Article

Not peer-reviewed version

An Empirical Evaluation of a Coding Legislation Methodology

Hanif Bhuiyan , Guido Governatori * , Alida Binte Saqi

Posted Date: 28 April 2025

doi: 10.20944/preprints202504.2295.v1

Keywords: Legal coding; Explainable AI; Defeasible Deontic Logic

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2025 doi:10.20944/preprints202504.2295.v1

Article

An Empirical Evaluation of a Coding Legislation Methodology

Hanif Bhuiyan [1], Guido Governatori [2,3,*] and Alida Binte Saqi [4]

[1] City of Gold Coast, Queensland, Australia
[2] School of Engineering and Technology, Central Queensland University
[3] Artificial Intelligence and Cyber Futures Institute, Charles Sturt University
[4] United Nations Development Programme, Bangladesh
* Correspondence: g.governatori@cqu.edu.au

Abstract: We propose a novel methodology to encode legislation in a formal language (specifically Defeasible Deontic Logic). After the presentation of the methodology, we report on the results of an empirical evaluation and compare them with the outcome of a previous experiment. The comparison suggests a significant improvement over the previous experiment.

Keywords: Legal coding; Explainable AI; Defeasible Deontic Logic

1. Introduction

Rules as Code (RaC) is an innovative approach to drafting legislation, regulations, and policies. Instead of writing them in natural language, RaC encodes them into machine-readable formats. This method aims to improve the clarity, transparency, and efficiency of legal rules by making them understandable and executable by computers. It has been argued that Rules as Code provides several benefits, including [1,2]:

- Reduced Ambiguity: Encoding rules in a precise, machine-readable format helps identify and eliminate ambiguities present in natural language.
- Improved Compliance: Easier interpretation of rules can lead to better compliance by citizens and organizations.
- Enhanced Transparency: Machine-readable rules provide greater transparency, making it easier to understand and follow the regulations.
- Efficiency in Service Delivery: Automating the interpretation of rules can streamline public service delivery and reduce administrative burden.

However, encoding legal provision in a machine-understandable and executable language is a form of legal interpretation [3–5]. In addition, encoding a piece of legislation is a time-consuming activity (anecdotally, it has been reported that a proficient coder can encode four to five pages of legal text per day; moreover, the burnout rate is very high). A possible solution is to use a team of coders working simultaneously on different parts of the legal text to encode. However, this requires a high degree of alignment between the coders. Previous experiences with large-scale coding projects indicate that this high degree of alignment is very hard to achieve (see [3] for an empirical experiment on this matter), and proper encoding methodologies and protocols are needed [3].

In this paper, we propose a logic-based methodology to encode legal provisions. We evaluate the proposed approach with the help of an empirical experiment, and we compare the results with the outcome of a similar experiment [3]. To the best of our knowledge, [3] was the first and only experiment to empirically evaluate the alignment of legal coding when done by different coders.

The paper is organised as follows: Section 2 presents the methodology to encode legal provisions in Defeasible Deontic Logic, including the protocols to identify the atomic propositions (Section 2.1), the deontic components (Section 2.2), the logical rules (Section 2.3), and a quick overview of the underlying logic (Section 2.4). The experiment is described in Section 3. Section 4 provides a comparative analysis

 doi:10.20944/preprints202504.2295.v1

with the results of the experiment carried out in [3]. Finally, Section 5 concludes the paper with some final remarks.

2. Methodology

We introduce a Defeasible Deontic Logic (DDL) based encoding methodology to convert copyright rules into the machine-executable format.

This proposed methodology works in four steps (Figure 1):

1. Define atoms,
2. Identify deontic modalities,
3. Define the if-then structure and
4. Rule encoding using DDL.

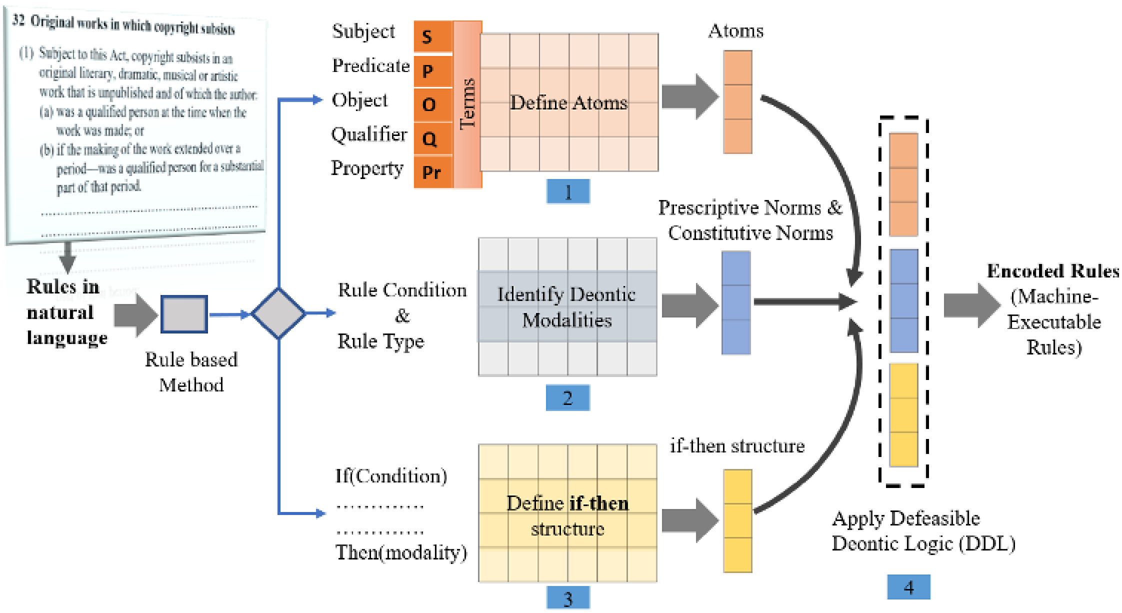

Figure 1. Rule Encoding Methodology.

These steps are done manually. The methodology's input is a set of copyright rules in natural language. In the first step, atoms are defined by the rules. In the second step, norms are determined. Then, the if-then structure is identified from the rules in the third and final step. Finally, DDL is applied to the atoms, deontic modalities, and the if-then structure to make the machine-executable (M/E) rules format. The steps are explained below.

2.1. Define Atoms

This section briefly outlines how atoms are defined by rules. An atom is a predicate symbol including constants or variables that contain no logical connectives [6]. Here, the atoms are extracted based on the occurrences of terms/expressions in the sentences or the textual provision of the rules. A *term* is a *variable* or an (individual) *constant* in the textual provision [7]. This work deals with expressions (predicates, variables, and constants) that refer to subject (s), predicate (p), property (pr), object (o), and qualifier (q) in a rule sentence.

In natural language, a subject (or entity) refers to the term about which something is said in the sentence. The something which is said about something is the predicate of the sentence. The predicate of a subject-predicate sentence indicates a relation or a property. The object is what the subject does something to. In other words, the object is the result of the action. Qualifiers are terms that usually enhance or limit another word's meaning. Qualifiers can refer to either the subject, the predicate, or any of the objects in a sentence.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2025 doi:10.20944/preprints202504.2295.v1

Before generating terms, some article pre-processing is done on the text. For example, verbs (auxiliary, principal, modal, etc.) are not considered terms. In logic, subjects and objects are variable or constant in the rule sentence correspondence to the entity [7]. A predicate is a constant in the rule sentence that always refers to the properties or actions of entities. Properties indicate the relationship between a subject and a predicate. Object refers to the properties of the entities. Qualifiers refer to the variables that enhance or limit entities.

An atom is a combination of these terms/expressions that form a (primitive) Boolean expression. For example, "Copyright subsists in a work". According to the linguistic perspective, the term "copyright" is the subject of the sentence. The term "work" is the object of the sentence as the subject is doing something to it. The term "subsists (in)" is the predicate of the sentence as it expresses the relation between the subject (copyright) and the object (work). In logical terms, the sentence can be rendered as a binary predicate with the form

$$Predicate(Subject, Object);$$

hence, for the specific sentence we have

$$Subsist(Copyright, Work).$$

Consider now the sentence "copyright is the exclusive right to reproduce the work in material form". In this more complex expression, the subject is still "copyright", and "(is the) exclusive right" is a property of the subject. The predicate is "reproduce", the object is "work". The term "material form" is the qualifier of the sentence, as this enhances the subject meaning through the object. In the logical approach, "copyright" is that variable that refers to the entity (subject) of the sentence. "reproduce" is the predicate constant that refers to the action of the entity. "work" is an (individual) constant of the sentence which is referring to the properties of the entity. "material" is the variable (qualifier) which is referring to the enhancement of the entity. To convert the sentence to its logical form, some further analysis is needed. Specifically, we have to fill some gaps. Who has the exclusive right, and who is going to perform the act covered by copyright? Hence, we can rephrase the sentence as "the copyright owner has the exclusive right to reproduce the work in material form". Thus, we have

1. "copyright subsists in the work";
2. "a person owns the copyright on the work";
3. "the person is permitted to reproduce the work in material form" or "it is permitted that the person reproduces the work in material form";
4. "no body else is permitted to reproduce the work in material form".

When we examine the sentences in 3 we notice that there is a modal (permitted), and the Boolean sentence "the person reproduces the work in material form" is in the scope of the deontic operator. When we analyse the sentence, logically, it can be represented as a predicate in one of the following two forms:

$$Predicate(Subject, Qualifier, Object)$$
$$PredicateQualifier(Subject, Object).$$

thus, we can model it as either

$$Reproduce(Person, MaterialForm, Work)$$

or

$$ReproduceMaterialForm(Person, Work).$$

In general, a rule uses natural language to define the cases (events and facts) they are meant to regulate (terms, conditions, and legal provisions). Depending on the events, the description of these

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2025 doi:10.20944/preprints202504.2295.v1

cases varies. There is no general structure in how these rules are written. Due to this heterogeneity of the rule information, the atom structure varies. Based on extensive analysis and encoding of various regulations, we noticed that the following five patterns suffice to represent the vast majority of provisions.

- S-P-O: Subject-Predicate-Object;
- S-P-Q-O: Subject-Predicate-Qualifier-Object;
- S-Pr: Subject-Property,
- S-P-O-O: Subject-Predicate-Object-Object;
- S-Q-P-O: Subject-Qualifier-Predicate-Object.

Based on these combinations, some examples of atoms from the Australian Copyright Act and the patterns used to obtain them are shown below.

For the representation of the atoms, there are some possible conventions. One is to use the convention typically used in logic

$$Predicate(argument_1, \ldots, argument_n)$$

Where the arguments are the subject, objects, and qualifiers. This would be the best option for a first-order based formalisation. The second convention is to create atoms by combining the terms in the order in which they appear (this is the favourite option for a propositional based formalisation). We adopted the second option, but we inverted the order for the S-Pr pattern.

Example 1 (S-P-O)**.** Part III, Division 1, 32: "copyright subsists in the work"

Subject	Predicate	Object
copyright	subsist in	work

Defined Atom: `Copyrigth_Subsist_Work`

As we discussed above, this is a simple instance of the Subject-Predicate-Object pattern

Example 2 (S-P-Q-O)**.** Part III, Division 1, 32, 2(d): "the author was a qualified person".

Subject	Predicate	Qualifier	Object
author	was	qualified	person

Defined Atom: `Author_Was_Qualified_Person`

In this case, we recognise that the term "qualified" is the qualifier of the object ("person").

Example 3 (S-Pr)**.** Part I (Interpretation), 10, "work means literary [...] work"

Subject	Property
work	literary

Defined Atom: `Literary_Work`.

A property can be seen as a special qualification of a subject, and is often signalled by an auxiliary verb ("be" or "have"), and it corresponds to a unary predicate. In this case, we have the verb "mean"; this can also be understood as "a literary work is a work".

Notice that the sentence in Example 2 can be analysed as:

Subject	Property
Author	Qualified Person

Defined Atom: `Qualified_Person_Author`.

Example 4 (S-P-O-O)**.** "a person has authorised the reproduction of a work"

 doi:10.20944/preprints202504.2295.v1

Subject	Predicate	Object	Object
person	authorise	reproduction	work

Defined Atom: `Person_Authorise_Reproduction_Work`

With this interpretation, we analysed the sentence with two objects: the first "reproduction" and the second "work". An alternative reading would be

Subject	Predicate	Qualifier	Object
person	authorise	reproduction	work

Defined Atom: `Person_Authorise_Reproduction_Work`

With the alternative parsing the term "reproduction" is taken as the qualifier for the term "work". However, the two alternative readings produce the same atom.

Example 5 (S-Q-P-O)**.** "the first publication of the work took place in Australia".

For this sentence, we have two options. The first option is to examine the sentence as it appears. Here, the subject is "first publication", where "first" is the qualifier of "publication".

Subject	Qualifier	Predicate	Object
publication	first	took place	Australia

Defined Atom: `First_Publication_TookPlace_Australia`

For the second option, we consider the passive form, where the subject is "work". While semantically the two expressions share the same meaning, the focus in this case is to stress that the publication is a property of a work.

Subject	Qualifier	Predicate	Object
work	first	published	Australia

Defined Atom: `WorK_First_Published_Australia`

Based on previous experiments and encodings, we believe that these patterns suffice to cover the majority of cases under any regulation. In some cases, complex sentences can be split using multiple patterns, as we did in the case of the complex sentence "copyright is the exclusive right to reproduce the work in material form". The proposed methodology aims to improve encoding efforts' uniformity, consistency and repeatability. [3] reports very high (syntactic) variability of encoding when a team of coders does them; moreover, they report that adopting a common naming convention and sharing an encoding methodology greatly increases the agreement among the encodings by the different coders. Also, fitting textual provisions in the patterns allows us to identify expressions that could have different syntactic structures but the same semantic meaning; the same atoms will formalise such expressions.

2.2. Identify Deontic Modalities

Deontic modalities are expressions in the rule that (legally) qualify terms and actions. Deontic modalities help us determine the types of norms we are going to encode. Each norm is represented by one or more constitutive or prescriptive rules (see [8]). Constitutive rules define terms specific to legal documents. Prescriptive rules prescribe the "mode" of the behaviour using deontic modalities: obligation, permission, and prohibition. Here, we follow the definition given by LegalRuleML for obligation, prohibition, and permission (see [9]). An obligation is an action or course that the subject must perform or achieve, whereas a prohibition is an action or course that the subject must not perform or avoid. Permission is the state of an action that is not subject to a prohibition, or the opposite is not obligatory.

This research identifies norms based on both constitutive and prescriptive forms of rules. For example (Figure 2), in the Australia Copyright Act 1968, part III, Division 1, 31 (1) states that "copyright, in relation to a work, is the exclusive right to reproduce the work in a material form [. . .]”. This law gives the permission (right) to the owner of the copyright to reproduce the work (and in general to perform any of the acts described in subsection (a)). Notice that what we call “prescriptive rules” include prescriptive rules in the strict sense (rules establishing an obligation or a prohibition) as well as permissive rules (rules whose effect is a permission). In this case, the modality of the prescriptive rule is “permission”[1]; also, in the specific case, the provision specifies that the right is exclusive.

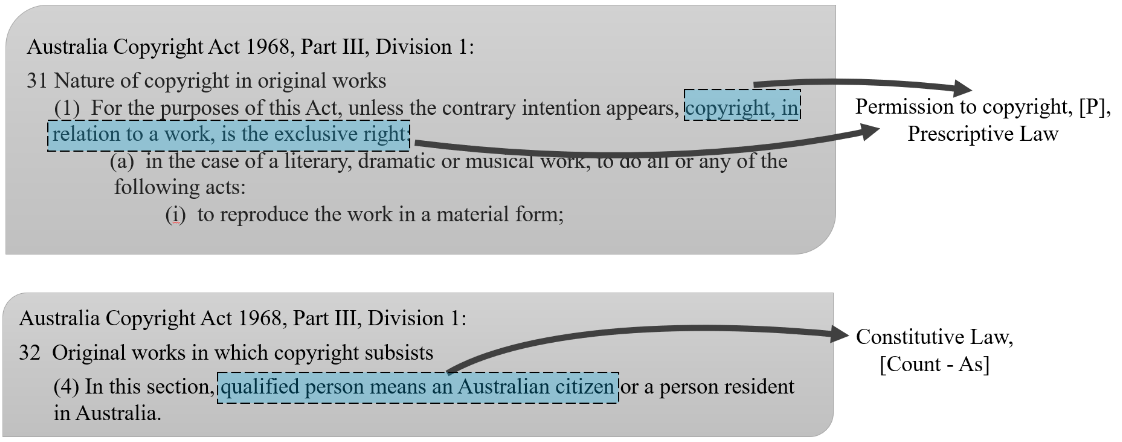

Figure 2. Identifying Deontic Modalities.

Another example, Australia Copyright Act 1968, part III, Division 1, 32 (4) states that “In this section, qualified person means an Australian Citizen . . . ”. Here, this phrase is meant to define the qualified person, which is a constitutive norm (Count-As) within this statement.

2.3. Define the If-Then Structure

Rules specify the actions of the subject (or the conditions the subject must ensure to hold). They consist of deontic modalities and conditions that control the subject’s behaviour. A rule comprises two parts: if (antecedent or premise) and then (consequent or conclusion). Therefore, a rule can be represented as:

If (antecedent)
Then (consequent)

If the premise becomes true, then the consequent part of the rule triggers. A rule may have multiple antecedents joined by logical operators: OR and AND. From a legal perspective, rules use conditions on some actions to achieve/mandate specific behaviours. Therefore, the if-then structure is identified from rules using atoms and deontic modalities (norms) for encoding. For example, let us consider again the Australia Copyright Act 1968, part III, Division 1, 31. This section recites:

31 Nature of copyright in original works

(1) For the purposes of this Act, unless the contrary intention appears, copyright, in relation to a work, is the exclusive right:

(a) in the case of a literary, dramatic or musical work, to do all or any of the following acts:

(i) to reproduce the work in a material form;

The provision contains an explicit permission for the owner of the copyright to do all or any of the acts specified in the rest of the section. At the same time, it forbids the doing to persons who are not the copyright owner. In addition, there is an exception in case the “contrary intention appears”.

[1] A “right” is in general, a permission on a party implying an obligation or prohibition on other subjects.

```
r_{31.1.a.i}
IF
   AND
      Copyright_Subsist_Work
      Person_Own_Copyright
      OR
        Literary_Work
        Dramatic_Work
        Musical_Work
THEN [P]
    Person_Reproduce_Work_MaterialForm
--
r_{31.1.a.i.F}
IF
  AND
    Copyright_Subsist_Work
    NEG Person_Own_Copyright
    OR
        Literary_Work
        Dramatic_Work
        Musical_Work
THEN [O]
    NEG Person_Reproduce_Work_MaterialForm
--
r_{31.1}
IF
    OppositeIntention_Appear_Copyright
THEN (count-as)
    NEG Copyright_Subsist_Work
```

2.4. Rule Encoding

After defining and identifying atoms, deontic modalities, and if-then structures for rules, the expressions are converted into a Defeasible Deontic Logic (DDL) [10]. DDL is an extension of Defeasible Logic (DL) [11] with Deontic Operators and compensatory obligation operators introduced [12]. DDL is a formalism that provides a conceptual approach to encoding the norms and simultaneously exhibits a computationally feasible environment to reason about them. DDL has been successfully used in legal reasoning to handle norms and exceptions, and it does not undergo problems affecting other logics used for reasoning about compliance and norms [10,13–15]. Efficient implementations of the logic have been proposed [16–18] that have been used for large scale coding project in various legal domains, including: Australian Spent Conviction [19], Traffic Rules [20], Building Code [21], Consumer Data Right [5] and more. Below is a brief overview of Defeasible Logic and Deontic modalities.

Defeasible Logic (DL) is a non-monotonic, sceptical logic that prevents the derivation of contradictory conclusions. For example, suppose there is a piece of information that supports the conclusion A, but there is also a second piece of information that supports not A, thus preventing the conclusion of A. DL recognises the opposite conclusions and does not derive them. However, if A's support has priority over $\neg A$, then it might be possible to conclude A. DDL extends DL with deontic operators (obligation, O and permission P), classifies rules in constitutive rules and prescriptive rules, and introduces the compensation operator of [12].

Defeasible Deontic Logic comprises five separate knowledge foundations: strict rules, facts, defeasible rules, defeaters (see below), and superiority relations [10,11]. Knowledge is organised in

doi:10.20944/preprints202504.2295.v1

theories, where a theory D is a triple $(F, R, >)$ where F is a set of facts, R is a set of rules, and $>$ is a superiority relation in R.

Expressions in Defeasible Logic are built from a finite set of literals, where a literal can be either an atomic statement or its negation, or a deontic literal. A deontic literal is a literal inside the scope of a deontic operator. Given a literal A, $\sim A$ denotes its complement. That is, if $A = B$, then $\sim A = \neg B$, and if $A = \neg B$, then $\sim A = B$.

Facts (F) are unequivocal and conclusive statements. A fact represents a state of affairs (literal) or an act that has been performed that is believed to be true.

A *rule* (an element of R) specifies the relationship between premises and a conclusion and can be characterised as its strength. Strict rules, defeasible rules, and defeaters can be distinguished based on the relationship strength of the rules [14]. The following expressions describe these rules:

- $A_1, \ldots, A_n \rightarrow_X B$ (Strict Rules),
- $A_1, \ldots, A_n \Rightarrow_X B$ (Defeasible Rules) and
- $A_1, \ldots, A_n \rightsquigarrow_X B$ (Defeaters)

where $A_1, \ldots, A_n$ is the antecedent or premises (clauses), and B is the rule's consequent or conclusion (effect). Moreover, X denotes the mode of the rule; if $X = \mathsf{C}$, the rule is a constitutive and if $X = \mathsf{O}$ or $X = \mathsf{P}$, then the rule is a prescriptive rule. Prescriptive rules include prescriptive rules in the strict sense, where the conclusion of the rule is an obligation, and permissive rules, where the conclusion is a permission.

Strict rules are rules in the classical sense: if the premises are unarguable (for example, a fact), then so is the conclusion. For example, "a literary work is a work", formally:

$$\mathit{LiteraryWork} \rightarrow_{\mathsf{C}} \mathit{Work}.$$

This is a strict constitutive rule.

Defeasible rules are rules that can be defeated by contrary evidence. For example, "copyright subsists in a work unless the opposite intention appears" can formally be written as:

$$\mathit{OppositeIntentionApperCopyrigth} \Rightarrow_{\mathsf{C}} \neg \mathit{CopyrightSubsitWork}.$$

Defeaters are rules that cannot be used to derive any conclusions on their own. Their purpose is to preclude some conclusions, i.e., to undermine some defeasible rules by supplying opposite evidence.

Suppose we have two rules for opposite conclusions. In this scenario, we cannot establish any conclusion unless we prioritise the rule. The superiority relation ($>$) is the DDL mechanism to define the priorities among the rules. For example, based on the following defeasible rules:

$$r_{31.1.a.1.F} : \mathit{CopyrightSubsistWork}, \neg \mathit{PersonOwnCopyright}, \mathit{LiteraryWork} \Rightarrow_{\mathsf{O}} \neg \mathit{PersonReproduceWorkMaterialForm}$$
$$r_{40.1} : \mathit{PersonFairDealingWork}, \mathit{LiteraryWork}, \mathit{PersonUseWorkResearch} \Rightarrow_{\mathsf{P}} \mathit{PersonReproduceWorkMaterialForm}$$

No conclusive decision can be made about whether the reproduction infringed the copyright. However, if we establish a superiority relation with $r_{40.1} > r_{31.1.a.1.F}$, then we can state that the fair use of literary work for the purpose of research is permitted (and hence it is not an infringement on the copyright).

The reasoning mechanism of Defeasible Deontic Logic has a three-step argumentation structure:

1. propose an argument (rule) for the conclusion to prove;
2. identify all possible counter-arguments (rules for the opposite);
3. rebut the counter-arguments:
    (a) undermine the counter-argument (show that some of the premises of the counter-argument do not hold);
    (b) defeat the counter-argument (show that the counter-argument is weaker than an argument supporting the conclusion).

 doi:10.20944/preprints202504.2295.v1

A complete definition of the defeasible deontic logic reasoning mechanism can be found in [10,13–15].

3. Experiment

A comprehensive experiment has been conducted to evaluate the proposed encoding methodology. Although the proposed encoding methodology is designed as a generic approach, it is not feasible to experiment with several rules. The Australian Copyright Act 1968 was considered to evaluate this methodology's effectiveness. This is the first experiment of this kind where two groups (4 participants in each group) of participants use the proposed methodology to encode some rules of the Copyright Act. In this copyright rule, we considered Part III, Division 1 (sections 31, 32), Division 2 (section 36), Division 3 (sections 40, 41, 41A, 42), and Division 4 (section 46). Each group of participants encoded these provisions based on the proposed methodology. First, they made atoms from these norms and then encoded rules (machine-executable rules). Then, we compared their outcomes.

There are several reasons for the choice of the Australian Copyright Act 1968 in our experiment. First, while there are differences in copyright regulations across jurisdictions, the general principles are common among the jurisdictions; second, copyright rule is a well-understood sector in the legal domain, with ample jurisprudence, and established interpretations of terms and common situations: this reduces the difference in encodings due to disparate interpretations of the norms. Finally, the same sections of the Act were used in a previous experiment [3] where the encodings from separate coders were compared: This provides us with a baseline to evaluate the success of the proposed encoded methodology.

Our hypothesis is that the methodology will be effective if both groups can define the atoms and do the encoding based on the proposed methodology. If the encoding similarity of these two groups becomes reasonable, then it can be stated that the proposed methodology is effective.

3.1. Participants and Experiment Procedures

Participants were recruited from the Rule Department of the University of Asia Pacific, Dhaka, Bangladesh. A total of 8 participants were recruited, irrespective of gender. Then, randomly, two groups (four in each group) were formed among them. The reason to recruit them is that rule students understand rules much better than the general public. The research team conducted several training sessions to teach and train participants to understand Defeasible Deontic Logic (DDL) as well as the proposed encoding methodology.

The experiment was conducted in two steps. First, after the training session, with the monitoring facility from the research team, participants were asked to encode Part III, Division 1 (rule 31, 32), and Division 2 (rule 36) of the copyright rule. Afterwards, participants were asked to encode rules 40, 41, 41A, 42 (Division 3), and 46 (Division 4) independently.

Discussion between the two groups was prevented by asking each group's representative to avoid discussion among themselves. To recognise their voluntary involvement in this research, each participant was compensated (AUD$70).

3.2. Results

The effectiveness of the proposed encoding methodology is evaluated based on the percentage of similarity between the two groups' encodings. The experiment was conducted in four ways:

1. Defining atoms from rules with monitoring
2. Defining atoms from rules individually
3. Encode rules with monitoring
4. Encode rules individually

After the experiment, the evaluation was conducted to check the similarity between these two groups while defining atoms and encoding. The evaluation was conducted in two aspects:

1. without semantic mapping,
2. with semantic mapping.

By semantic mapping, we mean that we manually examined the atoms and the rules created by the two teams during the experiment, and we matched syntactically different atoms/rules with the same semantic meaning. As we discussed, legal coding is a matter of interpretation, and sometimes a provision, and the terms in the provision can be analysed in different ways. For instance, the sentence reported in Example 5 can be parsed in different ways. However, there is no difference in the meaning of the resulting atoms.

Table 1 reports the outcome of the experiment, and shows that, with training and monitoring, for rules 31, 32, and 36, the degree of similarity between the two groups for defining an atom is 88.9%. Semantic mapping increased the degree of similarity by around 98%. A similar trend is shown for rule encoding. Without semantic mapping, the degree of similarity between the two groups is 86.7%, and with semantic mapping, it is 97.53%.

Table 1. Results with and without semantic mapping.

Independent Encoding		
	Without Semantic	**With Semantic**
	Atoms	Atoms
Sections 40, 41, 41A, 42, 46	Total = 79 Matched = 37 Percentage = 46.83%	Total = 79 Matched = 46 Percentage = 58.22%
	Encoding	Encoding
	Total = 72 Matched = 13 Percentage = 18.05%	Total = 72 Matched = 27 Percentage = 37.5%

Encoding with Monitoring		
	Without Semantic	**With Semantic**
	Atom Define	Atom Define
Sections 31, 32, 36	Total = 81 Matched = 72 Percentage = 88.9%	Total = 81 Matched = 79 Percentage = 97.53%
	Encoding	Encoding
	Total = 75 Matched = 65 Percentage = 86.7%	Total = 75 Matched = 73 Percentage = 97.33%

In the case of independent tasks by participants, for rules 40, 41, 41A, 42, and 46, without semantic aspect, the degree of similarity between the two groups for defining atoms is 46.83%. In contrast, with semantic mapping, the agreement rate becomes 58.22%. For the rule encoding, the agreement percentage between the two groups is more than double in terms of semantic mapping (37.5%) compared to without semantic mapping (18.05%).

3.3. Result Analysis

Table 1 shows the proposed encoding methodology's effectiveness based on the similarity between two groups of participants. There is a significant difference seen in terms of the degree of similarity between encoding with monitoring and fully independent encoding. For both defining atom and rule encoding, the similarity rate between the two groups is more than 85% with monitoring. While during independent work, the similarity rate is, on average, below 50%. The highest similarity rate is seen with monitoring, which are 97.53% and 97.33% for defining atom and rule encoding, respectively. The lowest similarity is seen for independent work, which is 46.83% and 18.05% for defining atom and rule encoding, respectively.

After analysing the participant task, it is seen that some participants overlooked critical issues in the rules. For example, they could not define the atom properly by identifying terms, norms, and the if-then structure of the rules. For some rules, participants could not understand the structure, such as the consequences and the conditions. Furthermore, during the monitoring, it is noticed that some participants struggle to understand the rule's subtle meaning. Some participants strongly understand the rules, while some have less understanding. Moreover, overthinking, lack of attention, and misunderstanding of rules were also noticed among the participants while working on this project. Furthermore, while the participants were sufficiently proficient in the English language, they were not native speakers.

The low degree of similarity between participants may not have happened if it had been possible to examine participants' knowledge level about rules before involving them in work. However,

 doi:10.20944/preprints202504.2295.v1

recruiting participants is a challenging task. It is difficult to find participants through interviews (examining).

4. Comparative Analysis

To conduct a comparative analysis of the proposed encoding methodology performance, we undertook a comparative analysis between our proposed methodology and a previous approach by [3]. Both methodologies were employed to encode specific sections of the Australian Copyright Act Part III, encompassing rules 31, 32, 36, 40, 41, 41A, 42, and additionally, rules 33 was considered in our case. Table 2 presents the comparison between the two encoding methodologies.

Both approaches involved encoding conducted in two steps. In the first step, participants defined atoms and encoded rules 40, 41, 42, and 42A independently, without semantic mapping consideration. In our approach, participants additionally encoded rule 46. This initial step is denoted as "Week 1" in [3] approach and "Without Semantic" in our approach. In the second step, participants defined atoms and encoded rules 31, 32, and 36 with training and monitoring, considering semantic mapping. This step is denoted as "Week 2" in [3] approach and "With Semantic" in our approach.

Table 2. The comparison of the proposed rule encoding methodology with the previous approach by [3] in terms of defining atoms and rule encoding

Metrics	**Without Semantic**		**With Semantic**	
	[3]	**Our**	[3]	**Our**
Defining Atoms				
Week 1 / Independently	4.27%	46.83%	39.62%	58.22%
Week 2 / With monitoring	57.64%	88.9%	84.86%	97.53%
Encoding				
Week 1 / Independently	0%	18.05%	16.85%	37.5%
Week 2 / With monitoring	1.01%	86.7%	38.72%	97.33%

Table 2 provides a comparative analysis of the two approaches in terms of defining atoms and rule encoding. It demonstrates that in both cases of defining atoms and rule encoding, our proposed encoding methodology outperforms the previous approach by [3]. Regarding defining atoms, our proposed approach works significantly better than the approach used in the previous experiment without semantic consideration. When semantic mapping is considered, the progress of our proposed approach is slightly superior to the other approach. In terms of rule encoding, our proposed methodology also exhibits superior performance compared to the approach by [3]. However, the intervention methodologies in the two approaches in the second phases (those labelled as "Week 2") were different. In [3] the coders had a common session to agree on a shared set of atoms and then proceeded to encode the rules independently (and create new atoms as needed). While the experiment reported in this paper, the coders were instructed to analyse the textual provisions in some specific ways and examples were provided. Moreover, the coders were already accustomed to coding legislation, having worked on previous coding projects in DDL, and they were native English speakers.

5. Conclusions

This paper reports on a methodology to encode legal provisions in Defeasible Deontic Logic and on an empirical evaluation of the proposed methodology. The results of the experiment are encouraging. They show that the adoption of the proposed methodology drastically increases the degree of alignment on the encoding produced by different coders. While the results are promising, there are some limitations: the coding was restricted to a small (though significative and somehow challenging) set of provisions, and the number of participants (in both experiment) was low (comparison on 3 datasets by 3 coders in [3] and 2 datasets by 2 groups of coders in our experiment). Thus, we believe that more extensive experiments are needed to further validate the methodology.

 doi:10.20944/preprints202504.2295.v1

Finally, while the two experiments used Defeasible Deontic Logic as the encoding language (mostly to enable the coder with some tool to execute and play with the encoding), the methodology developed in this paper is neutral about the target language and it is compatible with the LegalRuleML standard. LegalRuleML is agnostic about the execution language and should provide an interoperable medium for multiple RaC frameworks.

Author Contributions: Conceptualization, H.B. and G.G.; methodology, H.B., G.G.; investigation, all authors; data curation, H.B.; resources A.S.; writing—original draft preparation, all authors; writing—review and editing, all authors. All authors have read and agreed to the published version of the manuscript.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study, and their voluntary participation in the experiment was compensated (70AUD).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DL	Defeasible Logic
DDL	Defeasible Deontic Logic

References

1. Mohun, J.; Roberts, A. Cracking the code: Rulemaking for humans and machines. Oecd working papers on public governance., OECD, Paris, France, 2020. https://doi.org/10.1787/3afe6ba5-en.
2. Waddington, M. Rules as Code. *Law in Context* **2020**, *37*, 179–186.
3. Witt, A.; Huggins, A.; Governatori, G.; Buckley, J. Encoding legislation: a methodology for enhancing technical validation, legal alignment and interdisciplinarity. *Artificial Intelligence and Law* **2024**, *32*, 293–324. cited By 2, https://doi.org/10.1007/s10506-023-09350-1.
4. Anna Huggins, Mark Burdon, A.W.; Suzor, N. Digitising legislation: connecting regulatory mind-sets and constitutional values. *Law, Innovation and Technology* **2022**, *14*, 325–354. https://doi.org/10.1080/17579961.2022.2113670.
5. Godfrey, N.; Burdon, M. Fidelity in legal coding: applying legal translation frameworks to address interpretive challenges. *Information and Communications Technology Law* **2024**, *33*, 153 – 176. Cited by: 0; All Open Access, Hybrid Gold Open Access, https://doi.org/10.1080/13600834.2024.2312620.
6. Wolfson, O.; Silberschatz, A. Distributed processing of logic programs. In Proceedings of the Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 1988; SIGMOD '88, p. 329–336. https://doi.org/10.1145/50202.50242.
7. Gamut, L. *Logic, language, and meaning, volume 1: Introduction to logic*; University of Chicago Press, 1991.
8. Gordon, T.F.; Governatori, G.; Rotolo, A. Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. In Proceedings of the RuleML 2009; Governatori, G.; Hall, J.; Paschke, A., Eds., Heidelberg, 2009; Number 5858 in LNCS, pp. 282–296. https://doi.org/10.1007/978-3-642-04985-9_26.
9. Palmirani, M.; Governatori, G.; Athan, T.; Boley, H.; Paschke, A.; Wyner, A. LegalRuleML Core Specification Version 1.0. OASIS Committee Specification 2, OASIS, 2020.
10. Governatori, G.; Olivieri, F.; Rotolo, A.; Scannapieco, S. Computing Strong and Weak Permissions in Defeasible Logic. *Journal of Philosophical Logic* **2013**, *42*, 799–829. https://doi.org/10.1007/s10992-013-9295-1.
11. Antoniou, G.; Billington, D.; Governatori, G.; Maher, M.J. Representation Results for Defeasible Logic. *ACM Transactions on Computational Logic* **2001**, *2*, 255–287. https://doi.org/10.1145/371316.371517.
12. Governatori, G.; Rotolo, A. Logic of Violations: A Gentzen System for Reasoning with Contrary-To-Duty Obligations. *Australasian Journal of Logic* **2006**, *4*, 193–215. https://doi.org/10.26686/ajl.v4i0.1780.
13. Governatori, G.; Rotolo, A.; Sartor, G. Logic and the Law: Philosophical Foundations, Deontics, and Defeasible Reasoning. In *Handbook of Deontic Logic and Normative Reasoning*; College Publications: London, 2021; Vol. 2, pp. 655–760.

14. Governatori, G. Practical Normative Reasoning with Defeasible Deontic Logic. In *Reasoning Web 2018*; d'Amato, C.; Theobald, M., Eds.; Number 11078 in LNCS, Springer International Publishing: Cham, 2018; pp. 1–25. https://doi.org/10.1007/978-3-030-00338-8_1.
15. Governatori, G. Practical Normative Reasoning with Defeasible Deontic Logic. In *Reasoning Web 2018*; d'Amato, C.; Theobald, M., Eds.; Number 11078 in LNCS, Springer International Publishing: Cham, 2018; pp. 1–25. https://doi.org/10.1007/978-3-030-00338-8_1.
16. Lam, H.P.; Governatori, G. The Making of SPINdle. In Proceedings of the International Symposium on Rule Interchange and Applications; Governatori, G.; Hall, J.; Paschke, A., Eds., Heidelberg, 2009; Number 5858 in LNCS, pp. 315–322, [papers/2009/ruleml09spindle.pdf]. https://doi.org/10.1007/978-3-642-04985-9_29.
17. Cristani, M.; Governatori, G.; Olivieri, F.; Pasetto, L.; Tubini, F.; Veronese, C.; Villa, A.; Zorzi, E. Houdini (unchained): An effective reasoner for defeasible logic. In Proceedings of the Proceedings of 6th Workshop on Advances in Argumentation in Artificial Intelligence; Confalonieri, R.; Porello, D., Eds. CEUR-WS.org, 2022, CEUR Workshop Proceedings.
18. Governatori, G. An ASP Implementation of Defeasible Deontic Logic. *Künstliche Intelligenz* **2024**, *38*, 79–88. https://doi.org/10.1007/s13218-024-00854-9.
19. Governatori, G.; Casanovas, P.; de Koker, L. On the Formal Representation of the Australian Spent Conviction Scheme. In Proceedings of the Rules and Reasoning; Gutiérrez Basulto, V.; Kliegr, T.; Soylu, A.; Giese, M.; Roman, D., Eds., Cham, 2020; Vol. 12173, *LNCS*, pp. 177–185. https://doi.org/10.1007/978-3-030-57977-7_14.
20. Bhuiyan, H.; Governatori, G.; Rakotonirainy, A.; Bond, A. Traffic Rules Compliance Checking of Automated Vehicle Maneuvers. *Artificial Intelligence and Law* **2024**, *32*, 1–56. https://doi.org/10.1007/s10506-022-09340-9.
21. Dimyadi, J.; Governatori, G.; Amor, R. Evaluating LegalDocML and LegalRuleML as a Candidate Standard for Sharing Normative Information in the AEC/FM Domain. In Proceedings of the LC3 2017: Volume I Proceedings of the Joint Conference on Computing in Construction (JC3); Bosché, F.; Brilakis, I.; Sacks, R., Eds., 2017, pp. 639–646. https://doi.org/10.24928/0012.