To Drive or Not to Drive: A Logical and Computational Analysis of European Transport Regulations*

Ana de Almeida Borges[a], Juan José Conejero Rodríguez[a], David Fernández-Duque[b,*], Mireia González Bedmar[a], Joost J. Joosten[a]

[a] *University of Barcelona, C. Montalegre 6, 08001 Barcelona, Spain*
[b] *Ghent University, St. Pietersnieuwstraat 33, 9000 Ghent, Belgium*

Abstract

This paper analyses a selection of articles from European transport regulations that contain algorithmic information, but may be problematic to implement. We focus on issues regarding the interpretation of tachograph data and requirements on weekly rest periods. We first show that the interpretation of data prescribed by these regulations is highly sensitive to minor variations in input, such that near-identical driving patterns may be regarded both as lawful and as unlawful. We then show that the content of the regulation may be represented in mondadic second order logic, but argue that a more computationally tame fragment would be preferrable for applications. As a case study we consider its representation in linear temporal logic, but show that a representation of the legislation requires formulas of unfeasible complexity, if at all possible.

Keywords: linear temporal logic, monadic second order logic, formalized law, transport regulations, automated law enforcement, tachograph

*This paper is part of the project RTC-2017-6740-7 funded by the "Ministry of Science, Innovation and Universities", the "State Agency for Research" and the "European Regional Development Fund" (ERDF). David Fernández-Duque's research was partially supported by COST Action 17124 DigForAsp, supported by COST (European Cooperation in Science and Technology), www.cost.eu.

*Corresponding author

Email addresses: anadealmeidagabriel@ub.edu (Ana de Almeida Borges), juan.conejero@ub.edu (Juan José Conejero Rodríguez), davidstoteles@gmail.com (David Fernández-Duque), m.gonzalezbedmar@ub.edu (Mireia González Bedmar), jjoosten@ub.edu (Joost J. Joosten)

1. Introduction

The authors of the paper are involved in research projects in collaboration with industry, lawyers and legislators, where the main goal is to develop verified legal software. The industrial and social need is evident: various legal decisions are made on the basis of algorithmic processing of data, in consequence of which individuals can be fined or even sent to jail. Software contains errors, but in the legal context such errors should not be acceptable.

In particular, the above-mentioned projects have as first and main objective to eradicate errors from software that interprets data from tachographs. A tachograph is to a truck what a black-box is to an aeroplane: it registers all kinds of activities from the truck and driver, such as speed, movement and others. In practice, a police officer may pull a truck over for an inspection where the tachograph data is read and interpreted by some software. Depending on the verdict of the program, the driver may be instantly fined or sometimes even imprisoned. It is known that many erroneous automated verdicts are issued. This is highly undesirable both from an industrial and from a civil rights perspective. It is here that logic tries to come to the rescue.

The aim of the project is to recast the transport legislation into an unambiguous, mathematically formulated language, such that proof-checkers may show that the developed code indeed satisfies the legislation. This paradigm allows us to honestly speak of error-free software.[1]

The multi-disciplinary nature of the project poses many challenges. For one, legislation is often intended to leave room for various interpretations and applications of the law. In contrast, mathematical definitions and algorithms are deterministic in nature and disallow ambiguity. The main mitigation of this challenge seems to be the accepted tendency to require unambiguous laws if they prescribe an algorithm. We will show how easily overlooked subtleties in the law could produce drastic changes in the output, possibly making a legal sequence of activities appear to be illegal.

Section 2 gives a basic introduction to the aspects of the European transport regulations that we work with. Sections 3-6 are devoted to treating the problems mentioned above and are largely based on [1].

Once this data has been appropriately processed, Sections 7-10 are largely

[1]It has its subtleties, though. Software will be as good as the specification, which may be erroneous, and we must trust the small kernel of the proof-checker, apart from the hardware and middle-ware involved.

based on [2] and devoted to the next challenge, which lies in choosing the right ontology and logico-mathematical framework in which to recast the interpreted and disambiguated laws. We will argue that the content of the European regulation can be modelled in monadic second order logic (MSO) and its fragments.[2]

To illustrate this claim we identify some passages that may be problematic from a logical perspective, most notably because they involve second order quantification. All laws we consider will fall in the Σ_1^1 fragment of monadic second-order logic, and model-checking formulas in this fragment can be reduced to satisfiability of first-order formulas. Nevertheless, this satisfiability problem is PSPACE-complete, so we argue that such laws should instead be representable in a comptuationally tame fragment of MSO. We justify the use of linear temporal logic (LTL) as a case-study to explore the representability in such a fragment, but show that the law cannot be represented by an LTL formula of 'reasonable' complexity.

2. European transport regulations

Regulation (EU) 2016/799 [5] lays down the requirements for the use of tachographs – digital devices that record the activities of road transport drivers. This data is used to determine whether drivers have complied with Regulation (EC) 561/2006 [6], which is written assuming a minute-by-minute temporal resolution. Items (51) and (52) of Regulation (EU) 2016/799 regulate how the second-by-second data recorded by tachographs is to be translated into a minute-by-minute format. They read as follows:

(51) Given a calendar minute, if DRIVING is registered as the activity of both the immediately preceding and the immediately succeeding minute, the whole minute shall be regarded as DRIVING.

(52) Given a calendar minute that is not regarded as DRIVING according to requirement 051, the whole minute shall be regarded to be of the same type of activity as the longest continuous activity within the minute (or the latest of the equally long activities).

[2] To be precise, we claim that MSO is suitable for formalizing the algorithmic content of the European transport regulation. A general formalization of the law with its obligations, prohibitions, and permissions, involves deontic reasoning which, it has been argued, cannot be adequately captured within temporal or first order reasoning [3, 4].

Once data has been formatted according to these items, it must be checked whether they comply with Regulation (EC) 561/2006. For instance, drivers must have a weekly rest period (similar to a weekend), as prescribed by the following articles.

§4(h) 'regular weekly rest period' means any period of rest of at least 45 hours.

§4(i) 'a week' means the period of time between 00.00 on a Monday and 24.00 on the following Sunday.

§8.6. In any two consecutive weeks, a driver shall take at least:

- two regular weekly rest periods, or
- one regular weekly rest period and one reduced weekly rest period of at least 24 hours. However, the reduction shall be compensated by an equivalent period of rest taken en bloc before the end of the third week following the week in question.

A weekly rest period shall start no later than at the end of six 24-hour periods from the end of the previous weekly rest period.

§8.7. Any rest taken as compensation for a reduced weekly rest period shall be attached to another rest period of at least nine hours.

§8.9. A weekly rest period that falls in two weeks may be counted in either week, but not in both.

We have selected these passages because they have been particularly problematic during our attempts at implementing the law. There are of course other conditions that drivers must comply with (e.g. daily rest periods), but we will ignore them for the sake of exposition.

3. Potential issues with the regulations

In this section we informally discuss some potential pitfalls that may occur when implementing the above-mentioned regulation. Later in the text we provide a more rigorous analysis of these issues. We begin with the items of Regulation (EU) 2016/799.

3.1. Order of application

Requirement (51) is problematic in that it refers to an activity being registered in function of the preceding and succeeding minute, with no previous reference to when an activity is considered registered. To be able to apply Requirement (51), one needs to have previously registered some activity, and only in (52) is a criterion for doing so established.

However, since Requirement (52) references Requirement (51), it seems clear that in some sense (51) precedes (52). One possible interpretation is that the requirements are to be applied as many times as needed to satisfy both statements: we would have the chain of applications (51)-(52)-(51)-(52)-$\cdots$. In Section 5 we will prove that this sequence of applications is equivalent to the sequence (52)-(51).

Another interpretation, which in a legal context might seem more customary, is that the requirements are to be applied in the order of appearance, (51)-(52). In this case, after the application of (52) it is possible to reach a configuration that violates (51), as we shall see in Section 5.

In any case, the ambiguity of the regulation is undeniable. If we trace these requirements back in time, we find that there was another sentence immediately preceding (51) which was removed by an amendment in Commission Regulation (EU) 1266/2009 [7], changing the global meaning of the excerpt. Despite these issues, in this paper, we will restrict our attention to the legal text in its current form.

Since the internal functioning of commercial tachographs is subject to proprietary software restrictions, we cannot freely check the implementation of the regulation that they have chosen. However, Guretruck S.L. has conducted experimental tests and deduced from them that commercial tachographs apply Requirement (52) followed by Requirement (51), which is a dubious interpretation of the law. They also disregard leap seconds, which are part of the UTC time standard prescribed by Regulation (EU) 2016/799.

Guretruck S.L. has conducted experimental tests with real-world driver data as well, finding that the minute labellings computed with the proper UTC calendar vary up to 8% of driving time with respect to the minute labellings computed disregarding leap seconds, even using a small sample of driver files.

3.2. Placement of weekly rest periods

Let us consider a case implied by Article §8 of Regulation (EC) 561/2006 (see Figure 1). Each letter-divided segment denotes a week and the smaller

segments denote a day. Furthermore, each serpentine line denotes weekly rest periods of 68 hours except the last one, which only lasts 45 hours.

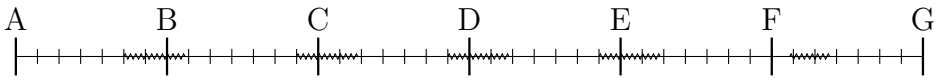

Figure 1: Six weeks of activity of a hypothetical driver.

Figure 1 represents the activities of a driver who starts resting Saturday at 00:00h and retakes her activity on Monday at 20:00h. Then, until the fourth week, the driver periodically starts her weekly rest on Sunday at 00:00h and retakes her activity on Tuesday at 20:00h. During the sixth week she rests 45 hours, from Monday at 20:00h to Wednesday at 17:00h.

Since all except the last of these weekly rest periods fall between two weeks, it is reasonable to want to find a procedure that will determine whether there exists a way of counting each of them within one week or the other as per §8.9 such that the situation becomes legal.

In our simple example, the segment FG has a fixed rest period of 45 hours. In the remaining weeks we have to choose where to assign the resting periods, but it is evident that we cannot arrange them in a way that makes the whole interval AG legal. One might argue that this situation is a bit controversial, given that all other articles exposed above except §8.9 are complied beyond their minimum requirements.

Here, the Regulation does not pose a logical problem, nor is it inconsistently worded. But logic is not entirely unrelated to this issue. The complexity that results from §8.9 generates a potential combinatorics problem. As an example, we could encounter situations which follow the structure from Figure 1 with many more occurrences of the *in between* segments. Verifying the legality of the situation could, in principle, require checking a large number of possible assignments of rest intervals to weeks. This non-locality feature has been discussed and formalised in Coq [8].

3.3. Timing of compensations

The second potential source of problems comes from the compensation mechanism of §8.6. To illustrate it, we construct weeks A, B, and $C_i, 1 \leq i \leq n$ such that the sequences $A \mid C_1 \mid \ldots \mid C_n$ and $C_1 \mid \ldots \mid C_n \mid B$ are both legal, but the full sequence $A \mid C_1 \mid \ldots \mid C_n \mid B$ is *not*. The question then arises: *where is the illegality?* It is in the combination between A and

B, where A and B can be arbitrarily far apart from each other. Clearly this is not a good feature for a law.

Throughout this subsection, line segments represent weeks, and the numbers attached to them represent the number of hours rested during each week. In Figure 2, the first and last segments represent the weeks A and B we mentioned before.

Figure 2: Illegal interval of six consecutive weeks performed by a hypothetical driver.

As shown in Figure 3, if we do not consider the last week, the remaining interval is rendered legal by the law, for we can assume that the hours to be compensated will be incorporated in the week we omitted.

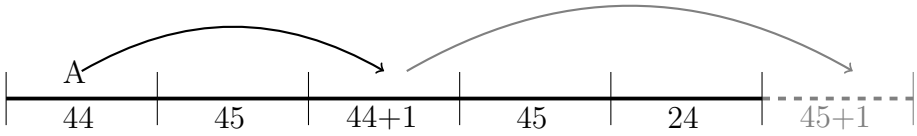

Figure 3: First five weeks of the example represented in Figure 2, together with a possible sixth week that would make the whole interval legal.

Similarly, if we remove week A from the example of Figure 2, the resulting interval (represented in Figure 4) is also legal, since we can assume that the compensation for the fourth week takes place in the weeks outside our interval.

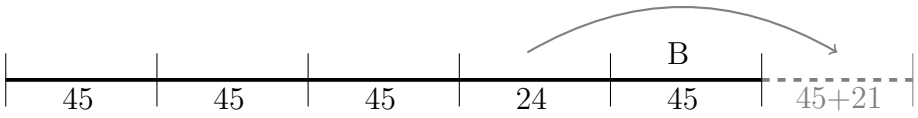

Figure 4: Last five weeks of the interval represented in Figure 2, together with a possible sixth week that would make the whole interval legal.

However, the interval of Figure 2 is illegal, as Figure 5 illustrates. This is because after compensating the first week according to article §8.6, we still have to compensate one hour, but we cannot allocate it within any of the

three following weeks without having two consecutive reduced weekly rest periods.

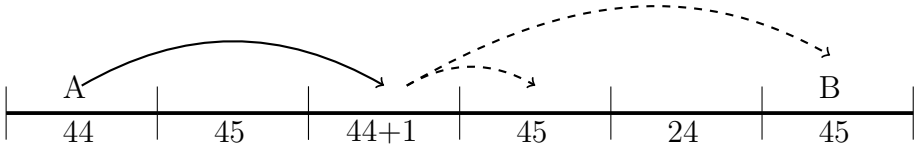

Figure 5: An attempt to assign compensations (dashed lines) that ultimately fails.

This example can be generalized to ensure that A and B are n weeks apart. The corresponding interval (illustrated in Figure 6) has a similar structure to the one we have treated. The first week has a 44 hour weekly rest period, and all the following weeks have 45 hour weekly rest periods except for the penultimate one, which has a 24 hour weekly rest period.

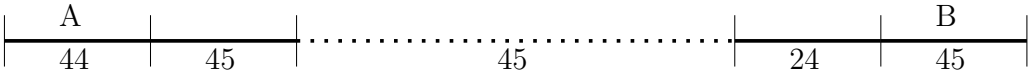

Figure 6: General example of an illegal interval that is legal when A or B is erased.

In this situation if we omit one of the weeks A or B the remaining interval will be legal, but the interval as it stands is illegal.

4. Second and minute labellings

In this section we will formalize the concepts and terminology we use for modelling labellings obtained from tachographs. We are given a list of consecutive seconds with their activity assigned by the tachograph. To represent the seconds, we will use the integer numbers $\mathbb{Z}$. The space of activities will be $\mathbb{A} := \{\text{DRIVING}, \text{REST}, \text{AVAILABILITY}, \text{WORK}, \text{UNKNOWN}\}$. A labelling is any function $f : D \to \mathbb{A}$, where $D \subseteq \mathbb{Z}$ is an interval. Although D is finite in practice, in this section we will assume for convenience that $D = \mathbb{Z}$; this is not an issue since we may extend f by letting $f(x) = \text{UNKNOWN}$ for $x \notin D$. We will use the following convention: when $\mathbb{Z}$ is being interpreted as seconds, we will say that $\mathcal{S} : \mathbb{Z} \to \mathbb{A}$ is a *second labelling,* and when $\mathbb{Z}$ is interpreted as minutes, we will say that $\mathcal{M} : \mathbb{Z} \to \mathbb{A}$ is a *minute labelling.*

When dealing with time shifts, we also use $\mathbb{Z}$ to represent the list of minutes of the calendar and we encode a shift as an integer $d \in \mathbb{Z}$. The

set $\mathbb{Z}_{60}$ is the set of integers from 0 to 59, both included. Given a shift d, we convert seconds to pairs consisting of a minute and a second of minute through a function $c^d : \mathbb{Z} \to \mathbb{Z} \times \mathbb{Z}_{60}$ defined for each second $s \in \mathbb{Z}$ as:

$$c^d(s) := \left(\left\lfloor \frac{s-d}{60} \right\rfloor, (s-d)\%60 \right),$$

where % denotes the Euclidean division remainder. We use c_0^d for the first component of c^d and c_1^d for the second.

Summarising, the calendar is such that minute 0 starts exactly at second d, minute 1 starts at second $d + 60$, and so on.

Example 4.1. Given a shift $d = 20$, we have that $c^{20}(30) = (0, 10)$, meaning that second 30 is the second 10 of the minute 0.

Requirements (51) and (52) give instructions for converting a second labelling to a minute labelling, corresponding to the following transformations.

Definition 4.2. Given a minute labelling $\mathcal{M} : \mathbb{Z} \to \mathbb{A}$, we define the labelling after applying Requirement (51) as $\mathrm{R51}(\mathcal{M}) : \mathbb{Z} \to \mathbb{A}$ defined, for $i \in \mathbb{Z}$, by:

$$\mathrm{R51}(\mathcal{M})(i) := \begin{cases} \textsc{driving} & \text{if } \mathcal{M}(i-1) = \mathcal{M}(i+1) = \textsc{driving}, \\ \mathcal{M}(i) & \text{otherwise.} \end{cases}$$

Definition 4.3. Given a shift d, a second labelling $\mathcal{S} : \mathbb{Z} \to \mathbb{A}$ and a minute labelling $\mathcal{M} : \mathbb{Z} \to \mathbb{A}$, we define the labelling after applying Requirement (52) as the minute labelling $\mathrm{R52}(d, S, M) : \mathbb{Z} \to \mathbb{A}$ defined, for $i \in \mathbb{Z}$, by:

$$\mathrm{R52}(d, \mathcal{S}, \mathcal{M})(i) := \begin{cases} A(d, \mathcal{S}, i) & \text{if } \mathcal{M}(i) = \textsc{unknown}, \\ \mathcal{M}(i) & \text{otherwise,} \end{cases}$$

where $A(d, \mathcal{S}, i)$ is the activity $a \in \mathbb{A}$ such that $\mathcal{S}^{-1}(a) \cap [d+60i, d+60i+60)$ contains the rightmost interval of maximal length.

In words, $A(d, \mathcal{S}, i)$ is the activity a that either contains the longest consecutive interval among all activities, or if there is a tie, then a has an interval of maximal length that is to the right of that for any other activity.

5. Order of application

Next we show that the application of Requirements (51) followed by (52), with no further applications, might yield a configuration where (51) is violated as requirement, while the application of (52) followed by (51) is stable: no further applications of either requirement produce any changes.

Definition 5.1. The unknown labelling is the minute labelling $\mathcal{U} : \mathbb{Z} \to \mathbb{A}$ such that, for all $i \in \mathbb{Z}$, $\mathcal{U}(i) = \textsc{unknown}$.

The algorithm starts as follows: we are given a second labelling $\mathcal{S}$ from the tachograph, a shift d given by the calendar that we are using, and the unknown labelling $\mathcal{U}$ as a starting minute labelling in which nothing is registered yet. Now, we can apply R51 and R52 above. It is an easy observation that $\mathrm{R51}(\mathcal{U}) = \mathcal{U}$, since there is no DRIVING registered yet.

Theorem 5.2. Let $\mathcal{U}$ be the unknown labelling. There are second labellings $\mathcal{S}$ and shifts d such that $\mathrm{R52}(d, \mathcal{S}, \mathrm{R51}(\mathcal{U})) \neq \mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathrm{R51}(\mathcal{U}))\big)$, i.e., such that the application of Requirement (51) after the application of Requirements (51) and (52) still changes the minute labelling.

Proof. Consider, for instance, $d = 0$ and $\mathcal{S}(s) := \textsc{driving}$ if $c_0^0(s)$ is even, $\mathcal{S}(s) := \textsc{rest}$ otherwise. In this case, for any $i \in \mathbb{Z}$,

$$\mathrm{R52}(0, \mathcal{S}, \mathrm{R51}(\mathcal{U}))(i) = \mathrm{R52}(0, \mathcal{S}, \mathcal{U})(i) = \begin{cases} \textsc{driving} & \text{if } i \text{ is even,} \\ \textsc{rest} & \text{otherwise,} \end{cases}$$

and $\mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathrm{R51}(\mathcal{U}))\big)(i) = \textsc{driving}$. □

Now, we are going to prove that, once we have applied Requirement (52) followed by (51), the configuration reached is stable: no further applications will produce any changes.

Theorem 5.3. Let $\mathcal{U}$ be the unknown minute labelling, let $\mathcal{S}$ be a second labelling and let d be a shift. Then the following both hold:

(i) $\mathrm{R51}\Big(\mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathcal{U})\big)\Big) = \mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathcal{U})\big)$,

(ii) $\mathrm{R52}\Big(d, \mathcal{S}, \mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathcal{U})\big)\Big) = \mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathcal{U})\big)$.

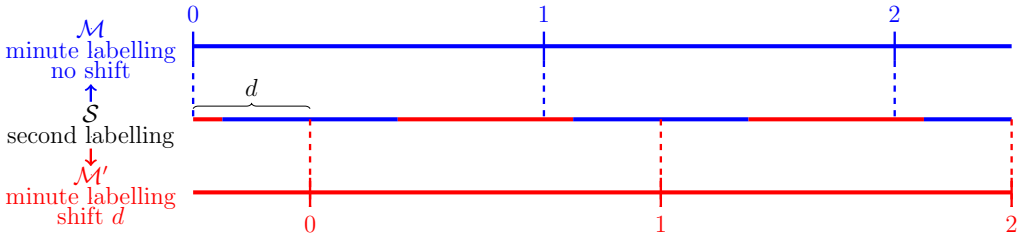

Figure 7: A second labelling $\mathcal{S}$ which gives rise to two different minute labellings depending on whether or not a shift d is applied. The color in the middle line indicates that the second labelling matches with the minute labelling of the respective minute of the same color.

Proof. Let $\mathcal{M}_1 = \mathrm{R52}(d, \mathcal{S}, \mathcal{U})$ and let $\mathcal{M}_2 = \mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathcal{U})\big)$.

For (i), if $i \in \mathbb{Z}$, by Definition 4.2 we have $\mathrm{R51}(\mathcal{M}_2)(i) :=$ DRIVING if $\mathcal{M}_2(i-1) = \mathcal{M}_2(i+1) =$ DRIVING, $\mathrm{R51}(\mathcal{M}_2)(i) := \mathcal{M}_2(i)$ otherwise. Hence, the non-trivial case is when $\mathcal{M}_2(i-1) = \mathcal{M}_2(i+1) =$ DRIVING. If $\mathcal{M}_2(i) =$ DRIVING, then we are done. Assume towards a contradiction that $\mathcal{M}_2(i) \neq$ DRIVING. Since $\mathcal{M}_2 = \mathrm{R51}(\mathcal{M}_1)$, we have that $\mathcal{M}_1(i) \neq$ DRIVING (because applying R51 never erases DRIVING). From this we see that $\mathcal{M}_2(i-1) := \mathcal{M}_1(i-1)$, hence $\mathcal{M}_1(i-1) =$ DRIVING, and similarly $\mathcal{M}_1(i+1) =$ DRIVING. But then, $\mathcal{M}_2(i) =$ DRIVING, which is a contradiction.

For (ii), just notice that R52 is the identity over the known part, and the unknown does not change because we keep the same $\mathcal{S}$. □

Note that since $\mathrm{R51}(\mathcal{U}) = \mathcal{U}$, starting with an application of R51 followed by R52 followed by R51 is the same as starting with R52, and thus by Theorem 5.3 we may conclude that all orderings of successive applications of R51 and R52 to $\mathcal{U}$ converge to the same labelling.

6. Time shifts

In this section we prove that, if we follow Regulation (EU) 2016/799, we obtain that the same sample from the tachograph can give two completely different minute labellings if interpreted in calendars with different shifts.

Theorem 6.1. Let d be such that $1 \leq d \leq 59$. Let $\mathcal{U}$ be the unknown minute labelling and let $\mathcal{M}$ and $\mathcal{M}'$ be two minute labellings. Then there exists a second labelling $\mathcal{S}$ such that $\mathrm{R52}(0, \mathcal{S}, \mathcal{U}) = \mathcal{M}$ and $\mathrm{R52}(d, \mathcal{S}, \mathcal{U}) = \mathcal{M}'$.

Proof. We can assume without loss of generality that $d \leq 30$, otherwise we switch $\mathcal{M}$ and $\mathcal{M}'$. For every $s \in \mathbb{Z}$, we define $\mathcal{S}(s)$ as follows:

$$\mathcal{S}(s) := \begin{cases} \mathcal{M}'(c_0^0(s) - 1) & \text{if } c_1^0(s) = 0, \\ \mathcal{M}(c_0^0(s)) & \text{if } 0 < c_1^0(s) \leq 30, \\ \mathcal{M}'(c_0^0(s)) & \text{if } c_1^0(s) > 30. \end{cases}$$

Notice that $\mathrm{R52}(0, \mathcal{S}, \mathcal{U})(i) = A(0, \mathcal{S}, i)$, which is the activity of the rightmost maximal interval in $[60i, 60i + 60)$. By the definition of $\mathcal{S}$, in that interval there is a first second $60i$ of activity $\mathcal{M}'(i-1)$, then 30 seconds of activity $\mathcal{M}(i)$, and then 29 seconds of activity $\mathcal{M}'(i)$. Therefore, $\mathrm{R52}(0, \mathcal{S}, \mathcal{U})(i) = \mathcal{M}(i)$. Now for the other condition: $\mathrm{R52}(d, \mathcal{S}, \mathcal{U})(i) = A(d, \mathcal{S}, i)$, which is the activity of the longest interval in $[d+60i, d+60i+60)$. By the definition of $\mathcal{S}$, in that interval there are first $31-d$ seconds of activity $\mathcal{M}(i)$, then 30 seconds of $\mathcal{M}'(i)$, and then $d-1$ seconds of $\mathcal{M}(i+1)$. If $d > 1$, then the activity with the longest interval is $\mathcal{M}'(i)$. If $d = 1$, then there is a draw between $\mathcal{M}(i)$ and $\mathcal{M}'(i)$, but the latest is $\mathcal{M}'(i)$. In any case, $\mathrm{R52}(d, \mathcal{S}, \mathcal{U})(i) = \mathcal{M}'(i)$.

This construction can be visualized in Figure 7. □

Informally, the theorem says that, given two minute labellings and a shift, there is a second labelling that the driver might have recorded which gives the first minute labelling in one calendar and the second minute labelling in the shifted calendar. In particular, the same activities from a driver can lead to a minute labelling with only driving or only resting. Requirement (51) does not solve this problem; it only places some mild restrictions on the labellings $\mathcal{M}$, $\mathcal{M}'$, as defined below.

Definition 6.2. A minute labelling $\mathcal{M} : \mathbb{Z} \to \mathbb{A}$ is said to be feasible if, for every $i \in \mathbb{Z}$ such that $\mathcal{M}(i-1) = \mathcal{M}(i+1) = \textsc{driving}$, we have that $\mathcal{M}(i) = \textsc{driving}$.

It is easy to see that only feasible minute labellings can be obtained by applying Requirement (51) to minute labellings. As a direct consequence of the above considerations, we obtain the following theorem.

Theorem 6.3. Let d be a time shift such that $1 \leq d \leq 59$. Let $\mathcal{U}$ be the unknown minute labelling and let $\mathcal{M}$ and $\mathcal{M}'$ be two feasible minute labellings. Then there exists a second labelling $\mathcal{S}$ such that $\mathrm{R51}\big(\mathrm{R52}(0, \mathcal{S}, \mathcal{U})\big) = \mathcal{M}$ and $\mathrm{R51}\big(\mathrm{R52}(d, \mathcal{S}, \mathcal{U})\big) = \mathcal{M}'$.

Thus we may obtain any two possible outcomes from the second labelling, provided a shift is applied. This is problematic, both because it is hard to argue that the same sequence of activities should change their legality if performed with a few seconds of delay, and because such shifts do occur due to whether the tachograph takes leap seconds or not. With this we conclude our discussion of issues with labellings, and in the sequel consider the problem of verifying that a given minute labelling complies with the law, assuming it has been suitably obtained from the original tachograph data.

7. Temporal and monadic logics

In order to evaluate the legality of a driving record, we need to transform the law into a format that may be checked algorithmically. We argue that driving records are essentially linear temporal logic (LTL) models, and hence well-understood logics such as LTL or monadic second order logic (MSO) may be suitable for representing the algorithmic content of the law.

Our computational priority is efficient model-checking, as the property 'The driving record $\mathcal{M}$ complies with the regulation φ' must be routinely verified using specialized software. Since MSO has decidable model-checking, it is preferable over alternatives such as second order arithmetic. In MSO, checking whether $\mathcal{M} \models \varphi$ can be solved in time linear on $|\mathcal{M}|$ for *fixed* φ. This is already good news, as checking for compliance with regulations in other contexts may be hard (e.g. the case of business processes [9]).

However, the problem is non-elementary when both $\mathcal{M}$ and φ are allowed to vary [10]. No model-checker can be expected to uniformly work efficiently for all choices of φ, so that changes in the law can turn software obsolete. Thus it is be desirable for regulations to be representable within a fragment of MSO with tractable model-checking. To this end, we will use LTL as a 'canonical' tame fragment of MSO; we further elaborate on this choice below.

7.1. Linear temporal logic

Linear temporal logic is based on the language $\mathcal{L}_{\Box\,\mathsf{U}}$ given by the grammar $\varphi, \psi := \bot \mid P \mid \varphi \to \psi \mid \bigcirc\varphi \mid \Box\varphi \mid \varphi\,\mathsf{U}\,\psi$, where P is an element of a countable set $\mathbb{P}$ of predicate symbols. We read $\bigcirc$ as 'next', $\Box$ as 'henceforth' and U as 'until'.

We will always interpret formulas of $\mathcal{L}$ over the structure $(\mathbb{N}, S)$, where $S(n) = n + 1$. Hence, for our purposes, an LTL model is merely a function $\cdot^{\mathcal{M}} \colon \mathbb{P} \to 2^{\mathbb{N}}$. We define the satisfaction relation $\models$ inductively by

1. $(\mathcal{M}, n) \models P$ iff $n \in P^{\mathcal{M}}$
2. $(\mathcal{M}, n) \not\models \bot$
3. $(\mathcal{M}, n) \models \varphi \to \psi$ iff $(\mathcal{M}, n) \not\models \varphi$ or $(\mathcal{M}, n) \models \psi$
4. $(\mathcal{M}, n) \models \bigcirc\varphi$ iff $\big(\mathcal{M}, S(n)\big) \models \varphi$
5. $(\mathcal{M}, n) \models \Box\varphi$ iff for all $k \geq 0$ we have that $\big(\mathcal{M}, S^k(n)\big) \models \varphi$
6. $(\mathcal{M}, n) \models \varphi \,\mathsf{U}\, \psi$ iff there exists $k \geq 0$ such that $\big(\mathcal{M}, S^k(n)\big) \models \psi$ and $\forall\, i \in [0, k)$, $\big(\mathcal{M}, S^i(n)\big) \models \varphi$

As usual, a formula φ is *satisfiable* over a set of models Ω if there is $\mathcal{M} \in \Omega$ and $n \in \mathbb{N}$ such that $(\mathcal{M}, n) \models \varphi$; and *valid* on Ω if for every $\mathcal{M} \in \Omega$ and $n \in \mathbb{N}$ we have $(\mathcal{M}, n) \models \varphi$.

We will consider the U-free fragment $\mathcal{L}_{\Box}$, the $\Box$-free fragment $\mathcal{L}_{\mathsf{U}}$ and the $\Box, \mathsf{U}$-free fragment, $\mathcal{L}_{\circ}$. We define other Booleans and $\Diamond$ as abbreviations in the standard way. Note that $\mathcal{L}_{\mathsf{U}}$ is expressively equivalent to $\mathcal{L}_{\Box\,\mathsf{U}}$, so we will seldom work over the full language. We could additionally consider past tenses, but they do not add expressive power to $\mathcal{L}_{\mathsf{U}}$ in models with a starting point (although there are issues with succinctness which we briefly discuss).

The articles we consider also require some counting, but this can be dealt with using the following abbreviations, where $n, m \in \mathbb{N}$. Below, an empty disjunction should be read as $\bot$ and an empty conjunction as $\top$.

- $\bigcirc^0\varphi := \varphi$ and $\bigcirc^{n+1}\varphi = \bigcirc\bigcirc^n\varphi$;
- $\Diamond^{<n}\varphi = \bigvee_{i=0}^{n-1} \bigcirc^n\varphi$ and $\Box^{<n}\varphi = \bigwedge_{i=0}^{n-1} \bigcirc^n\varphi$.

Variants with $\leq n$ instead of $<n$ are defined by reading $\leq n$ as $<n+1$.

Given any formula φ and a set $\Theta \subseteq \{\bigcirc, \Box, \mathsf{U}\}$, we define the Θ-*depth* of φ (in symbols, $\mathrm{dpt}_{\Theta}(\varphi)$) to be the nesting depth of tenses in Θ, defined in the standard way. If $\Theta = \{\vartheta\}$ we write ϑ-*depth* and $\mathrm{dpt}_{\vartheta}(\cdot)$ instead of Θ-depth and $\mathrm{dpt}_{\Theta}(\cdot)$, and if $\Theta = \{\bigcirc, \Box, \mathsf{U}\}$ we write *temporal depth* and $\mathrm{dpt}(\cdot)$ instead of Θ-*depth* and $\mathrm{dpt}_{\Theta}(\cdot)$. As a general rule we consider the $\bigcirc$-depth to be a negligible complexity measure with respect to the depths of the other tenses.

Let us say a few words about our choice of LTL as a 'canonical' tame fragment. First, the type of lower bounds we give require a concrete logic, so a choice must be made for the sake of our case-study. LTL is well-known and has relatively simple syntax and semantics. In practice one would want to use

'sugared' versions of LTL, such as metric temporal logic (MTL; [11]), which allow for expressions such as $\bigcirc^{n+1}$ to be represented succinctly. Thus we focus on U-depth rather than $\bigcirc$-depth or formula size; indeed, the standard translation of MTL into LTL does not increase U-depth. This leaves open the possibility of entirely different tame fragments of MSO (or even unrelated logics) that may better capture the law, such as logics based on intervals [12]. Our team is actively investigating such alternatives, but has not yet found an option with a substantial technical advantage over LTL.

7.2. Monadic second-order logic

Define a *term* to be given by the grammar $t := 0 \mid x \mid S(t)$, where x belongs to some fixed set of first-order variables $\mathbb{V}$. Then, the language $\mathcal{L}^2_\forall$ is defined by the grammar

$$\varphi, \psi := \bot \mid P(t) \mid t < s \mid \varphi \to \psi \mid \forall x\, \varphi \mid \forall P\, \varphi,$$

where x is a variable, t and s terms, and $P \in \mathbb{P}$. Once again we define other Booleans and $\exists$ as standard abbreviations, and define $\mathcal{L}^1_\forall$ to be the sub-language of $\mathcal{L}^2_\forall$ that does not allow quantifiers over elements of $\mathbb{P}$.

The language $\mathcal{L}^2_\forall$ is interpreted over models $\cdot^{\mathcal{M}} \colon \mathbb{V} \cup \mathbb{P} \to \mathbb{N} \cup 2^{\mathbb{N}}$ such that $x^{\mathcal{M}} \in \mathbb{N}$ if x is a variable and $P^{\mathcal{M}} \subseteq \mathbb{N}$ if P is a predicate symbol. For a variable x and $n \in \mathbb{N}$, let $\mathcal{M}[x/n]$ be the model that is the same as $\mathcal{M}$ except that $x^{\mathcal{M}[x/n]} = n$, and for $P \in \mathbb{P}$ and $A \subseteq \mathbb{N}$ define $\mathcal{M}[P/A]$ analogously. Extend $\cdot^{\mathcal{M}}$ to terms by defining recursively $0^{\mathcal{M}} = 0$ and $(S(t))^{\mathcal{M}} = t^{\mathcal{M}} + 1$. The satisfaction relation is then defined as follows:

1. $\mathcal{M} \not\models \bot$
2. $\mathcal{M} \models P(t)$ iff $t^{\mathcal{M}} \in P^{\mathcal{M}}$
3. $\mathcal{M} \models \varphi \to \psi$ iff $\mathcal{M} \not\models \varphi$ or $\mathcal{M} \models \psi$
4. $\mathcal{M} \models \forall x\, \varphi$ iff for all $n \in \mathbb{N}$, $\mathcal{M}[x/n] \models \varphi$
5. $\mathcal{M} \models \forall P\, \varphi$ iff for all $A \subseteq \mathbb{N}$, $\mathcal{M}[P/A] \models \varphi$

Satisfiability and *validity* are defined as before. MSO denotes the language $\mathcal{L}^2_\forall$ endowed with these semantics, and MFO denotes MSO restricted to $\mathcal{L}^1_\forall$. In order to unify our semantics for LTL and MSO, we regard an LTL model $\mathcal{M}$ as an MSO model by setting $x^{\mathcal{M}} = 0$ for all variables, and similarly regard an MSO model as an LTL model by restricting the domain to $\mathbb{P}$.

We say that a set Ω of models is *definable* in a language $\mathcal{L} \subseteq \mathcal{L}_{\Box \mathsf{U}}$ if there is φ in $\mathcal{L}$ such that for any model $\mathcal{M}$ we have $(\mathcal{M}, 0) \models \varphi$ if and only if $\mathcal{M} \in \Omega$. Similarly, Ω is definable in $\mathcal{L} \subseteq \mathcal{L}^2_\forall$ if there is φ in $\mathcal{L}$ such that for any model $\mathcal{M}$, $\mathcal{M} \models \varphi$ if and only if $\mathcal{M} \in \Omega$. With this in mind, we may regard MFO as a temporal logic in terms of the following.

Theorem 7.1 (Kamp [13])**.** Let Ω be a set of LTL models. Then, Ω is definable in $\mathcal{L}_{\mathsf{U}}$ if and only if it is definable in $\mathcal{L}^1_\forall$.

When discussing expressivity, we will go back and forth between LTL and MFO depending on which is more convenient for the application at hand.

8. Expressibility

In this section we show how the legal articles we have considered could be represented within monadic second order logic. It is crucial to stress that the articles allow for some interpretation and thus certain elements may admit readings different from those we propose. We will also make a few simplifying assumptions for the sake of exposition. From discussions with legal experts, we believe that our interpretations are reasonable modulo the aforementioned simplifying assumptions.

We have discussed previously how tachograph data is recorded second by second, and then translated into a minute by minute format. However, we will consider hourly labellings from now on for the sake of readability. Recall that we defined the set of activities $\mathbb{A} = \{\text{DRIVING}, \text{REST}, \text{AVAILABILITY}, \text{WORK}, \text{UNKNOWN}\}$. Each activity will be regarded as a propositional variable. Then a labelling $\mathcal{M}$ (in the sense of Section 4) with domain $\mathbb{N}$ may be viewed as an LTL model by setting $P^{\mathcal{M}} = \{n \in \mathbb{N} : \mathcal{M}(n) = P\}$ for $P \in \mathbb{A}$, and $P^{\mathcal{M}} = \varnothing$ for all other propositional variables. We use $\mathbb{N}$ as domain so that we may have Theorem 7.1 available. It is also possible to work with $\mathbb{Z}$ as in previous sections, but the analogue of Theorem 7.1 in this setting would require additional 'past' tenses, which would make some upcoming proofs more tedious.

We also introduce a predicate symbol WEEK which holds on the first hour of each Monday. This condition can be treated model-theoretically – i.e. models are assumed to be equipped with a correct valuation for WEEK– or syntactically by the $\mathcal{L}_\Box$ axiom

$$\text{WEEK} \wedge \Box\big(\text{WEEK} \rightarrow (\bigcirc \Box^{<167} \neg \text{WEEK} \wedge \bigcirc^{168} \text{WEEK})\big)$$

(assuming that the model begins on the first hour of a Monday). LTL models satisfying this formula at zero are called *weekly models.* With this in mind, we proceed to illustrate how the content of the legislation could be represented. However, since we want to isolate possible sources of impredicativity (i.e., second order quantification), we will work with simplified variants of the legislation that are more suitable for expository purposes.

8.1. Article §8.9

Article §8.6 requires that each two week period be assigned two rest periods with some additional constraints, and §8.9 indicates how rest periods should be assigned to specific weeks. Our goal in this subsection is to explore the possible impredicativity arising from the assignment itself, independently of the additional conditions of §8.6. Every week should contain at least one 24 hour rest period, but this by itself would not be sufficient to comply with §8.6. On the other hand, a driver resting 45 hours each week would comply with §8.6, so this would be a sufficient, but not necessary, condition for compliance. In order to not commit to either condition, we will consider the following general property: when is it that each week can be assigned a rest period of at least d hours, so that each rest period intersects the week it is assigned to? This simplified condition is already *prima facie* impredicative, as it requires a function mapping rest intervals to weeks. Thus it may be surprising that it can actually be defined in first order logic (and hence in LTL).

Theorem 8.1. Given $d \in [2, 85]$, there is an $\mathcal{L}^1_\forall$-formula $\varphi = \varphi_d \in \mathcal{L}^1_\forall$ such that given any LTL model $\mathcal{M}$, $\mathcal{M} \models \varphi$ if and only if there is an assignment of weekly rest periods such that every week is assigned a rest period of length at least d.

Proof. In this proof we will assume that variables range over weeks. It is clear that using our fundamental ontology this can be established in first order logic, as a week can be identified with its starting point, which is already marked by the predicate WEEK. Let $E(x)$ be a formula which holds if and only if x is a week with an *early* rest period (of length at least d), which means that it overlaps with the previous week, $L(x)$ a formula that holds if x contains a *late* rest period overlapping with the following week, and $I(x)$ be a formula that holds if and only if x is a week with an *internal* rest period disjoint from (but possibly contiguous with) any early or late rest periods in the week x.

Clearly, E, I, L are first-order definable (although their definition depends on d). The condition $d \leq 85$ ensures that if $E(x) \wedge L(x)$ holds then the week x contains disjoint early and late rest periods.[3] Define $\check{E}(x) = E(x) \wedge \neg I(x) \wedge \neg L(x)$, and define $\check{I}(x)$, $\check{L}(x)$ analogously. Then set

$$\begin{aligned}\varphi = &\forall x \left(E(x) \vee I(x) \vee L(x)\right) \\ &\wedge \ \forall x \, \forall y \left(x < y \wedge \check{L}(x) \wedge \check{E}(y) \rightarrow \exists z \in (x, y) \ I(z)\right).\end{aligned}$$

We claim that φ holds if and only if there is an assignment such that each week is assigned one rest period of length at least d. First assume that such an assignment exists. Clearly $\forall x \left(E(x) \vee I(x) \vee L(x)\right)$ holds, since if x were a counterexample no rest period could be assigned to week x.

Now, suppose that $x < y$ are such that $\check{L}(x) \wedge \check{E}(y)$, and choose x, y such that $y - x$ is minimal among all such pairs. Note that x is assigned to its late rest period (as this is the only one available) and y is assigned to its early rest period. It follows that there is a least $z \in (x, y]$ that is not assigned to its late rest period. By minimality $z-1$ is assigned to its late rest period, hence z cannot be assigned to its early rest period. However, z must be assigned to *some* rest period by assumption, and since this rest period is neither early nor late, the week of z must contain some internal rest period, and $I(z)$ holds.

Now assume that φ holds and define an assignment recursively as follows. Let R be a rest period and suppose that all earlier rest periods have been assigned to some week. If R is internal, assign it to its current week. If R is late for week w and w has not been assigned a rest period, assign R to w. Otherwise, assign R to $w + 1$.

We prove by induction that every week is assigned to some rest period. Fix y and assume that all earlier weeks have been assigned to some period. Note that $E(y) \vee I(y) \vee L(y)$ holds by φ. If $I(y)$ holds then the week of y has a rest period assigned to it. If $L(y)$ holds then the late rest period of y is assigned to it, unless an earlier one was already assigned to it. So we are left with the hypothetical case where $\check{E}(y)$ holds, and the early rest period of y has been assigned to $y-1$. Let $x < y$ be minimal with the property that

[3] If $E(x)$ holds then we need to assign at most $d-1 \leq 84$ resting hours of x to the early rest period, and likewise for $L(x)$. In a week there are $7 \times 24 = 84 \times 2$ hours. Hence if $E(x) \wedge L(x)$ holds, we can assign hours from the first half of x towards the early rest period, and from the second half towards the late rest period, ensuring that they are disjoint.

every $z \in [x, y)$ has had its late rest period assigned to it. First note that $E(x)$ fails, since otherwise $x > 0$ and either $x - 1$ has had its late rest period assigned to it, contradicting the minimality of x, or else the early rest period of x would have been assigned to the week of x by our recursion. Note also that $I(z)$ fails for all $z \in [x, y)$, since any internal rest period is automatically assigned to the current week. We conclude that $\check{L}(x)$ holds and $I(z)$ fails for all $z \in (x, y)$, thus φ fails. □

8.2. Article §8.6

Now that we have seen that the possibility of assigning rest periods is not itself impredicative, we isolate the compensation mechanism from the rest assignments and analyse it in a similar fashion. We claim that the content of Article §8.6 admits a representation in $\mathcal{L}^2_\forall$ by a Σ^1_1 formula over the class of weekly models.

Before we continue, we mention some remarks regarding our interpretation of the law. A weekly rest period must be compensated in the future, but it may be compensated on the same week as the rest period itself. The compensation must fall entirely within the union of the current calendar week and the subsequent three calendar weeks. We will further assume that, after compensation, each week should have a weekly rest period of *exactly* 45 hours assigned; this is not an issue, since if a week has a longer weekly rest period, a portion of that rest period can simply be 'unassigned'. Finally, we assume that each week is assigned a single weekly rest. The legislation does not explicitly forbid weeks with two weekly rests, but we find this to be the most intuitive interpretation.

Our representation will use the following second order variables: R_E is a variable meant to denote the union of all early weekly rest periods, meaning that they begin on the week previous to the one they are assigned to; R_I is a variable meant to denote the union of all intermediate weekly rest periods, meaning that they are entirely contained in the week they are assgined to; R_L is a variable meant to denote the union of all late weekly rest periods, meaning that they intersect the week after the one they are assigned to; C_0, C_1, C_2, and C_3 are variables meant to denote periods of compensation, such that C_0 compensates the current weekly rest, C_1 compensates the previous weekly rest, C_2 compensates the weekly rest of two weeks ago, and C_3 compensates the weekly rest of 3 weeks ago.

If W is a week, let $S(W)$ be the successor week to W. We express §8.6

by a formula

$$\psi_{\S 8.6} := \exists\, R_E\ \exists\, R_I\ \exists\, R_L\ \exists\, C_0\ \exists\, C_1\ \exists\, C_2\ \exists\, C_3\, \psi^0_{\S 8.6},$$

where $\psi^0_{\S 8.6}$ contains no second order quantifiers.[4] The formula $\psi^0_{\S 8.6}$ is a conjunction over formulas expressing the following properties:

- The sets $R_E, R_I, R_L, C_0, C_1, C_2, C_3$ are mutually disjoint and all contained in REST.
- Each of R_E and R_L is a union of non-contiguous intervals of length at least 24 and no more than 45.
- For any week W, $R_I \cap W$ is an interval which is either empty or of length at least 24 and no more than 45.
- Given a week W, there is a unique maximal interval R_W, called *the rest period assigned to* W, such that $R_W \cap W \neq \varnothing$ and either a) $R_W \subseteq R_E$ and the beginning of R_W lies before W, b) $R_W \subseteq R_I \cap W$, or c) $R_W \subseteq R_L$ and R_W intersects $S(W)$.
- Given a week W, $R_{S(W)}$ must start no later than 6×24 hours after R_W, and at least one of R_W, $R_{S(W)}$ must be a 45-hour interval.
- Given a week W, we define $C_W = \bigcup_{i=0}^{3} C_i \cap S^i(W)$ and add the following conditions: a) C_W is an interval, disjoint from R_W, and begins after R_W ends; b) There is an interval J such that $C_W \subseteq J \subseteq$ REST and with $|J \setminus C_W| \geq 9$[5]; c) $|R_W \cup C_W| = 45$.

It should be clear that each of these conditions is first order definable, hence $\psi_{\S 8.6}$ is Σ^1_1. Moreover, some inspection shows that over the set of weekly models, $\psi_{\S 8.6}$ coincides with our reading of §8.6. Indeed, the variables R_E, R_I, and R_L are used as auxiliary variables to define R_W, the weekly resting

[4] We remark that we can quantify in first order logic over intervals, since an interval $[a, b]$ can be identified with its endpoints. Conditions of the form $|X| \geq n$, where n is fixed, are expressed by stating that there exist distinct $x_1, \ldots, x_n$ belonging to X.

[5] Note that in our reading of §8.7, we allow $J \cap C_{W'} \neq \varnothing$ for some week $W' \neq W$. This is because the law does not explicitly state that the nine-hour rest period attached to a compensation period cannot be used to compensate an additional week. However, a more strict reading where we demand $J \cap C_{W'} = \varnothing$ can be formalized similarly.

period assigned to W. If R_W is a reduced rest, the variables C_i are then used to define the compensation period C_W. Once these auxiliary variables are fixed, checking whether each weekly rest is of suitable lenght or has been compensated appropriately is a first order property.

We conclude that the content of §8.6 admits a Σ^1_1 representation over the set of weekly models, as claimed.

9. Stratified bisimulations

In this section we present a version of stratified bisimulations for $\mathcal{L}_{\mathsf{U}}$ proposed by Kurtonina and de Rijke [14]. Since all languages we consider contain Booleans and $\bigcirc$, it is convenient to begin with a 'basic' notion of bisimulation for this language.

Definition 9.1. Given $k \geq 0$ and two LTL models $\mathcal{M}$ and $\mathcal{N}$, a binary relation $Z \subseteq \mathbb{N}^2$ is a *k-$\bigcirc$-bisimulation (between $\mathcal{M}$ and $\mathcal{N}$)* if whenever $x\ Z\ y$, $P \in \mathbb{P}$, and $j \leq k$, we have $x + j \in P^{\mathcal{M}}$ iff $y + j \in P^{\mathcal{N}}$.

We will use bounded $\bigcirc$-bisimulations as a basis to define bounded bisimulations for more powerful languages.

Definition 9.2. Fix $k \geq 0$ and two LTL models $\mathcal{M}$ and $\mathcal{N}$. Let $\vec{Z} = (Z_i)_{i=0}^{\infty}$ be a sequence such that for all $i \in \mathbb{N}$, Z_i is a k-$\bigcirc$-bisimulation and $Z_{i+1} \subseteq Z_i$.

1. $\vec{Z}$ is a *k-$\Box$-bisimulation (between $\mathcal{M}$ and $\mathcal{N}$)* if whenever $x\ Z_{i+1}\ y$:

   Forth $\Box$. For all $x' \geq x$ there exists $y' \geq y$ such that $x'\ Z_i\ y'$.

   Back $\Box$. For all $y' \geq y$ there exists $x' \geq x$ such that $x'\ Z_i\ y'$.

2. $\vec{Z}$ is a *k-U-bisimulation (between $\mathcal{M}$ and $\mathcal{N}$)* if whenever $x\ Z_{i+1}\ y$:

   Forth U. For all $x' \geq x$ there exists $y' \geq y$ and a function $\xi\colon [y, y'] \to [x, x']$ such that every $z \in [y, y']$ satisfies $\xi(z)\ Z_i\ z$ and $\xi(z) = x'$ if and only if $z = y'$.

   Back U. For all $y' \geq y$ there exists $x' \geq x$ and a function $\eta\colon [x, x'] \to [y, y']$ such that every $z \in [x, x']$ satisfies $z\ Z_i\ \eta(z)$ and $\eta(z) = y'$ if and only if $z = x'$.

Stratified bisimulations are an essential tool in proving inexpressivity or succinctness results, given that they preserve the truth of formulas of small enough nesting depth.

Lemma 9.3 (Kurtonina and de Rijke [14])**.**

1. Given two LTL models $\mathcal{M}$ and $\mathcal{N}$ and a k-$\Box$-bisimulation $\vec{Z}$ between them, for all formulas $\varphi \in \mathcal{L}_{\Box}$ and for all $(x, y) \in Z_i$, if φ has $\bigcirc$-depth at most k and $\Box$-depth at most i then $(\mathcal{M}, x) \models \varphi$ iff $(\mathcal{N}, y) \models \varphi$.

2. Given two LTL models $\mathcal{M}$ and $\mathcal{N}$ and a k-U-bisimulation $\vec{Z}$ between them, for all formulas $\varphi \in \mathcal{L}_{\mathsf{U}}$ and for all $(x, y) \in Z_i$, if φ has $\bigcirc$-depth at most k and U-depth at most i then $(\mathcal{M}, x) \models \varphi$ iff $(\mathcal{N}, y) \models \varphi$.

In the next section we use Lemma 9.3 to show that certain legal properties we have considered are hard or impossible to define in fragments of linear temporal logic.

10. Non-expressibility

We have seen that Articles §8.9 and §8.6 are expressible in MFO and MSO, respectively. We will see that they are not expressible in $\mathcal{L}_{\Box}$ and that §8.6 is not reasonably expressible in $\mathcal{L}_{\mathsf{U}}$. For this, we use constructions similar to the examples given in Section 3. However, since these constructions will be somewhat more elaborate, we settle some notation first.

Say that a model $\mathcal{M}$ is *eventually resting* if there is some m such that for all $n > m$ and all $P \in \mathbb{A}$, $n \in P^{\mathcal{M}}$ iff $P = \textsc{rest}$. The *end* of an eventually resting model is the least such value of m which is also a multiple of 168 (i.e., a whole number of weeks). A week-long model is an eventually resting models whose end is 168. We define the *concatenation* of two eventually resting models $\mathcal{A}, \mathcal{B}$, denoted $\mathcal{A} \mid \mathcal{B}$, as follows. Let m be the end of $\mathcal{A}$. Then, for a predicate symbol P and $n \in \mathbb{N}$, we set

$$n \in P^{\mathcal{A}|\mathcal{B}} \Leftrightarrow \begin{cases} n \in P^{\mathcal{A}} & \text{if } n \leq m \\ n - m \in P^{\mathcal{B}} & \text{if } n > m. \end{cases}$$

If k is a natural number then $\mathcal{A}^k$ denotes k concatenated copies of $\mathcal{A}$. If $n \in [24, 168)$, then n denotes a week with one weekly resting period of n; we assume that these weekly periods fall in the middle of each week without overlapping with other weeks, with the details being non-essential. However, we do assume that any two instances of the week represented by n are identical.

It will be convenient to represent a given moment in time both by the number of hours t since the beginning of time, and by $168w + h$, where w is the number of weeks since the beginning of time, and $h < 168$ is the number of hours since the beginning of that week.

10.1. Article §8.9

We have seen that the possibility of assigning weekly rest periods to each week is first order definable. One may then ask if $\mathcal{L}_\Box$ suffices to define it, and the answer is negative. We prove this via the following construction.

Definition 10.1. Fix $d \in [24, 84]$. Define the following week-long models:

- E is a model whose first $\lfloor d/2 \rfloor$ hours are resting.
- I is a model whose hours $(\lfloor d/2 \rfloor + 1, \lfloor d/2 \rfloor + d)$ are resting.
- L is a model whose last $\lceil d/2 \rceil$ hours are resting.
- Concatenations of letters denote unions of resting hours, i.e., EL denotes a week with a beginning and an end rest period.

Then, for each $n \in \mathbb{N}$, define the eventually resting models $\mathcal{A}_n = (L \mid EL^n \mid EIL \mid EL^n \mid E)^{n+1}$ and $\overline{\mathcal{A}}_n = L \mid EL^n \mid E \mid \mathcal{A}_n$.

Given $d \in [24, 84]$ and a model $\mathcal{M}$, we say that $\mathcal{M}$ *admits a weekly rest assignment* if it is possible that each week is assigned a weekly rest period of length at least d.

Lemma 10.2. The model $\mathcal{A}_n$ admits a weekly rest assignment but $\overline{\mathcal{A}}_n$ does not.

Proof. It is easy to see that $\mathcal{A}_n$ satisfies the formula φ_d of Theorem 8.1 and that $\overline{\mathcal{A}}_n$ does not. □

Lemma 10.3. There is a bounded $168n$-$\Box$-bisimulation $\vec{Z}$ between $\mathcal{A}_n$ and $\overline{\mathcal{A}}_n$ such that $0\ Z_n\ 0$.

Proof. Define $r := 2n + 3$ and for $x = 168w + h, y = 168v + \ell \in \mathbb{N}$, let $x\ Z_i\ y$ if $h = \ell$ and one of the following holds: (A1) $x = y = 0$ and $i \leq n$, (A2) $0 < v$, $\max\{w, v - n - 2\} \leq (n - i)r$ and $v \equiv w + n + 2 \pmod r$, or (A3) $v = w + n + 2$. We need to show that $\vec{Z}$ is a stratified bisimulation.

It is clear that $Z_{i+1} \subseteq Z_i$. Assume that $x\ Z_i\ y$ and write $x = 168w + h$, $y = 168v + \ell$; note that by definition we must have $h = \ell$. If $i = 0$, then some inspection shows that x and y share the same formulas of the form $\bigcirc^j p$ with $j \leq 168n$, since the current and subsequent n weeks are of the same form. It is sufficient to check this for Z_0 because it contains all the Z_i's.

Otherwise, change variables so that $x \sim_{i+1} y$; we check that the required clauses hold.

FORTH $\Box$. Let $x' \geq x$ and write $x' = 168w' + h'$. We claim that there is v' such that $168v' + h' \geq 168v + h$ and $168w' + h'\ Z_i\ 168v' + h'$. If $168(w' + n + 2) + h' \geq 168v + h$ we may take $v' = w' + n + 2$, and the bisimulation holds by (A3). Otherwise, we have $v \geq w' + n + 2 \geq w + n + 2$, where the first inequality is strict unless $h' < h$, in which case the second inequality must be strict. Hence x, y do not satisfy (A1) nor (A3) and thus $\max\{w, v - n - 2\} \leq (n - i - 1)r$. Take $v' \in (v, v + r]$ with $v' \equiv w' + n + 2 \pmod r$ and set $y' = 168v' + h'$. Note that $w' + n + 2 \leq v < (n - i - 1)r + n + 2$ yields $w' \leq (n - i)r$, while $v' \leq v + r \leq (n - i - 1)r + r = (n - i)r$, and thus $v - n - 2 \leq (n - i)r$ as well. Thus $x'\ Z_i\ y'$ by (A2).

BACK $\Box$. Let $y' \geq y$ and write $y' = 168v' + h'$. As before, we claim that there is w' such that $168w' + h' \geq 168w + h$ and $168w' + h'\ Z_i\ 168v' + h'$. If $168(v' - n - 2) + h' \geq 168w + h$ we may take $w' = v' - n - 2$, and the result follows by (A3). Otherwise, we have $w \geq v' - n - 2 \geq v - n - 2$ with one inequality being strict,so that x, y do not satisfy (A3). If x, y satisfy (A2), then $\max\{w, v - n - 2\} \leq (n - i - 1)r$. If x, y satisfy (A1), we have that $w = v = 0$ and $i + 1 \leq n$, so that $\max\{w, v - n - 2\} = 0 \leq (n - i - 1)r$ as well. Take $w' \in (w, w + r]$ with $w' + n + 2 \equiv v' \pmod r$ and set $x' = 168w' + h'$. It is not hard to check that $\max\{w', v' - n - 2\} \leq (n - i)r$. Thus $x'\ Z_i\ y'$ by (A2).

Theorem 10.4. Given $d \in [24, 84]$, there is no $\mathcal{L}_\Box$ formula φ such for every model $\mathcal{M}$, $\mathcal{M} \models \varphi$ if and only if $\mathcal{M}$ admits a weekly rest assignment.

Proof. Suppose that $\varphi \in \mathcal{L}_\Box$ is such a formula. Let $d_\circ$ and $d_\Box$ be its $\bigcirc$-depth and $\Box$-depth, respectively. Choose n such that $d_\circ \leq 168n$ and $d_\Box \leq n$. Then by Lemmas 9.3 and 10.3, $(\mathcal{A}_n, 0) \models \varphi$ iff $(\overline{\mathcal{A}}_n, 0) \models \varphi$. But, according to Lemma 10.2, $\mathcal{A}_n$ admits a weekly rest assignment, while $\overline{A}_n$ does not. $\Box$

10.2. Article §8.6

Our goal now is to show that Article §8.6 is not expressible in $\mathcal{L}_\Box$, and that it needs a formula with a large U-depth to express it in $\mathcal{L}_\mathsf{U}$. As before,

we start by defining a model that complies with the article, and one that doesn't, and then prove that they are bisimilar.

Definition 10.5. For each $n \in \mathbb{N}$, we define the following weekly models: $\mathcal{B}_n = (44 \mid 45^n \mid 46 \mid 45^n)^n \mid 24 \mid 45 \mid 24$ and $\overline{\mathcal{B}}_n = 44 \mid 45^n \mid \mathcal{B}_n$.

Lemma 10.6. Given $n \in \mathbb{N}$, $\mathcal{B}_n \models \psi_{\S 8.6}$ but $\overline{\mathcal{B}}_n \not\models \psi_{\S 8.6}$.

Proof. In $\mathcal{B}_n$, the first week's missing hour can be compensated on the third week. This creates a chain reaction of compensations, as the third week also needs to be compensated (because it's interpreted as a reduced rest of 44 hours together with a compensation of 1 hour). However, it is always possible to compensate either two weeks after, or on the week of 46 hours, if it is close enough. It is thus never necessary to use up hours from the second block of n 45 hour rest weeks, which are all regular rest periods. This process happens n times, until we reach the last three weeks of the model. Two of them need to be compensated, but it is possible to do so using the unlimited hours of rest available after the end.

Consider now $\overline{\mathcal{B}}_n$. The 24 hour weeks near the end of the model cannot be used to compensate previous weeks, since 24 is the minimum allowed weekly rest. The last 45 hour week cannot be used to compensate previous weeks either, because then there would be more than one consecutive week with no regular rest period. Thus, we erase the last three weeks from consideration. There are $m := 2n^2 + 3n + 1$ weeks in the rest of the model, $2n^2 + n$ of which have 45 rest hours, $n + 1$ of which have 44 rest hours, and n of which have 46 hours, for a total of $45m - 1$ rest hours. Thus there are not enough rest hours to distribute among the period such that each week is assigned 45 hours of weekly rest. □

Lemma 10.7. There is a stratified $168n$-$\Box$-bisimulation $\vec{Z}$ between $\mathcal{B}_n$ and $\overline{\mathcal{B}}_n$ such that $0\ Z_n\ 0$.

Proof. The stratified bisimulation and the proof are analogous to those used in the proof of Lemma 10.3. □

Theorem 10.8. There is no $\mathcal{L}_\Box$-formula equivalent to $\psi_{\S 8.6}$ over the class of weekly models.

Proof. Suppose that $\psi \in \mathcal{L}_\Box$ is a formula expressing Article §8.6 with $\bigcirc$-depth $d_\circ$ and $\Box$-depth $d_\Box$. Choose n big enough to ensure that $d_\circ \leq 168n$

and $d_\Box \leq n$, and let $\vec{Z}$ be the bisimulation of Lemma 10.7. Then by Lemma 9.3, $(\mathcal{B}_n, 0) \models \psi$ iff $(\overline{\mathcal{B}}_n, 0) \models \psi$. This contradicts Lemma 10.6. $\square$

Now we show that any formula of $\mathcal{L}_{\mathsf{U}}$ requires nesting depth 20 of U.

Definition 10.9. For $n \in \mathbb{N}$, we define models: $\mathcal{C}_n = (44 \mid 45^{2n+1})^{21} \mid 66 \mid 24 \mid 45 \mid 24$ and $\overline{\mathcal{C}}_n = (44 \mid 45^{2n+1}) \mid \mathcal{C}_n$.

Lemma 10.10. Given $n \in \mathbb{N}$, $\mathcal{C}_n \models \psi_{\S 8.6}$ but $\overline{\mathcal{C}}_n \not\models \psi_{\S 8.6}$.

Proof. First we see that $\mathcal{C}_n \models \psi_{\S 8.6}$. Intuitively, even weeks are compensated two weeks later, and the size of the compensation increases by one every $2n+2$ weeks. Thus for example one hour of week 0 is compensated by one hour of week 2, which is compensated by one hour of week 4, and so on until we reach week $2n+2$. Note however that this week only has 44 hours of rest and has used one hour to compensate the previous week, so we need to compensate two hours of rest. This is compensated by two hours on week $2n+4$, and so on until we reach the third 44 hour rest. Since two hours of this rest are used to compensate a previous week, now three hours need to be compensated, and so on. On week $21(2n+2)$ we use 21 hours to compensate, which is the maximum allowed, given that each week requires a 24 hour rest period. As before, the last $24 \mid 45 \mid 24$ block cannot be used to compensate, but can be compensated with the following unlimited rest.

More formally, every week w numbered $2k$ (including week zero) will be reduced and compensated by week $2k+2$, up to and including week $21(2n+2)$. The amount of the compensation is the unique $i > 0$ such that $(i-1)(2n+2) \leq w < i(2n+2)$.

As in $\overline{\mathcal{B}_n}$, the $24 \mid 45 \mid 24$ block at the end of $\overline{\mathcal{C}}_n$ cannot be used to compensate previous weeks (see the proof of Lemma 10.6). There are $m := 22(2n+2)+1$ remaining weeks in $\overline{\mathcal{C}}_n$, of which $22(2n+1)$ have 45 resting hours, 22 have 44 resting hours, and 1 has 66 resting hours, for a total of $45m-1$ resting hours. Thus there are not enough resting hours to distribute among the weeks. $\square$

Lemma 10.11. There is a stratified $168n$-U-bisimulation $\vec{Z}$ between $\mathcal{C}_n$ and $\overline{\mathcal{C}}_n$ such that $0\ Z_{20}\ 0$.

Proof. Define $r := 2n+2$ and for $168w + h \in \mathcal{C}_n$ and $168v + \ell \in \overline{\mathcal{C}}_n$, set $168w + h\ Z_i\ 168v + \ell$ if $h = \ell$ and one of the following properties holds: (C1) $\max\{w + r, v\} < (21 - i)r$ and $w \equiv v \pmod{r}$, or (C2) $v = w + r$. We

need to show that $\vec{Z}$ is a stratified bisimulation. To see this, assume that $x\ Z_i\ y$ and write $x = 168w + h$, $y = 168v + \ell$. Note that we must have $h = \ell$. If $i = 0$ then some inspection shows that x and y share the same formulas of the form $\bigcirc^j p$ with $j \leq 168n$, as the current and subsequent n weeks are of the same form. It is sufficient to check this for Z_0 because it contains all the Z_i. If $i > 0$, change variables so that $x \sim_{i+1} y$; we check that the required clauses hold.

FORTH U. Let $x' \geq x$ and write $x' = 168w' + h'$. Consider two cases. First assume that $w' \leq w + r$. Set $y' = y + (x' - x)$ and for $z \in [y, y']$ set $\xi(z) = x + (z - y)$. It is then not hard to check that if $x\ Z_{i+1}\ y$ by (C1) then $\xi(z)\ Z_i\ z$ by (C1), and similarly if $x\ Z_{i+1}\ y$ by (C2) then $\xi(z)\ Z_i\ z$ by (C2). The other requirements on ξ are easy to check, so that ξ witnesses FORTH U.

Otherwise $w' > w + r$. We claim that there is v' such that $168v' + h' \geq 168v + h$ and $168w' + h'\ Z_i\ 168v' + h'$. If $168(w' + r) + h' \geq 168v + h$ we may take $v' = w' + r$. Otherwise, we have $v \geq w' + r > w + r$ so that x, y do not satisfy (C2) and thus $\max\{w + r, v\} \leq (21 - i - 1)r$. Take $v' \in (v, v + r]$ with $v' \equiv w' \pmod{r}$ and set $y' = 168v' + h'$; from $(21 - i - 1)r \geq v \geq w' + r$ and $v' \leq v + r \leq (21 - i)r$ we obtain $x'\ Z_i\ y'$ by (C1).

We now construct the function $\xi\colon [y, y'] \to [x, x']$. First define $\xi(y') = x'$. For $z = 168u + t \in [y, y')$, we consider two cases. If $168(u - r) + t \in \big[x, x'\big)$ take $\xi(z) = 168(u - r) + t$, which in view of (C2) satisfies all desired properties. Otherwise, $168(u - r) + t \notin [x, x')$, and choose $d \in (0, r]$ such that $w + d \equiv u \pmod{r}$, then set $\xi(z) = 168(w + d) + t$. The assumption that $w' > w + r$ yields $\xi(z) \in [x, x']$. It remains to show that $\xi(z)\ Z_i\ z$, for which it suffices to check that $\max\{w + d + r, u\} < (21 - i)r$.

If $168(u - r) + t < x$ then since $z \geq y$, either $u > v$ and hence $v < u \leq w + r$, or else $u = v$ and $t \geq h$, so that forcibly $u - r < w$ and thus $v < w + r$. But $v < w + r$ together with $168v + h'\ Z_{i+1}\ 168w + h$ means that (C1) holds so that $\max\{w + r, v\} < (21 - i - 1)r$. Thus we have $u - r \leq w < (21 - i - 1)r$ so that $u < (21 - i)r$. Similarly $w + d \leq w + r < (21 - i - 1)r$ yields $w + d + r < (21 - i)r$.

Otherwise $168(u - r) + t \geq x'$. But then since $z < y'$, either $u = v'$ and hence $t < h'$, so that $v' - r = u - r > w'$; or else $u < v'$ and $v' - r > u - r \geq w'$. Thus $w' + r \neq v'$, which together with $168w' + h'\ Z_i\ 168v' + h'$ yields $\max\{w' + r, v'\} < (21 - i)r$. From $u \leq v' < (21 - i)r$ and $w + d + r \leq (w + r) + r < w' + r < (21 - i)r$ we obtain $\max\{w + d + r, u\} < (21 - i)r$,

as needed.

BACK U. This is essentially symmetric and we omit it. □

Theorem 10.12. All $\mathcal{L}_{\mathsf{U}}$ formulas equivalent to $\psi_{\S 8.6}$ have U-depth at least 20.

Proof. Suppose that $\psi \in \mathcal{L}_{\mathsf{U}}$ is a formula expressing Article §8.6 with $\bigcirc$-depth d and U-depth less than 20. Choose n big enough to ensure that $d \leq n$, and let $\vec{Z}$ be the bisimulation of Lemma 10.11. Then by Lemma 9.3, $(\mathcal{C}_n, 0) \models \psi \iff (\overline{\mathcal{C}}_n, 0) \models \psi$. This contradicts Lemma 10.10. □

Remark 10.13. One can ask how Theorem 10.12 would differ if we included 'since' in the language. In this case, $(\mathcal{C}_n, 0)$ and $(\overline{\mathcal{C}}_n, 0)$ are only about 10-bisimilar. However, the nesting depth of 20 is determined only by the resolution of our models. If instead we used a minute-wise resolution (which, as we have mentioned, is the resolution required by the law itself), we could stretch this to 20×60 by replacing the 44 hour reduced weekly rests by 44:59h reduced weekly rests. Thus any LTL definition of $\psi_{\S 8.6}$ would have to exploit the temporal resolution in an essential way, making it arguably unnatural.

11. Concluding remarks

We have analysed the interdependence between Requirements (51) and (52) of Regulation (EU) 2016/799, and shown that for any two different time standards, there exists a labelling of second activities such that under one time standard all minutes would be labelled as driving minutes while under the other labelling they would all be labelled as resting minutes. This is not only a theoretical issue: by analysing real-world data we have checked that differences in driving time may indeed amount up to 8%.

We have also shown that the Σ^1_1 fragment of monadic second order logic is sufficient for formalizing even the most problematic passages we have found in our study of European transport regulations. The upshot is that evaluating whether a given truck driver's record complies with regulations can then be transformed into a model-checking problem over this fragment. Moreover, truth of Σ^1_1 MSO formulas is equivalent to validity for MFO, and via Kamp's theorem we may further reduce it to validity of LTL formulas, for which many solvers are already available. Nevertheless, validity in LTL is PSPACE-complete, and moreover the translation of MFO into LTL is non-elementary in the worst case, so this approach is not ideal from a complexity perspective.

On the other hand, LTL is suitable for representing portions of content of the regulation, and the model-checking problem (over deterministic models) is polynomial [15]. In fact, the advantage of having such a general tool available can be viewed as an argument to use 'sugared' versions of LTL (say, with counting modalities) in the design of future and revision of current laws.

Indeed, we can consider a variant of §8.6 where the requirements are: (i) in every two consecutive weeks, the driver must take two weekly rest periods, at least one of which is regular, and (ii) in every four consecutive weeks, the sum of the weekly rest periods must be of at least 180 hours. This version of the article can be easily checked to be definable by a not-too-large LTL formula and maintain the spirit of the original, as drivers are required to compensate reduced rest periods within the following three weeks.

References

[1] D. Fernández-Duque, M. González Bedmar, D. Sousa, J. J. Joosten, G. Errezil Alberdi, To drive or not to drive: A formal analysis of Requirements (51) and (52) from Regulation (EU) 2016/799, in: Edición Especial: Personalidades jurídicas difusas y artificiales, Vol. 4 of TransJus Working Papers Publication, Institut de Recerca TransJus, 2019, pp. 159–171.

[2] A. de Almeida Borges, J. J. Conejero Rodríguez, D. Fernández-Duque, M. González Bedmar, J. J. Joosten, The second order traffic fine: Temporal reasoning in european transport regulations, in: J. Gamper, S. Pinchinat, G. Sciavicco (Eds.), 26th International Symposium on Temporal Representation and Reasoning, TIME 2019, October 16-19, 2019, Málaga, Spain, Vol. 147 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 6:1–6:16.

[3] G. Governatori, Thou shalt is not you will, in: Proceedings of the 15th International Conference on Artificial Intelligence and Law, ICAIL 2015, San Diego, CA, USA, June 8-12, 2015, 2015, pp. 63–68.

[4] H. Herrestad, Norms and formalization, in: Proceedings of the Third International Conference on Artificial Intelligence and Law, ICAIL '91, 1991, pp. 175–184.

[5] European Parliament, Council of the European Union, Commission implementing Regulation (EU) 2016/799 of 18 March 2016 implementing

Regulation (EU) No 165/2014 of the European Parliament and of the Council laying down the requirements for the construction, testing, installation, operation and repair of tachographs and their components, Official Journal of the European Union (2016).

[6] European Parliament, Council of the European Union, Regulation (EC) No 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport, Official Journal of the European Union (2006).

[7] European Parliament, Council of the European Union, Commission Regulation (EU) No 1266/2009 of 16 December 2009 adapting for the tenth time to technical progress Council Regulation (EEC) No 3821/85 on recording equipment in road transport, Official Journal of the European Union (2009).

[8] J. del Castillo Tierz, When the laws of logic meet the logic of laws, Master's thesis, University of Barcelona, Barcelona (2018).

[9] S. C. Tosatto, G. Governatori, P. Kelsen, Business process regulatory compliance is hard, IEEE Trans. Serv. Comput. 8 (6) (2015) 958–970.

[10] M. Frick, M. Grohe, The complexity of first-order and monadic second-order logic revisited, Ann. Pure Appl. Log. 130 (1-3) (2004) 3–31.

[11] J. Ouaknine, J. Worrell, On the decidability and complexity of metric temporal logic over finite words, Log. Methods Comput. Sci. 3 (1) (2007).

[12] L. Aceto, D. D. Monica, V. Goranko, A. Ingólfsdóttir, A. Montanari, G. Sciavicco, A complete classification of the expressiveness of interval logics of allen's relations: the general and the dense cases, Acta Informatica 53 (3) (2016) 207–246.

[13] H. Kamp, Tense logic and the theory of linear order, Ph.D. thesis, UCLA (1968).

[14] N. Kurtonina, M. de Rijke, Bisimulations for temporal logic, Journal of Logic, Language and Information 6 (4) (1997) 403–425.

[15] D. Harel, J. Tiuryn, D. Kozen, Dynamic Logic, MIT Press, Cambridge, MA, USA, 2000.