this category did not come from a unique manual; among them there was one rule in California and Pennsylvania describing situations when a driver should set aside or continue driving.

Another interesting case is the wildlife category, in which few rules exist. Most of the rules in this category come from the state of Florida, with several preventative actions a user could take to avoid or prevent a crash with an animal; this could be a reflection of Florida's wide variety of species.

We now explain each category and give an illustrative example in natural language of which rules are contained inside.

**Safety:** Refers to behaviors made by the drivers or other actors on the road that contribute to maintaining safety all around.

– *Keep a safe distance around your vehicle, this gives you time to react if another driver makes a mistake.*

**Accident Response:** What the driver or passengers inside a vehicle should do after being or after witnessing an accident.

– *If an aggressive driver is involved in a crash stop a safe distance from the crash scene*

**Emergency Driving Responses:** Actions that the driver can take in an emergency situation that might prevent or mitigate a potential accident scenario.

– *If hitting something is unavoidable try to make it a glancing blow*

**Pedestrian/cyclist safety:** How vehicles interact with pedestrians and bicyclists.

- *If a vehicle has stopped to give way at a pedestrian or children's crossing you must not overtake the stopped vehicle*

**Vehicle Peripherals and Accessories:** Actions or advice concerning vehicle inside workings and accessories.

- *If using your high beams you must dim your lights when an oncoming vehicle is within 500 feet*

**Interactions between Vehicles:** How a vehicle interacts with other vehicles on the road.

- *Never move into the same lane space as a motorcycle even if the lane is wide and the motorcycle is to one side*

**Law Enforcement Interactions:** Rules instructing the driver how to behave in situations where law enforcement is involved.

- *If a traffic officer signals you to stop at a green light for example you must stop*

**Mechanical Issues:** How to react to mechanical failures while on the road.

- *Headlight failure if your headlights suddenly go out try your emergency lights parking lights and directional signals*

**Emergency Vehicles Interactions:** Refers to how a driver should react to emergency response vehicles on the road.

- *If drivers are already in the roundabout exit first then pull over to let the emergency vehicle pass*

**Sanctions:** How the state can sanction a driver for breaking the law.

- *If a death or personal injury occurs as a result of consumption the parent or legal guardian may face criminal penalties*

**Substances:** The influence of alcohol and other drugs on the driver.

- *Do not drive if you are very weary are on medication or have been drinking beverages that contain alcohol.*

**DMV:** Driving license and registration procedures.

- *A current inspection is required within the last months prior to the sale if a registration plate will be requested.*

**Other Driving Indications:** Other advice when cruising the road.

- *Use a 3-point turn only if the road is too narrow for a u-turn and you can't go around the block*

**Wildlife:** Interactions and responses to animals on the road.

- *If you have time to avoid hitting an animal reduce your speed tap your brakes and sound your horn*

**Weather:** How to react to different weather events.

- *you may only drive with fog lights on if you are in fog or hazardous weather conditions causing reduced visibility*

In response to expanding our ruleset, encompassing fewer than 1,000 rules, we opted for manual classification and labeling. We categorized the rules into 16 distinct groups. To enhance the accuracy of our categorization, we thoroughly examined each manual, pinpointing the specific chapter from which each rule originated. While we meticulously recorded the chapter names, we deliberately avoided noting each subsection to prevent an over-specialization that could result in excessive categories with sparse rule allocation. Likewise, we noticed that some subjects intercepted with one another, which led us to the decision that rules should be able to belong to several categories.

This systematic approach enabled us to discern trends and commonalities in the source of the rules. A notable observation was the frequent presence of chapters titled with variations of "Safe Driving" or terms akin to "Defensive Driving." Based on this, we established a distinct "Safety" category, which encompasses all rules pertinent to this

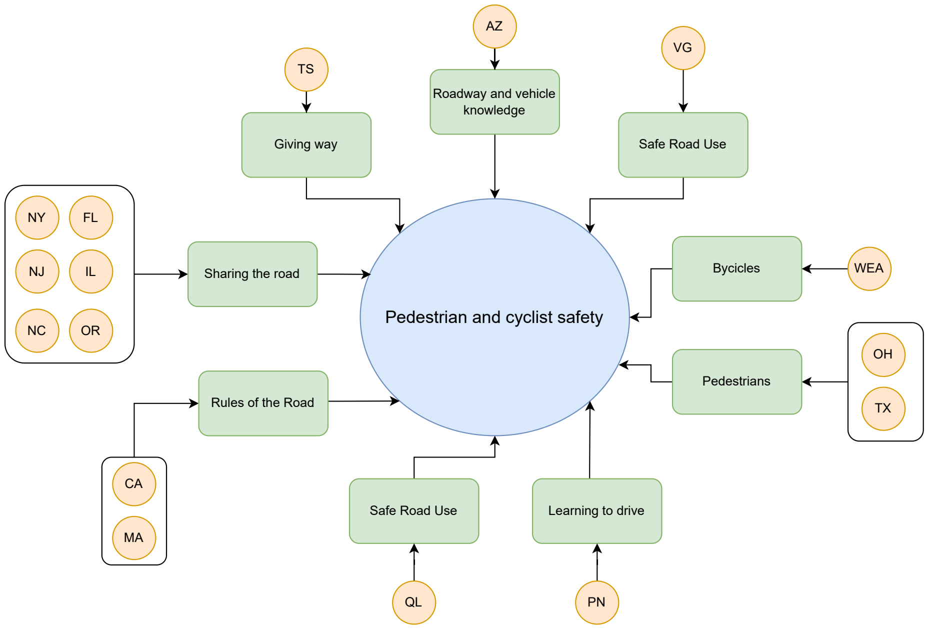


Fig. 2. Pedestrian and cyclist safety category. In this case, the rules had similar themes in the chapters index which lead us to a direct category based on these similarities.

theme. A more graphic visualization behind our approach and the categorization outcome for "Pedestrian and Cyclist Interactions" is exemplified in Figure 2.

Furthermore, our analysis identified subsets of rules that, despite thematic similarities, were distributed across various chapters without a discernible, consistent pattern. Acting as outliers, these rules demonstrated no clear association with their respective chapter titles. We introduced new categories for rules that lacked specific chapter correlations to address this. An example is the “Vehicle Peripherals and Accessories" category, which includes rules dispersed throughout different chapters. A visual representation of the process of forming these novel categories is depicted in Figure 3.

This categorization can give a clear insight into the types of limitations AVs should abide by while also helping to get a picture of how to determine the different factors that make a good driver. With the main three categories referencing indications, safety, and road signs, we are able to detect the main focus of the manuals and, in the future, build a scale in prioritization of the behaviors an AV may take on the road.

### *8.4. Are Driving Manual Rules Easy to Incorporate into an AV?*

To ascertain the complexity of integrating each rule into an AV system, we established a set of criteria centered primarily on the directness of the instruction in natural language and the extent to which the rule depends on variables outside the AV’s control. These variables encompass scenarios involving the driver’s administrative responsibilities and specific interactions valid only when law enforcement communicates with a human actor. Recognizing the need for further categorization, future iterations of this research will separately address rules that entirely necessitate human actors. However, in the current phase, classifying these as the most challenging to implement enables retaining valuable data within our relatively small dataset.

We have defined three difficulty levels for rule implementation: Easy, Moderate, and Hard. "Easy" rules are characterized by their straightforward directives, often unambiguous and occasionally accompanied by specific numerical criteria. "Moderate" rules encompass instructions that involve qualitative adjectives, such as "careful" or "cautious", highlighting the importance of the action but lacking explicit quantification and potentially harboring

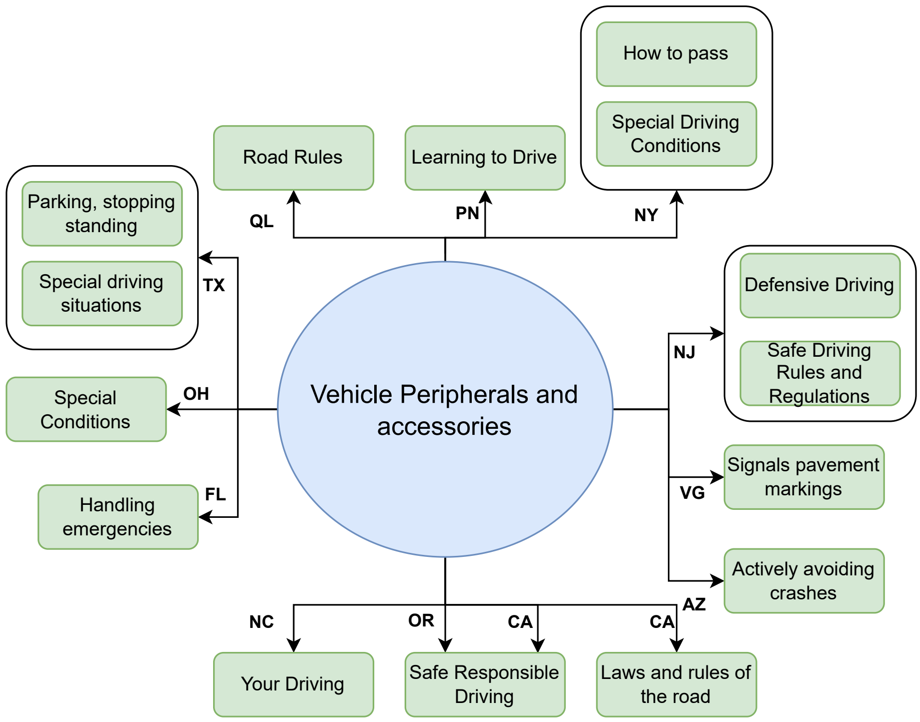


Fig. 3. Vehicle Peripherals and accessories category. In this case, we defined the category based on the common themes detected in the rules.

some ambiguity in either the conditional or consequent part of the statement. "Hard" rules pertain to those involving dynamic human interactions, emotional considerations of the actors, or ambiguity in both the conditions and consequences of the rule. A comprehensive explanation of these categorization criteria is presented in Table 5.

| Easy | Moderate | Hard |
|---|---|---|
| Concrete action (eg: break, stop, turn) | Presence of an adjective directed to the action | No clear action or condition in the rule. |
| Presence of measurable values regarding the action. | Action is measurable but there are no clear values. | Action is not measurable |
| There is a max of 2 conditions and consequences in the rule | Between 2 or 3 conditions or consequences. | More than 3 if condition is measurable |
| Involves no human interaction | Involves a human feeling or suggestion | Requires an interaction between human actors |

Table 5

Difficulty level criteria

To illustrate, we provide examples for each difficulty level:

- **Easy:** "If no limit is posted, drive no more than 55 mph."
- **Moderate:** "Maintain a safe distance from the light rail vehicle if it shares the road with other traffic."
- **Hard:** "If one danger is greater than the other, give more space to the most dangerous situation."

Considering this classification, we categorized all 709 rules in their respective difficulty level. This gave us 416 rules in the easy category, 163 in the moderate, and 95 in the difficult category. Having the majority of the rules at the easy level indicates that there is information that can be put into practice with relative ease in different AV systems. The more complex rules with more ambiguous or abstract reasonings would need to be further analyzed to determine how they could be used to help improve training systems by grounding the concepts into thresholds that could be used to influence planning decisions.

As we can see, most of the rules according to our classification are easy, which means the instructions a vehicle should follow are clear. This fact means that there is valuable information in the manuals that can be easily transformed into codifiable statements that vehicles could follow or restrict their behavior based on them. Moderate rules may lack a clear measurement or involve abstract concepts, but given the context, they may be able to be complemented with the easier ones. Finally, the hard rules, given the level of abstraction, require grounding and additional work that allows them to be applicable in day-to-day situations.

### *8.5. Region-Specific Rules*

The driving manuals can also give us insights into the unique challenges that drivers from different regions might experience. For example, the following rule:

`IF (self, encounter, duststorm), THEN (self, Reduce, speed)`

Represents the rule extracted from the following text “If you encounter a severe dust storm, reduce your speed immediately.” This rule was found in Arizona, but we didn’t find similar warnings in driving manuals from other states.

Another obvious regional difference is the side of the road you drive on and the associated rules. For example, drivers in Australia must drive on the left-hand side of incoming traffic, while drivers in the United States must drive on the right-hand side of the road.

For example, the following rule indicates how a vehicle should behave in West Australia if it is moving at a lower speed than other traffic.

`IF (self, isA, movingslow), THEN (moveleft, allow, traffic)`

This rule was extracted from the following text:

`if you are driving a slow moving vehicle pull well over to the left to allow following trafc to overtake`

In contrast, the following rule in California suggests the opposite behavior.

`IF (self, choose, slow), THEN NOT(self, do, left lane)`

Extracted from the following text:

`if you choose to drive slower than other traffic do not drive in the fast lane`

This diversity of the rules across the different territories indicates that not every situation is taken into consideration in each manual. This specificity could help researchers identify additional situations where unlikely events might collide with the usual interpretation of a norm or territories that have not considered possible cases for which an AV would need to be prepared.

## 9. Limitations and Future Work

Our main goal was to demonstrate a proof of concept for extracting driving rules, identify interesting commonsense rules, classify them into logical and well-structured categories, and share them with other researchers.

There is room for improvement in various areas. The first is to find or develop more robust text parsing tools. As explained in our section on refining rules, a large percentage of the rules that needed refinement were originally parsed incorrectly, mostly by the parser chopping words into letters separated by spaces. We focused on extracting driving rules from PDF manuals, as PDF files were available from all our target states. However, some states are releasing manuals in HTML format, and this might make parsing the text a bit easier. We will look into this option in future work.

After obtaining accurate text, the next room for improvement is to guarantee good levels of soundness and completeness for the rules generated. Our tokenization, looking for if-then rules, guarantees that we will get a conditional statement, but the expressiveness of this rule might not capture more complex situations. Although improved from our first iteration, issues regarding jumps and lists persist, and handling those presents a significant challenge.

In future work, we would like to expand our rules to include those that use temporal logic, and other types of logic: first-order predicate calculus, event calculus, etc. To use those logics, we can extend our rule keyword set to include other options, e.g., `whenever`, `as long as`, etc.

Finally, our next step will be to codify these rules into a specification that can be tested and helpful in training AVs in simulation environments. We want to identify what are the challenges for autonomous vehicles in implementing the various common-sense rules available in driving manuals and also to see how we can improve learning and if they can be monitored for compliance.

## 10. Conclusions

Our goal here was to extract a set of rules that show the variety of rules, conditions, and recommendations that future autonomous vehicles will have to deal with, even if partially so. We extracted overall 709 driving rules that we make publicly available to other researchers. These rules can guide the future development of AV systems by giving modules a more comprehensive picture of expected behavior. Having this rule set applied to learning systems in the AVs' planning and perception modules can significantly impact how society interacts with AVs on the road and make them more acceptable as they show great promise to improve road efficiency and safety.

## References


[1] A. Rizaldi, F. Immler and M. Althoff, A Formally Verified Checker of the Safe Distance Traffic Rules for Autonomous Vehicles, in: *NASA Formal Methods*, S. Rayadurgam and O. Tkachuk, eds, Springer International Publishing, Cham, 2016, pp. 175–190. ISBN 978-3-319-40648-0.

[2] A. Collin, A. Bilka, S. Pendleton and R.D. Tebbens, Safety of the Intended Driving Behavior Using Rulebooks, in: *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2020. doi:10.1109/iv47402.2020.9304588. http://dx.doi.org/10.1109/IV47402.2020.9304588.

[3] *https://www.cpuc.ca.gov/* (2023). https://www.cpuc.ca.gov/news-and-updates/all-news/cpuc-approves-permits-for-cruise-and-waymo-to-charge-fares-for-pass

[4] R. Mitchell, San Francisco's North Beach Streets clogged as long line of Cruise Robotaxis come to a standstill, Los Angeles Times, 2023. https://www.latimes.com/california/story/2023-08-12/cruise-robotaxis-come-to-a-standstill.

[5] K. Paul, The rebel group stopping self-driving cars in their tracks – one cone at a time, *The Guardian* (2023).

[6] M. Cassidy, S.F. viral video shows person attacking Cruise car with hammer, *San Francisco Chronicle* (2023).

[7] A.Y. Bin-Nun, P. Derler, N. Mehdipour and R.D. Tebbens, How should autonomous vehicles drive? policy, methodological, and social considerations for designing a driver, *Humanities and Social Sciences Communications* **9**(1) (2022). doi:10.1057/s41599-022-01286-2.

[8] J. Stilgoe and M. Mladenović, The politics of Autonomous Vehicles, *Humanities and Social Sciences Communications* **9**(1) (2022). doi:10.1057/s41599-022-01463-3.

[9] J. De Freitas, B.W. Smith, A. Censi, L. Di Lillo, S.E. Anthony and E. Frazzoli, From driverless dilemmas to more practical common-sense tests for automated vehicles, *Proceedings of the National Academy of Sciences of the United States of America* **118** (2021). doi:10.31234/osf.io/ypbve.

[10] B. Hancock, P. Varma, S. Wang, M. Bringmann, P. Liang and C. Ré, Training classifiers with natural language explanations, *arXiv preprint arXiv:1805.03818* (2018).

[11] M. Molineaux and D.W. Aha, Learning unknown event models, in: *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[12] L.H. Gilpin and L. Kagal, An Adaptable Self-Monitoring Framework for Complex Machines, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2019, Montreal, Quebec, May 13-17, 2019* (2019), 3.

[13] A. Censi, K. Slutsky, T. Wongpiromsarn, D. Yershov, S. Pendleton, J. Fu and E. Frazzoli, Liability, Ethics, and Culture-Aware Behavior Specification using Rulebooks, in: *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8536–8542. doi:10.1109/ICRA.2019.8794364.

[14] J. Chow, V. Richmond, M. Wang, U. Guajardo, D. Jackson, N. Arechiga, G. Litt, S. Kong and S. Campos, Certified Control: A New Safety Architecture for Autonomous Vehicles.

[15] D.J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A.L. Sangiovanni-Vincentelli and S.A. Seshia, Scenic: a language for scenario specification and scene generation, in: *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, 2019, pp. 63–78. doi:10.1145/3314221.3314633.

[16] S. Shalev-Shwartz, S. Shammah and A. Shashua, On a formal model of safe and scalable self-driving cars, *arXiv preprint arXiv:1708.06374* (2017).

[17] L. Morgenstern, C. Welty, H. Boley and G. Hallmark, RIF primer, *W3C Recommendation* **22** (2010), 190.

[18] L.H. Gilpin, Anomaly detection through explanations, PhD thesis, Massachusetts Institute of Technology, 2020.

[19] S. Ceri, G. Gottlob, L. Tanca et al., What you always wanted to know about Datalog(and never dared to ask), *IEEE transactions on knowledge and data engineering* **1**(1) (1989), 146–166.

[20] V. Chandola, A. Banerjee and V. Kumar, Anomaly detection: A survey, *ACM computing surveys (CSUR)* **41**(3) (2009), 15.
[21] J. Holtz, A. Guha and J. Biswas, Interactive robot transition repair with SMT, in: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, AAAI Press, 2018, pp. 4905–4911.
[22] R. Ramakrishnan, E. Kamar, B. Nushi, D. Dey, J. Shah and E. Horvitz, Overcoming Blind Spots in the Real World: Leveraging Complementary Abilities for Joint Execution (2019).
[23] R.J. Brachman, What IS-A is and isn't: An analysis of taxonomic links in semantic networks, *Computer* **16**(10) (1983), 30–36.