A Computational Approach to Competition Impact Assessment

Emerson S. Bañez*

July 3, 2025

Abstract

Competition does not take place in a vacuum but is embedded within an existing legal and regulatory environment. Competition authorities are thus encouraged to evaluate existing laws to identify and remediate competition effects. To this end, organizations such as the OECD and the World Bank have released guidance on the conduct of competition impact assessments. Despite the importance and complexities of a competition impact assessment, the literature is sparse when it comes to implementation specifics. From selection of laws to be reviewed to the actual assessment of legal provisions, much is left to the subjective evaluation of assessors. This could mean that errors could compound in the course of analysis and lead to implausible results. For example: 1. There are no parameters for law selection that aligns with market definitions; 2. There is no consistent, granular unit of analysis; 3. There is a lack of provable basis for attributing specific competition effects to legal texts. The work aims to apply techniques of computational law to these problems.

*Assistant Professor, University of the Philippines College of Law; LL.D. Student, Kyushu University Graduate School of Law.

First: The work will encode the norms applicable to a specific sector (digital payments market in the Philippines) into forms that are amenable to computational processing – such as modelling market entities and interactions into a knowledge graph, and the normative constraints into inference rules. Second, these representations can then be subjected to automated reasoning in order to provide insights useful to competition analysis, such as: determination of other relevant laws, evaluation for consistency and compliance, and proving of specific competition effects.

Keywords: Computational Law, Competition, Impact Assessment, Artificial Intelligence

1 Introduction

1.1 Competition and the Legal Environment

Competition does not take place in a vacuum but is embedded within an existing legal and regulatory environment.[1] Barriers to entry and exit (which can cause failure to produce a competitive market) can be due not just to the structural features of a market, or the behavior of its actors, but also the policy environment maintained by the government.[2] The law can affect competition in a number of ways: It can openly favor some players, providing them with tax exemptions, and subsidies. It can also put other players at a disadvantage, by making it more expensive for them to operate in the industry (through barriers to entry and exit). In both of the above cases, the law works explicitly in limiting competition through advantages and constraints directly addressed to industry players. The enactment

[1]This evokes the vision of law not as a neutral, static stage, but as Laurence Tribe describes it - one possessed of a curvature, a shape, that can affect the movement of the actors on it. See Generally Laurence H. Tribe. "The Curvature of Constitutional Space: What Lawyers Can Learn from Modern Physics". In: *Harvard Law Review* (Nov. 1989), pp. 1–68.

[2]See Erlinda M Medalla. "Understanding the New Philippine Competition Act". In: *Philippine Institute for Development Studies (PIDS) Discussion Paper Series* (No. 2017-14 2017), pp. 1–24. URL: http://hdl.handle.net/10419/173591 (visited on 01/08/2024), at 5.

of a competition policy, specifically, anti-trust law, is explicitly directed at the competitive behavior of firms, and is designed to restrain monopoly and maintain market competition.

However, even the general legal environment outside of competition law can also work against competition through more subtle mechanisms. The law can control the flow of information between and amongst buyers and sellers, constraining their strategic choices. More importantly, the law can allow the state itself, with its size, economic power, and monopoly on regulatory powers, to be a direct player (as a buyer or seller) in any industry.[3]

1.2 Competition Impact Assessments

The law's impact on competition underscores the need for detailed studies on how the current legal and regulatory backdrop affects competition. This can be performed through a **competition impact assessment** of the laws that operate in specific sectors of economic activity. A competition impact assessment refers to a systematic, evidence-based process of reviewing existing or proposed policies in order to determine their impact on competition. This is with the view to formulating alternative policies that are more conducive to competition.[4] The underlying logic is that while governments may pursue important policy goals

[3]In some cases, the general legal environment is just us important as the competition law in the maintenance of a competitive market. See generally Iftekhar Hasan and Matej Marinč. "Should Competition Policy in Banking Be Amended during Crises? Lessons from the EU". in: *European Journal of Law and Economics* 42.2 (Oct. 2016), pp. 295–324. ISSN: 0929-1261, 1572-9990. DOI: 10.1007/s10657-013-9391-2. URL: http://link.springer.com/10.1007/s10657-013-9391-2 (visited on 01/08/2024), which suggests that competition policy in the financial sector can be inconsistent in times of crisis. Financial regulators, through prudential standards, bear the greater responsibility in ensuring against concentration. To the extent that this overlaps with market structure concerns of competition authorities, greater coordination is required. See also Tomaso Duso, Jo Seldeslachts, and Florian Szücs. "The Impact of Competition Policy Enforcement on the Functioning of EU Energy Markets". In: *The Energy Journal* 40.5 (Sept. 2019), pp. 97–120. ISSN: 0195-6574, 1944-9089. DOI: 10.5547/01956574.40.5.tdus. URL: http://journals.sagepub.com/doi/10.5547/01956574.40.5.tdus (visited on 01/08/2024), Competition policy may have significant impacts, but only to the lightly regulated sectors. On the other hand, highly-regulated firms are less likely to respond to competition policy.

[4]Nakaizumi defines Impact Assessment as "a process that prepares evidence for political decision-makers on the advantage and disadvantage of possible regulatory options by assessing their potential impacts" and Competition Assessment as a form of Impact Assessment that assesses "whether the

through legislation - there are multiple pathways to these goals, and governments should pursue those paths that least impact competition. This in turn springs from the premise that more competition is beneficial, especially for the consumers.[5].In addition to isolating and predicting the effect of laws, a competitive impact analysis can lead to greater transparency in policymaking, by disclosing the law's interactions with other commitments, as well as providing a venue for experts and other stakeholders to participate[6].

The Philippine Competition Commission has already conducted several such assessments of selected laws - either at its own instance or upon request by Congress or regulatory agencies. It has also worked with organizations such as the OECD, which has performed competitive impact assessments of certain economic sectors.[7] The OECD and other organizations interested in advocating for competition policy have also issued guidelines for the conduct of competition impact assessments.[8] The Philippine Competition Commission cur-

---

regulation unnecessarily restricts competition or not". Takuya Nakaizumi. *Impact Assessment for Developing Countries: A Guide for Government Officials and Public Servants.* Contributions to Economics. Singapore: Springer Nature Singapore, 2022. ISBN: 978-981-19549-3-1 978-981-19549-4-8. DOI: 10.1007/978-981-19-5494-8. URL: https://link.springer.com/10.1007/978-981-19-5494-8 (visited on 07/14/2024), at 3; An impact assessment refers to both the textitex-ante evaluative process as well as the output of that process Nakaizumi, *Impact Assessment for Developing Countries*, at 6.

[5]OECD. *Competition Assessment Toolkit - Volume 1 (Principles).* 2019. URL: https://www.oecd.org/daf/competition/46193173.pdf (visited on 10/10/2023), at 7.

[6]Nakaizumi, *Impact Assessment for Developing Countries*, at 6-7.

[7]See for example OECD. *Competition Assessment Reviews: Logistics Sector in the Philippines.* 2020. URL: https://www.oecd.org/daf/competition/oecd-competition-assessment-reviews-philippines-2020.pdf (visited on 10/10/2023); See also OECD. *Competitive Neutrality Reviews: Small-Package Delivery Services in the Philippines.* 2020. URL: https://www.oecd.org/daf/competition/oecd-competitive-neutrality-reviews-philippines-2020.pdf (visited on 10/10/2023).

[8]See OECD, *Competition Assessment Toolkit - Volume 1 (Principles)*, for Part 1 of the Organization for Economic Cooperation and Development's (OECD) 3-part guidelines for competition impact assessment. This and subsequent volumes will be referred to collectively as the "OECD Guidelines"; See OECD Guidelines, supra. See also the International Competition Network's (ICN) recommended practices. Subsequently referred to as the "ICN Guidelines" ICN Advocacy Working Group. *Recommended Practices on Competition Assessments.* International Competition Network, 2014. URL: https://www.internationalcompetitionnetwork.org/wp-content/uploads/2018/07/AWG_RP_English.pdf (visited on 01/10/2024).

rently has unpublished draft guidelines[9] that it uses to guide its competition assessment exercises. The PCC Guidelines disclose that it is based on the OECD Guidelines as well as the World Bank's Markets and Competition policy Assessment Toolkit.[10] the full documentation of which is not publicly available. To the extent that these guidelines and instances of their implementation converge into common methodology, these guidelines will be idealized into a "canonical approach" to competition impact assessment, and represented by the OECD Guidelines as the focus of analysis.

1.3 The Canonical Approach to Competition Impact Assessment

The canonical approach to conducting competition assessments starts with identifying laws that are relevant to a sector, then proceeds to evaluating such laws for competition effects. As elaborated in the OECD Guidelines, the process of competition impact assessment involves the following steps:[11]

1. **Identify the laws to be assessed** - This can be straightforward in the case of assessing new or pending legislation or regulation. On the other hand, for situations where the impact of laws on an entire economic sector is required, discretion is involved in defining the boundaries of what will be reviewed. This is expected to result in a list of "relevant laws".

2. **Apply threshold tests** - The list of relevant laws can be narrowed down through a threshold test. This is based on a checklist of questions designed to identify potential restrictions to competition. This will result in a smaller set of flagged laws that can

[9] "PCC Guidelines", on file with the author.

[10] See The World Bank. *Markets and Competition Policy.* World Bank. URL: `https://www.worldbank.org/en/topic/competition-policy` (visited on 01/16/2024), Subsequently, "the World Bank Assessment Toolkit".

[11] The enumerated steps are from OECD. *Competition Assessment Toolkit - Volume 3 (Operations Manual).* 2019. URL: `https://web-archive.oecd.org/2020-01-22/370055-COMP_Toolkit_Vol.3_ENG_2019.pdf` (visited on 10/10/2023), at 14-15.

be subject to a more detailed review.

3. **Detailed review of flagged laws** - Performing a more detailed review to determine whether or not there are "actual and significant" restrictions on competition. Those with such restrictions form a set of "critical laws" for which the next stage of the process should be applied.

4. **Generate alternatives** - For those critical laws where restrictions are found, identify alternative measures that can achieve policy objectives while being less restrictive or competition.

5. **Selecting the best option** - Once policy alternatives have been generated, a judgment must be determined as to the "best" option. Once the "best" option has been identified, legislation must be drafted and passed that will implement this policy recommendation.

6. **Ex-post assessment** - Review and monitoring of the impacts of the law implementing the selected policy alternative.

The canonical approach requires a search methodology to enumerate the laws that can apply to the actors and transactions in a given market. The documentation assumes that the government will select or prioritize a sector to be assessed. The guidelines suggests a number of prioritization principles to aid in this determination, such as: 1. Selecting sectors with high economic impact (in terms of share of GDP, consumer expenditure, employment); 2. Or those have been the subject of frequent complaints or interventions; 3. The constraints of time, financial resources, and the availability of technical talent.[12] For the purpose of this work, it will be assumed that selection and prioritization can proceed independently, prior to the methodology to be outlined in this work. Aside from simplifying the scope of the work, the assumption is compatible with the notion that selection and prioritization of the sector is a matter of policy, to be made by accountable, human institutions.

[12] OECD, *Competition Assessment Toolkit - Volume 3 (Operations Manual)*, at 18-19.

Once a sector has been selected, the next step is to compile legislation that is relevant to the sector. This, in turn, is predicated on delineating a conceptual boundary for the sector. The guideline acknowledges that a boundary-setting exercise can be challenging. To provide some structure into this exercise, the guidelines provide some suggestions on how to proceed: 1. Focusing on legislation relevant to one ministry. Using the correlation with the ministry concerned as a proxy for a relevant boundary, however, simply restates the problem especially where the ministry has a broad mandate. It can also risk missing laws that require inter-agency coordination; 2. Focusing on standard definitions - This can be done by referring to standard industry classifications, such as the United Nation's International Standard Industrial Classification of All Economic Activities[13], or the Statistical Classification of Economic Activities in the European Union[14]. The guideline, however, notes that these classification systems will often segregate economic activities in ways that are counter to both intuition as well as grounded knowledge as to how industries are actually run.

Assuming that the boundary of a market sector can be defined for purposes of the finding relevant laws - this process may still yield numerous laws for any modern regulatory environment.For this, the canonical approach suggests a process for filtering relevant laws in order to arrive at a set of critical laws.[15]

1.4 Problems with the Canonical Approach

Despite the importance of evaluating the competition impact of the legal environment, the methodological toolset for impact assessment has fallen behind (in terms of sophistication and rigor) those used in other areas of competition policy, such as: merger control, assess-

[13]United Nations. *International Standard Industrial Classification of All Economic Activities (ISIC)*. Revision 4. United Nations, 2008. ISBN: 978-92-1-161518-0. URL: `https://unstats.un.org/unsd/publication/seriesm/seriesm_4rev4e.pdf`.

[14]European Commission. *Statistical Classification of Economic Activities in the European Union*. Rev. 2. 2008. URL: `https://ec.europa.eu/eurostat/documents/3859598/5902521/KS-RA-07-015-EN.PDF` (visited on 05/13/2024).

[15]OECD, *Competition Assessment Toolkit - Volume 3 (Operations Manual)*, at 17.

ment of anticompetitive agreements and abuse of dominance.[16] According to the OECD Guidelines, Step 3 and the associated competition checklist lie at the heart of the competition impact assessment process. Despite the importance assigned to this section of the process, both the OECD Guidelines and the literature on competition impact assessment do not provide a detailed, rigorous, and consistent methodology for performing this step. The OECD Guidelines provide a checklist of questions that can be used to identify potential restrictions to competition. However, the OECD Guidelines do not provide a detailed methodology for applying the checklist. Much is left to the subjective evaluation of assessors. This could mean that errors could compound in the course of analysis and lead to implausible results. This rise to the following problems:

Law selection Assuming that the economic sector has been selected and its conceptual boundaries have been delineated, the assessor is expected to derive from this model "an exhaustive list of laws and regulations that influence the economic activities that take place in each of the sectors under examination".[17] There is no elaboration as to how the conceptual mapping in the previous step can translate into a search strategy that can be documented, refined, and shared. There are no parameters for law selection that aligns with market definitions. The literature require that laws relevant to a market be subject to competition analysis. Although there are well-established methods for defining a market, criteria for selecting laws that will be subject to analysis are not aligned with these market definitions. Assessors are likely to under- or over-select the laws.

[16] See Nicole Robins and Hannes Geldof. "Ex Post Assessment of the Impact of State Aid on Competition". In: *European State Aid Law Quarterly* 17.4 (2018), pp. 494–508. ISSN: 16195272, 21908184. DOI: 10.21552/estal/2018/4/6. URL: http://estal.lexxion.eu/article/ESTAL/2018/4/6 (visited on 01/03/2024), which proposes a greater role for financial and economic analysis in evaluating the impact of state action on competition, at 494-495.

[17] The Guidelines acknowledge that this stage of the process is not trivial, since ensuring the inclusion of all relevant legislation requires casting a broad net. The guideline suggests an iterative process, which requires not just reference to texts initially found in electronic databases (as well as the laws they refer to, such as implementing rules), but also through consultation with stakeholders. OECD, *Competition Assessment Toolkit - Volume 3 (Operations Manual)*, at 21.

Unit of analysis No consistent, granular unit of analysis. Competition authorities may look at individual laws and analyze these for competition impact. However, a statute may not be the appropriate unit of analysis, since the competition impacts operate through key provisions that work with other critical provisions found in other laws. Looking at more atomic levels of distinct rules within provisions can also enable more detailed forms of analysis.

Lack of proof Lack of provable, measurable basis for correlating textual provision with an anti-competitive effect. The purpose of impact assessment is to provide objective basis for decisions, resulting in policy that is better in the long term even if it clashes with immediate emotional considerations.[18]Even if a law is properly selected and studied at the appropriate level of description, the actual evaluation of competition impact is described on an intuitive, sometimes *ad hoc* basis. It is not encoded in a way that can be reliably communicated, proved, and further analyzed.

1.5 Specific Problems with Law Search and Selection

The first step in the canonical approach to competition impact assessment is to find the laws that are applicable to a sector. Given an industry or market sector under consideration - What law "covers" an i dustry or a market with all its actors and transactions? The goal is to arrive at either a complete or a heuristic but consistent mapping between actors (and their actions relevant to a market) - and the laws that would cover these actors and actions.

Risk of under-inclusion In evaluating a legal environment, assessors may overlook critical legal standards pertinent to the industry, due to a disconnect in the terms used by the industry and those employed in legal contexts. This phenomenon is particularly evident in emerging digital finance businesses. These entities often operate under novel designations or through distinct modalities, such as exchanges and applications. Consequently, there is

[18]Nakaizumi, *Impact Assessment for Developing Countries*, at 4.

a prevailing misconception that such entities fall outside the purview of traditional financial regulations. This overlooks the fact that, despite their innovative approaches, these entities perform analogous functions and are subject to similar risks as their conventional counterparts.

On the other hand, focusing on a single law nominated by a stakeholder may result in losing many critical signals. This is because competitive issues such as barriers to entry, disproportionate costs, and preferential treatment can arise not from a single law but from the interaction of key provisions spread across several legislative enactments.

The OECD framework cautions that "When performing this exercise, it is important to remember that, in addition to sector-specific regulation, there also exists horizontal, cross-sectoral, legislation (such as planning restrictions or environmental standards) that may have a considerable impact on the economic activities performed in that sector and may be a cause of additional competition restrictions."[19]

Risk of over-inclusion If we take a connected view of the law, all of the law can be relevant to a particular market. Every law has the potential to change the relative rights and obligations of parties involved in economic activity and thus result in some competitive impact. Considered this way, even laws that apply to everyone and only incidentally touch economic activities in a sector - can be interrogated for potential competitive impact, no matter how small or contingent. The Philippine Civil Code, for example, gives a preferential status for unsecured debt evidenced by a notarized instrument, over those that are embodied in a private document. Family and tax law provides rights and privileges that are accessible only to married heterosexuals. Nevertheless, these laws should usually not be the subject of competitive impact analysis. As they apply to everyone, in a large enough population their application to specific individuals can appear random and evenly distributed - any *de minimis* competitive impact is contingent and cancel each other out. More importantly,

[19]See OECD, *Competition Assessment Toolkit - Volume 1 (Principles).*

these laws do not directly relate to the sector under consideration, and their ultimate effects on the actors in the sector will only be coincidental.[20]

A lawyer's theoretical training and experience, and openness to economic thinking can perform the required analysis while compensating for these limitations. However, that lawyer may not always be available. It is also possible that the limited pool of legal talent will not be able to scale to the demands of extensive, industry-wide competition analysis, which could involve hundreds (if not thousands) of laws and regulations, all of which could interact with each other. Abstracting the problem into a computational form can allow parts of the analysis to be done by non-lawyers (i.e., the staff of a competition authority), or even by computers. The goal is not to supplant the human component of competition impact analysis but to augment it.

1.6 Computational Law in Aid of Competition Impact Assessment

These problems relating to scale, rigor, consistency, and predictability may be the appropriate setting for the application of computational law, which can look at legal rules as discrete units that can be evaluated. In this light, the question of competition impact can be structured as a computational problem. This work aims to apply computational techniques to: 1. The selection of laws for competition impact assessment, based on their relevance to a market; 2. The representation of legal rules into discrete computable units; 3. The automated analysis and evaluation of legal rules for their competition impact. It hopes to introduce improvements to the problem of search and prioritization of laws: That

[20] "Plaintiffs stress that the LMRDA is a remedial measure and seek a liberal construction. This maxim is useless in deciding concrete cases. Every statute is remedial in the sense that it alters the law or favors one group over another... But after we determine that a law favors some group, the question becomes: How much does it favor them? Knowing that a law is remedial does not tell a court how far to go. Every statute has a stopping point, beyond which, Congress concluded, the costs of doing more are excessive — or beyond which the interest groups opposed to the law were able to block further progress." - Richard Stomper, et al., Plaintiffs-appellees, v. Amalgamated Transit Union, Local 241, Defendant-appellant, 27 F.3d 316 (7th Cir. 1994)

is, the process of making an exhaustive mapping of concepts in a market in order to identify relevant laws, as well as applying the threshold tests in a consistent and rigorous manner.

Automated reasoning can also allow competition authorities and policy makers to make extensive evaluations efficiently and at scale. A responsive competition assessment system should not only evaluate retroactively, for existing laws, but also conduct the exercise for new or proposed legislation. While it is possible for each new law to carry not only prospective effects in a particular subject matter - it is also possible for it to interact with the existing legal environment in a way that would change the competition impact of prior laws. A new baseline understanding of the competition impact of the entire legal environment may have to be inspected with the passage of each new law, compounding the complexity of the task and adding to the burden of competition authorities. Part of this regression analysis can be automated through the computational approach.

It should be noted that the primary concern of this paper is in finding the relevant laws as well as evaluating them for competition effects into a computable problem. The work will explore formalizations for representing the above problems in a way that can be processed by computers. The emphasis is in developing tools for the facilitation of competition impact assessment. This assumes that an appropriate body is responsible for conducting such assessment. This and other features of a competition assessment regime, such as the location of the assessment in the larger policy development process, the involvement of the competition agency, will not be within the scope of this work. Although Computational Law is usually associated with automation of legal determinations though a computer - it is not the goal of this work to implement an automated counterpart for the sections of the competitive impact assessment process that will be encoded into a computational form. Some experimental code might be featured in order to demonstrate the feasibility of some proposals, but these are not production-quality implementations. It should be noted that the advantage of the computational approach goes beyond machine execution of routine legal tasks, but in helping develop notations through which we can understand and share problems of legal reasoning.

Computational law and good institutional design Nakaizumi (2022) suggests that impact analysis will be more effective if embedded within good institutional design with the following components: 1. Established ground rules; 2.Guidelines on the conduct of assessments;3. Separate institutions tasked to conduct assessments;4. Involvement of economists or other scientific researchers. A computational approach to competition impact assessment directly contributes to the first two components by making explicit the standards, definitions, and procedures of an assessment exercise. To the extent that the computational encoding is transparent, portable, and accomodates the input of experts, it can also contribute to the last two requirements.

1.7 Scoping and Limitations

Finding a constrained version of the problem The goal of the proposed work is to use the computational approach for competition analysis. Particularly, for the stages of legal search, selection, and threshold testing. The scope of the study is further limited to applications for the special case of the Philippine digital payments sector - where both the assumptions, definitions, and constraints are more explicit.

The study can be limited to well-defined standards in competition - i.e. those that are already extensively documented and tested in the economics literature, and so can be a source of explicit rules on how a competitive market ought to behave. As to the laws that will be evaluated, the plan is to focus on a segment that is already digitized and subject to very specific constraints. The digital payments sector is a good candidate. Out of necessity, the sector does not involve many entities and transactions with open-ended states. It is also a field characterized by extensive, semantically rich constraints from industry standards, government regulations, user contracts, and the functionality of the digital platforms themselves.

Miscellaneous considerations : The problem of competition impact analysis has the same shape as other problems that involve the involve analyzing and evaluating an extensive body of rules (such as: 1. gap analysis; 2. impact analysis; 3. compliance analysis). They all involve some form of legal comparison and evaluation - old law against new law, n-level law versus n-1 level law, etc. So an advance in the solution of one problem can contribute to the other.

Why is all the effort towards abstraction preferable to the usual intuitive approach? In addition to gving us scale and automation, the computational approach can help us in two ways: 1. It can help us make our analysis more rigorous, and 2. It can help us make our analysis more transparent. Sharing legal knowledge through a formalized notation can help us build richer systems of legal knowledge.

It should be noted that regardless of the formalization, inference rules, defeasible deontic logic is already embedded in the practice. Just as a baker can make a cake without knowing the chemistry of baking, we can make legal determinations without knowing the formalisms of logic. But just as knowing the chemistry of baking can help us make better cakes, knowing the formalisms of logic can help us make better legal determinations.

Despite the initial wariness about the costs and consequences of large language models, their growing sophistication is compelling. Recent literature suggests that knowledge graphs can embedded into large language models, making the latter more efficient, more attuned to "ground truth", and therefore more reliable. Since both argumentation frameworks and proposition networks can be framed as extensions of the information contained in knowledge graphs, it may be possible to combine these approaches as well.

2 An Overview of Computational Law

2.1 Historical background

The project of applying computational techniques to the legal domain - e.g. encoding law into computational terms, and mechanically applying or analyzing these - was among the earliest directions of artificial intelligence research. Despite its early promise, however, the approach did not bear fruit.[21] During the 1980's there was initial optimism about the prospect of computers performing automated legal reasoning. Grossman summarizes the research and programming activity towards this end. They note that while computers cannot replace lawyers, these machines can, in time nevertheless run "legal reasoning systems" that can assist attorneys.[22] Computerized legal reasoning offered speed, reliability, and the ability to carry out numerous,repetitive tasks. It could also provide a consistent application of the law.[23] It was also hoped that the availability of such systems could have knock-on effects on legal reasoning itself, molding the thought processes of legal professionals towards logical rigor, and force the field to be more explicit about its assumptions.[24]

Initial approaches Early attempts at replicating legal reasoning through software tried to emulate the fact that lawyers employed both deductive and analogical reasoning when working on a case[25]:

1. **Deduction -** Looking at conditions, propositions in the law as well as fact patterns,

[21] See Michael Genesereth and Nathaniel Love. "Computational Law". In: *Proceedings of the 10th International Conference on Artificial Intelligence and Law - ICAIL '05*. The 10th International Conference. Bologna, Italy: ACM Press, 2005. ISBN: 978-1-59593-081-1. DOI: 10.1145/1165485.1165517. URL: http://portal.acm.org/citation.cfm?doid=1165485.1165517 (visited on 09/18/2021), at 205.

[22] Garry S Grossman and Lewis D Solomon. "Computers and Legal Reasoning". In: *ABA Journal* 69 (1983), pp. 66–70, at 66: "Primarily, a legal reasoning system would serve as a repository of knowledge, outlining the general parameters of the law. In lieu of searching through a treatise or similar task, given a specific factual situation, the system could be relied on to present only the relevant law."

[23] Ibid.

[24] Ibid.

[25] Ibid., at 67.

and then making inferences towards legal conclusions. For example: "If A or B then C". The computer stores representations of operations (e.g. the inference from A to B), as well as their premises (e.g. what A, B, and C stand for.)

2. **Analogy -** Looking at analogous cases, i.e., those that may have different fact patterns but similar relationships. A computer can attempt to reason by analogy by searching for relationships in fact patterns similar to those of the case at hand. The system can apply the rule of one case to another based on their similarity.

Instances of inductive reasoning in law?

An example of the deductive approach was JUDITH, developed in the early 1970's by Walter Papp and Bernhard Schlink. The law was modeled as a set of premises (as defined by its programmers). The user goes through the premises that may be stereotypical for a given problem, determining whether they applied or not (True or False). Based on this knowledge, the system attempted to determine whether a cause of action exists under a given set of facts.[26]

On the other hand, the TAXMAN system by McCarthy had an approach similar to analogy.[27]Instead of asking specific questions (like Helawell's system) it maintained an internal representation of: 1. The fact pattern at hand and 2. Fact patterns that are inherent or usually associated with corporate reorganizations. These representations come in the form of "semantic networks", compound statements elaborating the legal relationships in these fact patterns. Users were expected to enter a fact pattern (in a formal, structured language). The computer would then search through the semantic network for similar relationships.

[26]Another example of a system from this period using the deductive approach is Hellawell's tax plannning systems - one used to determine the treatment of redemptions, and another for the optimal choice of foreign subsidiary. Both systems involved no attemot to create an internal modek of the relevan laws. Instead, the explicit tests were programmed directly into the system, and tailored to specific problems. The design and implementation of both systems were not adaptable to other areas of law Grossman and Solomon, "Computers and Legal Reasoning", at 67.

[27]Ibid., at 67-68.

This required all relevant relationships to be thought of beforehand and represented in a formal language.[28]

The systems mentioned in the previous paragraphs were limited in terms of the legal problems they addressed. They dealt with areas where there were fewer ambiguities, or had rules that are susceptible to mechanistic analysis. They were also limited by the technology available at that time. Making the internal representations of facts and laws involved complexity and a lot of resources. None of the programs had standardized, user-friendly interfaces. Even then, their solutions were often superficial and were thus of limited value in real world settings.[29]

2.2 Definition and contemporary developments

According to Genesereth:"Computational law is that branch of *legal informatics*[30] concerned with codification of regulations in precise computable form."[31]. In terms of practical applications - it can provide the basis for computer systems performing compliance checks, legal planning, the analysis of regulations, and related functions. Many computer applications aid lawyers in their tasks, but these are not within the ambit of the term. Examples include legal databases to find the law, and office productivity suites to help the practi-

[28]On the other hand, Meldman's query-based system for assault and battery cases is cited as another example of a system that applies analogy. It works by taking in as input a series of word groups that describe the facts of a case. The system will then state whether it can identify cases wuth similar fact patterns. It can be classified as a research tool rather than a system for automating legal reasoning. Grossman and Solomon, "Computers and Legal Reasoning", at 69.

[29]Ibid., at 69.

[30]"Legal informatics is defined as the study of information, its technology, and it implication and impact in the field of law". This is to be differentiated from "Computer law", which is concerned with "problems relating to the social implications of information technology in the field of law." Christopher L. Hinson. "Legal Informatics: Opportunities for Information Science". In: *Journal of Education for Library and Information Science* 46.2 (2005), p. 134. ISSN: 07485786. DOI: `10.2307/40323866`. JSTOR: `10.2307/40323866`. URL: `https://www.jstor.org/stable/10.2307/40323866?origin=crossref` (visited on 11/02/2023), at 134-135.

[31]See Michael Genesereth. *Computational Law: The Cop in the Back Seat.* CodeX: The Center for Legal Informatics Stanford University. 2015. URL: `https://law.stanford.edu/publications/computational-law-the-cop-in-the-backseat/` (visited on 09/18/2021), at 2.

tioner prepare briefs, or systems to automate the backroom functions of the law office. In these instances, the legal reasoning is still performed by the human agent. The computer performs symbolic analysis for purposes of retrieval and presentation of data, without any recognition of the rules as such.[32]From a pragmatic perspective, Computational Law is important as the basis for computer systems capable of doing legal calculations, such as compliance checking, legal planning, regulatory analysis, and so forth".[33] The touchstone of the computational project is the creation of "Codex Machine" which contains within itself an extensive databases of encoded rules, and with all the required computational resources, provide responses indistinguishable from that made by a legal professional.[34] Despite the recency of the term, its goal is shared by early projects in artificial intelligence, which saw the legal domain as a natural site for the application of computational techniques.[35]

The prevalent approach to meeting these goals has two components: 1. First, representing law (and surrounding facts) into a formal logical form and 2. Second, the ability to process those representations to assist in legal determinations. This means: That it would be

[32]See Genesereth, *Computational Law: The Cop in the Back Seat*, at 2-3 for a proposed example. According to Genesereth the Turbo Tax program is a computational law application. The user supplies values, and the program makes computations of the user's tax obligation. When prompted, it can explain its results by making references to the applicable tax law. Legal rules (whether or not taxable, the base, rate, and tax due) are encoded (however indirectly) as code, and the result of the processing is a legal determination - whether or not tax is due, and how much. But see Hans Andersson. "Computational Law: Law That Works Like Software". CodeX – The Stanford Center for Legal Informatics, Feb. 10, 2014. URL: `https://www.academia.edu/9286857/Computational_Law_Anderrson_and_Lee`, at 3-4. There is a tendency to invoke a Turing test analogue for computational law system: "Any system whose users inputting, through whatever interface such system might present, a legal query to obtain a legal response would find themselves unable, given only the response, to determine whether a legal professional...had provided the system's response." Andersson rejects this criteria because it would include systems that only outwardly appear to be computational law without actually solving its fundamental problems. Based on his rejection of the Turing or "imitation" principle of what constitutes computational law, Andersson argues that Love and Genesereth's inclusion of Turbo Tax within the definition is inaccurate. Although the program appears to replicate the behavior of a tax professional - it does not formally represent laws, or performs automated reasoning based on those representations. This author notes that the whole point of the imitation principle is that the intent of representation does not matter - thus avoiding many philosophical questions.

[33]Genesereth, *Computational Law: The Cop in the Back Seat*, at 2.

[34]Andersson, "Computational Law: Law That Works Like Software", at 16.

[35]Genesereth and Love, "Computational Law", at 205.

possible through computational techniques, to arrive at consistent, correct (or at least plausible) legal conclusions from given set of premises and operations.

These determinations can be descriptive, recreating in computational form the law as it is, and guiding its users in evaluating whether certain actions or states of the world are in accordance with the encoded rules. It can also be prescriptive, meaning the rules as encoded can be analyzed and evaluated against standards (such as efficiency), or their alignment with other rules, in order to arrive at more suitable rules.[36] Despite its aspiration of being a visible system of explicit rules, so much of the law is actually dependent on tacit knowledge, i.e.: Other, higher order rules (for determining applicability, validity, interpretation) entities and concepts that are not provided in the legal text. Part of the project of computational law is to surface that tacit knowledge.

2.2.1 Computable contracts

Wolfram on the other hand sees computational law as part of a larger trend towards abstraction and formalization, not just in the law but in all spheres of human activity. The development of language and systems of writing themselves can be thought of as an initial step in this trend.[37] Written language enabled law to have coherent, codified forms, as well as a record for deciding ground facts and establishing precedent. While fields such as the natural sciences have progressed in terms of abstraction and formalization to build more intricate systems of knowledge, the law has lagged behind. What is required is the development of a symbolic discourse language for communication of legal and normative concepts, not just with each other, but with computers. Wolfram uses contracts as the starting point for demonstrating the feasibility of formalization and its consequences: A

[36] Andersson, "Computational Law: Law That Works Like Software", at 7.

[37] Stephen Wolfram. "Computational Law, Symbolic Discourse and the AI Constitution". In: *Data-Driven Law: Data Analytics and the New Legal Services*. Ed. by Ed Walters. Red. by Jay Liebowitz. Data Analytics Applications. Boca Raton, FL: CRC Press Taylor & Francis Group, 2019, pp. 144–174. ISBN: 13: 978-1-4987-6665-4. URL: `https://writings.stephenwolfram.com/2016/10/computational-law-symbolic-discourse-and-the-ai-constitution/` (visited on 01/14/2024), at 156.

contract in computational form can be defined relative to a set of underlying laws, that serve as the built-in functions of his hypothetical "symbolic discourse language".[38] Once a contracts are converted into a program written in a symbolic discourse language, we can perform all sorts of operations - like determining if a contract implies a certain outcome (or is contrary, complimentary with other contractual and normative commitments).[39]

One consequence of the computability of contracts is that these can then take in inputs from a variety of sources (including other computable contracts), in order to resolve automatically.[40]. The usefulness of computational contracts will depend on what the inputs are (and their quality, availability).[41] Some of those inputs will be natively computational - like the latency of a system, or the amount of digital currency present in an account. as more and more transactions become online, these type of inputs will be more useful. However, not every input is born digital, and will need to take into account the state of things and events in the outside world. Digital analogues may be available for some of these inputs, such as GPS coordinates for location, as well as IoT sensors for basic physical measurements(weight, temperature, vibration). For more complicated inputs that are required to produce legal consequences (i.e., is a person dead, or did the delivered goods meet the stipulated quality standards) may require either manual input, or the use of AI.[42]

[38]Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution", at 157.

[39]Wolfram also notes that even if built on a computational strata, our computable contract may still come across a problem of formal undecidability. i.e. there is no guarantee, even with a formal problem definition, that it is susceptible to solution based on systematic finite computation. ibid., at 158.

[40]Similar to, but at a higher scale and level of complexity, the automated "working out" of options transactions in electronic markets. ibid.

[41]These inputs could include: 1. Intrinsic - Such as the computer's date and time; 2. Extrinsic - Publicly accessible data like stock price, temperature, or a seismic event (which can be consolidated or mediated through something called an "oracle" that a computational contract has access to); 3. Non-public information - Humans, or machine learning systems can intervene ibid., at 162-163.

[42]The AI component, which may use machine learning techniques, will be less transparent and subject to algorithmic biases - just like human determinations which can also opaque and biased. The AI determinations, on the other hand, are at least more amenable to systematic analysis. To ensure its reliability, this component can be subjected to a security-risk model of evaluation and subject to cycles of exploit and patching ibid., at 160.

Computational contracts can be self-enforcing, automatically running just like any software process. A counterweight to this autonomy is the trustworthiness of the computed determinations - i.e. how can we be sure that the computation was reached with integrity (i.e. that the process was neither hacked nor erroneous)?[43] Wolfram imagines that existing contracts written in natural languages can be translated into a symbolic discourse language, which should be complete and expressive enough to describe ethical and normative systems.[44] More likely, however, new contracts can be written directly into the symbolic discourse language.

Adding a computable element to contracts is a way to deal with the growing cost and complexity of transactions. Wolfram also suggests that binding agreements, expressed in computational terms, may also be the means through which we can communicate normative constraints to Artificial Intelligence.[45]

2.2.2 Approaches to Implementing Computational Law

There are two distinct approaches to implementing computational law systems:[46] 1. The **semantic approach**, which focuses on structures and the meaning of legal concepts; 2. The **syntactic approach**, which depends on algorithms to tease out emergent connections from the data.Taking one approach over the other can involve tradeoffs in terms of accuracy, reliability, and cost.

[43]Technologies such as blockchain, encryption, as well as regular audits may help address these concerns. Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution", at 162.

[44]In cases of ambiguity, the translator-programmer can select an authoritative version, or provide alternative interpretations. ibid., at 163-164.

[45]The constraints we need to enforce on AI will have to be natively computational, since the behavior and possibilities of AI may be too broad, too complex to be expressed in natural language law. It would also not be enough to make it ingest the whole corpus of the law in natural language text as training data: It may be dangerous to give A.I. vaguely couched natural language constraints, since by default it will only literally follow the letter of the law, and exploit ambiguity to achieve hard-coded goals. ibid., at 167.

[46]Andersson, "Computational Law: Law That Works Like Software", at 10.

Semantic approach The semantic approach aims to represent the normative content of laws, rules, contracts into machine-readable formats. At the practical level, it involves humans looking at the raw legal text, pre-processing the text by looking for the meaningful elements, and encoding them into a structured format (e.g. markup languages based on the eXtensible Markup Language, or XML). The primary use-case for the approach is the translation of legal text into something "computable" (i.e. amenable to processing by computers) in order to facilitate "straightforward interpretation"[47]. This corresponds to the early work from (Ashley, 2001), which suggests that for computers to undertake analytical legal reasoning, we would require mechanisms for both knowledge representation (to capture relevant aspects of legal knowledge) as well as inferences (algorithms that would allow a program to use the knowledge representations to solve problems).[48]

The semantic approach shows greatest promise in contexts that have little ambiguity.[49] An example would of such setting would be the law governing financial options - particularly the determination of the expiration date. Translating legal text into a data structure would involve parties defining a standard as to what constitutes valid, meaningful entities and relationships. In the case of the previous example regarding financial options, parties would have to agree on what constitutes an option, and what an expired date means in relation to the former. Ambiguity or divergent understanding of terms may be resolved through:[50]

1. **Inference from unstructured information** - The designers of the system can use unstructured information such as the web or other collections of text, and apply heuristics to derive the approptiate meaning. Andersson notes however that

[47]By straightforward interpretation Anderson means a method that excludes heuristics and other imprecise shortcuts Andersson, "Computational Law: Law That Works Like Software", at 11.

[48]For Ashley, knowledge representation comes in the form of a conceptual hierarchy of legal information: 1. The first level composed of cases and their facts; 2. The second level of "factors", or stereotypical fact patterns; 3. The third level with elements of legal claims; 4. The fourth level, with the legal rules and principles. Kevin Ashley et al. "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers". In: *University of Chicago Law School Roundtable* 8.1 (2001), pp. 1–28, at 2.

[49]Andersson, "Computational Law: Law That Works Like Software", at 11-12.

[50]Ibid., at 15-17.

computers cannot, on their own, start with unstructured information to arrive at authoritative answers, especially to legal questions that do not have predetermined answers;

2. **Deductions from structured knowledge** - Designers of a legal reasoning system would be required to determine in advance where the possible ambiguities lie and how these ambiguities. The problem with this approach however is that in representing the knowledge, one has to make design choices that can limit fidelity to the real world represented. Furthermore, as this encoded structure becomes more extensive and complicated - the issue can become how it can be analyzed efficiently.[51]

Within these general approaches to resolving ambiguity are intermediate approaches: 1. The deterministic approach - where the computational system arrives at a single finite state. Users can or may be required to intervene during intermediate stages. While this may simplify computational legal reasoning, it detracts from the ultimate goal of making law behave more like software. 2. On the other hand, a non-deterministic approach is purer and more practical. This involves comparison of the facts (with their attendant ambiguity) with precedent, an automated identification of strong and weak points of argumentation, and then presenting all alternatives in the final output. The user will only intervene at the final stage by adjudicating between alternatives presented.[52]

Recently, open standards based on the eXtensible Markup Language, such as Legal XML, have been adopted by the legal technology community to facilitate the encoding of legal texts into machine-readable formats.[53]

Syntactic approach In the syntactic approach, legal concepts are transformed into data structures which are then subjected to symbolic manipulation. The emphasis of this

[51] Andersson, "Computational Law: Law That Works Like Software", at 17.
[52] Ibid., at 17.
[53] Ibid., at 20.

approach is in trying to mimic the process of reasoning, rather than the content of the law.[54] The approach is more heuristic, and less reliant on the formal structures of the law. The syntactic approach is more flexible, and can be used in a wider range of contexts. However, it is also more prone to errors, and may require more resources to implement. The syntactic approach can be most appropriate in areas of the law that involve complex analysis of facts, subject to interlapping rules.[55]

Legal texts need to be translated into formal representations. Engineers will need to ensure that formal representations accurately reflect informal legal texts, even as the legal regime is constantly evolving. If not, we have a state of incongruity.[56]. We can try to resolve incongruities through the following approaches:[57]

1. **Harmonization** - Refers to any strategy for ensuring congruity after finalization of the legal text. This can involve post-facto interpretation or amendment to ensure that the electronic representation fits the legal text. This approach may be considered a temporary bridge for the current mass of legal text, to be abandoned in the long term in favor of unification;

2. **Unification** - Involves rewriting laws in a way that is syntactically coherent for computers. Thay is, bound by the same formal rules as a programming language.

Based on the limitations of the above approaches, Andersson asserts that *post-hoc* disambiguation is intractable. Instead, he proposes the use of specialized tools to prevent ambibuity *ex-ante*, by 1. imposing constrainsts on how users construct legal documents; and 2. translating these constrained choices into machine readable format.[58]

[54]Andersson, "Computational Law: Law That Works Like Software", at 17-18.

[55]An example would be cause of action analysis for copyright infringement, which would involve separate determinations for: 1. Copyright eligibility; 2. Exceptions and affirmative defenses. ibid., at 18.

[56]prenote ibid., at 18-19.

[57]Ibid., at 19.

[58]For example, the authoring tool for financial contact will require an expiration date and price, with input validation constraints (e.g. expiration date must be a date in the future, and price must

A language for representing rules computationally must satisfy the following requirements: 1. It should accomodate semantically-rich rulesets even as it formally models behavior; 2. It should be grounded in logic, since we are concerned with performing automated analysis of the rules; 3. Finally, it should be capable of representing sequences of actions, and behavioral constraints on those action.

2.2.3 Relationship with Current AI Implementations

Computational Law appears to overlap with artificial intelligence in terms of function. Explicitly encoding the rules of law and legal reasoning can be considered an application of **declarative artificial intelligence**, an approach to AI that focuses on the representation of knowledge and reasoning. Recent developments show great promise from the connectionist approach to AI, which focuses on the use of neural networks and deep learning. The latter approach has been used to develop large language models (LLMs) such as GPT, which can perform a variety of tasks, including generate text that reads like plausible legal reasoning. Given this state of affairs, will it not be better to develop LLMs to perform the tasks of legal analysis and evaluation? One might suggest simply feeding ChatGPT the corpus of existing law, and expect it to perform legal reasoning. In some ways, the declarative approach is similar to the training of a machine learning model to perform the same function. The difference is that the former is a more explicit, transparent, and controllable process. The latter is more opaque, and the results are less predictable. Thus, for mission-critical domains like law, there is merit in explicitly encoding rules over LLMs and deep learning for the following reasons:

No ground truth In an ordinary conversation or task, humans rely on an internal theory of the world, a theory of mind for the entities that it is dealing with. We have a phenomenology. We are still not sure if this can be done for AI's. Certainly this is not what happens with LLMs. These models do not have ground truth. Without the capacity

be a positive number with a currency.) Andersson, "Computational Law: Law That Works Like Software", at 16-17.

for ground truth, LLM's can spiral into delusions - making their applications unsafe for mission-critical applications.

Prohibitive costs LLM's are expensive - These systems are expensive to build, train, and maintain, even on a per-query basis once everything is set up. These are also expensive to retrain. If a Large Language Model gets "poisoned" by malicious input - what are the ways to mitigate it? How can one "nudge back" if the mechanism is hard to trace and hard to explain. Fixing it will require a lot of resources and the solution might not be durable (It could also affect the accuracy and the responsiveness of the model)

The connectionist approach to AI and declarative computational law approaches, although distinct, can still converge and reinforce each other: Data-driven LLMs can inform and enrich our techniques for encoding laws. The more varied the laws we have to encode, the more diverse the rules that can form the basis of a computational law system, and the more accurate and relevant will its determinations be. At the same time techniques of logic used by computational law can enhance algorithms used by the data-driven approach by providing a more nuanced view of legal knowledge and legal reasoning It may be possible to combine LLM's with the declarative approach in mutually beneficial ways: Wolfram suggests that a sufficiently trained machine model can interact with norms defined in a symbolic discourse language: Overall goals and standards can be defined in the symbolic discourse language, while the machine learning model can fill in implementation details.[59]. Machine learning models can also be trained to convert a huge corpus of legal texts into a initial encoding in the symbolic discourse language.[60]

[59] Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution", at 165-166.

[60] Wolfram believes that machine learning models are likely to have limits in how they model concepts (such as the notion of space). Thus, human intervention will always be required in encoding laws and norms ibid., at 157.

2.3 Computational Law Examples

Let us imagine a business with the following configuration of employees and offices:[61]

John manages Ken	John is in office 22	John is male
John manages Kat	Kat is in office 24	Jill is female
Jill manages Mark	Ken is in office 22	Ken is male
Jill manages Mike	Kat is in office 24	Kat is female
		Mary is female
		Mike is male

Logic and programming constructs allow us to: First - use of variable to represent an arbitrary number of entities (X, Y, Z for employees and offices). Second - use of logical operators to express relationships between any of the above (not, and, or, if-then). These ”representational extensions” allow us to define new relations in terms of existing relations:

> If X is in office Z and Y is in office Z and X and Y are *distinct*, then X is an *officemate* of Y.

In addition to merely describing entities and their relationships, we can encode rules and regulations through the use of these programmatic tools. We can ascribe the attribute of illegality to some facts or relationships:

> If X manages Y and X is an officemate of Y, then that is *illegal*.

Within a given set of facts (entities and relationships) and rules (deontic assertions) it may be possible to derive other conclusions:[62] These patterns of reasoning are called ”inference rules” or rules of inference. Iterative use of inferential reasoning can generate all logical conclusion (facts and rules) from within a given set of premises (facts and rules).

[61] The following examples are from Genesereth, *Computational Law: The Cop in the Back Seat*, at pp. 3-5.

[62] According to Genesereth, 2015: ”...by matching facts and conclusions of rules to the conditions of other rules and asserting their conclusions”

Example of inferential rules discovery and compliance check:

- John is in office 22
- Ken is in office 22
- John is an officemate of Ken
- John manages Ken and John is an officemate of Ken
- John is not Ken
- That is illegal

We can invert this reasoning working backwards and arranging facts to avoid illegalities

- John is in office 22
- Jill is in office 24
- Ken is in office 22
- Kat is in office 24

The inference rule discovery process can be extended to look for inconsistencies within a set of regulations. For example: We might require every project to have managers and subordinates, and no manage have a subordinate who is also an officemate. This might be inconsistent with a subsequent rule requiring special projects personnel be housed in a common work room. Compliance checking (through automated legal reasoning) can feed legal planning and regulatory analysis.

2.4 Advantages of the Computational Law Approach

Computational law can cover a range of possible uses cases, often derived on the promise of being able to automate legal deteminations at a higher quality and less cost - without the human biases and limitations of human functionaries.[63] Based on the above definitions of computational law, and depending on the horizon of techological development considered, possible computational law implementations fall into two major categories:[64]

1. **Specific computational law** - Such as simply confirming the presence of necessary elements of a cause of action, as in a checklist.
2. **General computational law** - Capable of making nuanced determinations if presented with a complex fact pattern within a specific (or even several) legal regimes.

2.4.1 Enabling applications for automated legal reasoning

Rule/argument generation Computational law could lead to the development of applications that are capable of causal inference in law. Assuming facts and rules can be well-defined, a computational process can derive other applicable rules.[65] The availability of rule detection and automated legal analysis can enable legal self-help - actors structuring/planning their activities (especially electronic transactions) to be legally valid/compliant. Similar to word processors reducing reliance on typesetters.[66] Logical representations can make it possible to derive common baseline rules, or discover bridging rules (or the exact points of divergence). This will then make it easier to have analyze cross-border contracts, or do comparative legal analysis.[67]

---

[63]Simon F. Deakin and Christopher Markou. "From Rule of Law to Legal Singularity". In: *Is Law Computable? Critical Perspectives on Law and Artificial Intelligence*. Oxford: Hart Publishing, 2020, pp. 1–29, at 5.

[64]Andersson, "Computational Law: Law That Works Like Software", at 6.

[65]The simplest and oldest attempts at computational law applications often involve the mapping of legal rules into logical rules. This approach can be useful if the problems are stereotypical and clear, i.e. there is little to no context dependency that will make the application of rules contingent. Branting characterizes UCC-related problems as those most likely to be amenable to the approach. While those that involves broad standards, such as "reasonable care" is not Ashley et al., "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers", at 14-15.

[66]Genesereth and Love, "Computational Law", at 206.

[67]But see Benjamin Alarie. "The Path of the Law: Toward Legal Singularity". In: *University of Toronto* 66.4 (2016), pp. 443–445. ISSN: 1556-5068. DOI: `\https://doi.org/10.3138/UTLJ.4008`, at 1-3. The following discussion on capabilities and applications of Computational Law systems may be considered modest, especially when compared to Alarie's vision of a "legal singularity"

Legal outcomes prediction Related to the generation of other feasible rules is the prediction of legal outcomes. Given a state of affairs, sufficiently described, a computational law system can determine the legal consequences that are likely to follow from factual and legal premises.[68] A computer can treat factual circumstances in the present as data, while the applicable rules can be represented as algorithms that can process such data to determine likely results. Predictive systems may be adopted by legal practitioners, since advising their clients may often involve predicting the outcome of legal controversies. Having computational systems that look at the data from an uninterested perspective may be helpful since lawyers' calculations may be skewed by their optimism, or by an overestimation of their own skills. This can result in suboptimal outcomes for their clients, the courts, and society as a whole.[69]

Document processing Finally, computational law systems can serve the requirement for the drafting, preparation, and filing of legal documents. The ability to infer rules and predict outcomes can be combined with exiting sophisticated models (such as those provided by natural language processing) in order to create drafts.[70]

All of the above applications can be enabled by a body of formalized legal knowledge and the algorithms to process such knowledge. A computational model of the law can be serve the use cases of different stakeholders in different settings.

---

brought about by greater computational capabilities and availability of data. Using tax law as an example, some of the transformations brought about by this convergence include: 1. Improved dispute resolution and access to justice - A shift fr om standards (broad, adjudicated ex post facto) to a more complex but query-able system of rules (that are knowable ex-ante), 2. More complete specification of tax law. The emergence of a more complex regime that is nevertheless capable of precision, coherence, and distribution of burden (at least compared to the current system) Legal uncertainty can be eliminated under such a regime, and legal disputes will be rare. Agreed upon or discovered facts can be readily mapped to clear legal consequences.

[68] Robert Kowalski and Marek Sergot. "The Use of Logical Models in Legal Problem Solving". In: *Ratio Juris* 3.2 (July 1990), pp. 201–218, at 203-204.

[69] Ashley et al., "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers", at 15-16.

[70] Ibid., at 16.

2.4.2 Appropriate settings for computational law

The availability of rule detection and automated legal analysis can enable legal self-help - actors structuring/planning their activities (especially electronic transactions) to be legally valid/compliant. Similar to word processors reducing reliance on typesetters.[71] Logical representations can make it possible to derive common baseline rules, or discover bridging rules (or the exact points of divergence). This will then make it easier to have analyze cross-border contracts, or do comparative legal analysis.

Love and Genesereth maintains that such systems and self-help only extends to reducing transaction costs for legal compliance and does not mean that parties can appear *pro-se* in instances of conflict. The forum of computational law is within enterprises, and not courts.[72]

Genesereth sees potential in embedding computational law applications into software that supports workflows that are subject to legal, regulatory requirements - e-commerce, data privacy, etc. Genesereth points to Project Calc (A Stanford CodeX project under Harry Surden), which integrates into CAD software used by architects, routines for checking compliance with rules such as: building codes, environmental rules, accessibility laws.[73]

We can embed computational law applications in devices such as cellphones, car dashboards, smart glasses so that they can provide legal guidance at the point of decision. For example: An app that not only identifies the flower the picture of which you took, but also informs you that you should not pick it up. Compared to simply publishing an overwhelming mass of laws (often in a language inscrutable to the public) digitally-mediated legal

[71] Genesereth, *Computational Law: The Cop in the Back Seat*, at 7.

[72] See. Citing Loftus and Wagenar: "Optimism is rewarded...The most successful trial lawyers are thos whose estimates are least realistic, that us, are most overly optimistic...This means that as an institution, courts are rewarding behavior that isn't optimally beneficial to the system as a whole... Genesereth and Love, "Computational Law", at 206.

[73] Genesereth, *Computational Law: The Cop in the Back Seat*, at 6-7.

determinations can help make the notice requirement of due process more meaningful.[74] For automobiles (whether manned and unmanned), in addition to basic functions such as navigation and collision avoidance, the system can help compliance with legal requirements such as: a. speed limits; b. whether or not a street is one way c. whether u-turns are allowed; d. what areas allow parking.[75]

Different systems may be required for different participants in the legal system, with sophistication and capability scaling to the requirements of users along this continuum. Ordinary users may only need answers for simple scenarios. Lawyers may require argument generation based on legal premises and factual scenarios. Judges can use similar systems, but for evaluating the basic validity of arguments and precedents.[76] Finally, legislators and policy makers can use computational tools in order to evaluate proposed rules against other norms, as well as predict the impact of draft laws.[77]

Other applications include: Enterprise-wide monitoring and automated compliance; simulation of impact of rule changes; automated rule changes based on specified end goals.

2.4.3 Consistency and predictability

The utility that can be derived from the computational approach is compelling enough to warrant its pursuit. The more obvious advantages come from the speed and reliability of

[74] Genesereth, *Computational Law: The Cop in the Back Seat*, at 7.

[75] Ibid., at 7.

[76] Kowalski imagines an example where the encoding of social security laws can be turned into an application for government officials overseeing applications, taking into account the office's internal logic for evaluating claims. At the same time, the encoding can be used for a general version of the application available to the public, using legal knowledge to advise them with their applications. Kowalski and Sergot, "The Use of Logical Models in Legal Problem Solving", at 208-209.

[77] Ashley et al., "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers", at 14. The requirements for legal professionals can be further broken down to the following functions: 1. Problem formulation - Formulate the problem in terms of the relevant legal concepts, 2. Retrieval - Gather authorities relevant to the problem as formulated, 3. Problem analysis - Determining the legal consequences that follow from application of authorities to the facts. 4. Prediction - For each of the possible outcomes borne by the analysis - what are the probabilities of each outcome?

computers, as well as their ability to retrieve relevant legal text from memory.[78]. However, some of the more fundamental advantages to the profession can be indirect: The rigorous structured approach of these systems may "mold the thought processes of the lawyer" (and law students) into a more logical pattern, and the extended use and design of such systems will force legal scholars to confront and resolve the ambiguities of the law.[79] Investigations into computerized encoding and analysis can be useful not just for the development of practical application but also for clarifying and improving the process of legal reasoning.[80] Genesereth argues that simply publishing the overwhelming mass of laws, in a form inscrutable to the public is not adequate notice. Computational law, by providing digitally mediated legal determinations can help address this gap.[81]

Representing and analyzing laws with the computational approach can provide certain advantages. It can remove or minimize the degree of legal uncertainty (characterized by radicalization of legal realism, or postmodernism), and make law more transparent and consistent. The casting of law within a formalism can enable advanced analysis that goes beyond subjective inferences of human lawyers. Advanced analytical tools such as simulations, derivations, combinatorics can be applied to bodies of law. Finally, lawyers and legal scholars can have a stable point for discussion, without the ambiguity of language across jurisdictions. This can be a basis for interdisciplinary work, as well as a basis for testability and confirmation.

Wolfram notes that at the immediate level, the conversion of legal constructs into the computational form can give them new capabilities, such as automated annotation of implications, simulation of results, statistics and probability analysis.[82] On the other hand,

[78]Grossman and Solomon, "Computers and Legal Reasoning", at 66.
[79]Ibid., at 66.
[80]See Kowalski and Sergot, "The Use of Logical Models in Legal Problem Solving", at 217.
[81]Genesereth, *Computational Law: The Cop in the Back Seat*, at 8.
[82]Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution".

lawyers and law students can think about these legal constructs at a higher level.[83]. It gives rise to to clearer thinking about the law - without the semantic ambiguity, cultural baggage of natural language. Wolfram paints the broader implications of the technology by historical analogy: With growing literacy and the development of technology around the written word - there is a growing trend towards complexity of transactions and their corresponding legal instruments. Having a computational component will lead to even greater levels of complexity.[84]

2.4.4 Economic considerations

Beyond the technical feasibility of these systems, and the intellectual curiosity they may inspire - is there sufficient motivation and necessity for the development of computational law systems? Branting predicts that these systems are needed due to a vast, unmet demand for legal services, particularly in the growing government sector - which will need to navigate an ever more complex legal and regulatory regime in order to make the routine legal determinations necessary to carry out its functions.[85].

Building these systems do not mean starting from scratch, since we can leverage existing data on systems that embody business rules, such as those used in banking or human resources.[86] Computational law just extends this tendency by encoding public instead of private rules.

[83]Wolfram cites the Sapir-Whorf hypothesis - that is, language can affect patterns of thinking Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution", at 164.

[84]Ibid., at 165.

[85]Ashley et al., "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers", at 16-17. Branting also suggests that these systems can be a form of marketing for legal expertise, i.e. software can handle low-end requirements and lead clients to human legal experts for bespoke work.

[86]Genesereth, *Computational Law: The Cop in the Back Seat*, at 7.

2.5 Limitations of the approach

As will be discussed below, there are both fundamental limitations to computation, as well as policy reasons not to employ it in law. Thus, not all of legal reasoning may be subject to translation to a computational model. Instead of a outright substitute to legal reasoning by human experts, computational law is proposed as an aid to a subset of tasks such as those mentioned above (e.g. authority retrieval, argument generation, analysis and prediction).[87] The computational approach is often limited by the following:

1. **Open-texture problem** In the real world where lawyers operate, both the rules and assertions of facts may be open to interpretation.

2. **Incongruity with actual legal thinking** Legal decisionmaking seems to bypass explicit reasoning around rules and derive from specific cases, often through analogy.

3. **Incompleteness** Formalization can only provide a finite set of rules with which to analyze complex states of the world as well as its normative environments

Open texture One fundamental problem with computational law is how to square formalisms with the open-texture of the law: The complexity of the law (and the world it operates in) means that the facts and rules that one wants to encode in a categorial manner will be open to intepretation. Genesereth provides the example rule: "No vehicles in the park". This might be obvious to a human in the community, but problematic for someone trying to define the rule. What is a "vehicle"? Is a bicycle a vehicle? How about a skateboard? Roller skates? What about a baby stroller? A horse?[88] Genesereth's suggested response to the open-texture problem is to limit computational law applications to cases where such issues can either be 1. externalized - that is, allow human users to input their judgments on open-textured concepts, through data entry or on-the-fly determinations; 2.

[87] Ashley et al., "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers", at 14.

[88] Genesereth, *Computational Law: The Cop in the Back Seat*, at 6. For this matter - What is "the park"? What are its horizontal (and vertical) borders? Can a helicopter hover at ten feet? One hundred feet?

marginalized - simply do not use the computational approach in areas of law where there are many open-textured concepts.[89].

The programmatic approach (mapping facts and constructing, deriving inferential rules) can express many types of rules. Some rules however are more complicated. Genesereth refers to prior work from Sergot and Kowalski, et al (1986) which explores the formalization of the British Nationality act as a logic program, through conversion of a text into Extended Horn Clauses.[90] However, some legal texts are not readily formalizable with this approach. Such as: 1. When the applicable rule will depend on a person's subjective belief about the facts/ (e.g. if "...the Secretary of State is satisfied that...) 2. Some rules are dependent on default states that can change under some circumstances, such as contrary evidence (e.g. "...unless the contrary is shown...") and 3. Rules that require reference to other parts of the law, or other laws.

Incompleteness Complementary to the problem of open-texture are fundamental limitations to formal, logical approaches. The limits of formal reasoning means that one will not be able to generate enough explicit, categorical rules for resolving the terms of a legal problem.[91]

While these problems might be insurmountable in some legal domains, Love and Genesereth argue that domains where transactions are electronically mediated can make the problems of computational encoding and analysis more manageable. These systems can be considered more amenable to the computational law approach since: Like other legal domains, have entities and transactions that are subject to a system of rules (statutes, regulations, policies). The transactions in these systems are semantically rich - they are

[89] Genesereth, *Computational Law: The Cop in the Back Seat*, at 6.

[90] "A person born in the United Kingdom after commencement shall be a British Citizen if at the time of birth his father or mother is (1) a British citizen or (2) settled in the United Kingdom" ibid., at 5, citing Sergot and Kowalski, et al (1986).

[91] Ibid., at 6.

well-defined through documentation, code, or system constraints (they also note the industry's move towards semantic data) The information gap problem (when it comes to factual determinations) - is also addressed in these domains, since within these systems, each transaction (and agents involved) can be logged and verified.Finally, these domains are also the most likely users and beneficiaries of computational law systems.[92]

Incongruity with legal reasoning Another possible limitation of the computational approach is that not all legal reasoning is characterized by the formal logical methods employed in programming. As aptly put by Edwina Rissland, et al.: "Law is not a matter of simply applying rules to facts via modus ponens". Many legal determinations are not made from deducting from general principles but inducing from specific cases.We can't map enough deductive rules from a given body of law. Genesereth maintains that since computational law emphasizes deductive reasoning, it cannot be applied to instances of legal determinations that require analogic or inductive reasoning.[93] The use of analogical reasoning is a special problem for computers, since it will require discovering (or event constructing) the relevant principle that establishes that the cases are "similar". While computers can exhaustively search through a given set of predetermined rules that can establish similarity.

Other obstacles to formalization can arise from the ways law is formulated in the first place: 1. Legislation is not always coordinated, since they arise from different contexts (e.g. different historical settings that confront different problems)2. Legislation has gaps - some entities, actions, relationships, are not covered by any rule 3. Legislation may overlap, or be inconsistent with each other.Genesereth is convinced, however, that since the publication of Sergot, et al., many of the difficulties presented have been overcome by extensions to the language and reasoning of computational law.[94]

[92]Genesereth and Love, "Computational Law", at 205-206.
[93]Genesereth, *Computational Law: The Cop in the Back Seat*, at 6.
[94]Ibid., at 5.

Finally, there is some doubt as to whether or not computation can adopt the kind of analogical reasoning often used in legal interpretation. Analogy does not involve merely enumerating similarities from a given set of criteria. Reasoning by analogy does not proceed from premise to conclusion, but is based on the discovery (or even creation) of evaluative principles from which one can assert that one case is similar to another.[95] The search space for such principles may be infinite, given that humans can invent new ways to draw similarities between one category and another. The ability to discover new analogies can also be based on the human experience of being embodied, sensate, and embedded in a culture - attributes that a computer may never have.

2.6 Other countervailing factors

Institution will require additional expertise, as well as resources to fund the development costs of these systems. At the same time,lawyers are not likely to adopt systems that will reduce time billings (but may do otherwise for task based billing).[96]

2.7 Conflict with legal realism

Computational law's philosophy contrasts with the notion of Legal Realism. In its stronger formulation, legal realism means that the text of the law doesn't matter, or at least does not matter as much as other considerations, in order to perform a balancing of interests (usually based on factors extraneous to law) on a case-by-case basis.[97] Computational law may not be able provide this kind of normative flexibility.[98] Instead, it is more closely aligned with Legal Formalism. Thus it carries the notion that laws are definitive,

[95] Ashley et al., "Legal Reasoning and Artificial Intelligence: How Computers "Think" Like Lawyers", at 19-20.

[96] Ibid., at 17.

[97] In its extreme formulations, legal realism can go against the project of building a rules-based society. The author also has more practical objections: If we are not in the business of building and then recognizing enduring legal norms, then we are wasting our time teaching our students legal research and statutory interpretation. Better to instruct them on the non-legal mechanisms that actually shape decisions, such as economic interests and individual psychology.

[98] Genesereth, *Computational Law: The Cop in the Back Seat*, at 5.

and exhaustively account for all the preferences and calculations of the legislator.

Given its alignment and limitation, Genesereth suggests that computational law is most relevant to civil law jurisdictions - where the text of the law are interpreted literally or with very constrained space for interpretation. In contrast, it is least relevant in common law jurisdictions marked with on-the-fly legal innovation through judicial interpretation.[99] Although computational law has limits when applied to cases that require analogical or inductive reasoning (which often characterizes the reasoning in judge-made Laws), Genesereth suggests that the judicial process itself can generate categorical constraints from vaguely worded statutes. Judicial law can be a source of encoded rules.[100] Even in common law jurisdictions, however, there are categorical, codified statutes that may not be subject to significant judicial discretion. Examples include legislation on data privacy, securities, enterprise management, construction, electronic commerce, taxation. There is a growing tendency in these fields of law to move toward greater textual specification and codification. This makes them more amenable to the computational approach.[101]

[99] Genesereth, *Computational Law: The Cop in the Back Seat*, at 5.

[100] Ibid., at 6.

[101] To a certain extent, the end goal of the adjudicatory process is to come up with categorical interpretations of existing statutes. One can consider rules expressed in judicial decisions as expressed in judicial decisions as extensions of the legal text, and encode them computationally, as if they were part of the original statute. So to the extent that statutes are considered vague in a common law jurisdiction, judicial decisions can supplement them by coming up with interpretations which can be encoded. Genesereth is also convinced that as Computational Law becomes more useful, legislators and regulators will be encouraged to have more such categorical laws ibid., at 6.

3 Law as a Computable Structure

3.1 The nature of computability

The premise of computational law is that once we have both rigorous formal representations of law, and the appropriate logical methods to analyze them, law becomes computable. What is meant by a computable approach, or the computability of legal determinations? The formal meaning of a problem or a domain's computability relates to whether or not it can be solved through an algorithm. In other words, a problem is computable if there exists a step-by-step procedure that can be executed by a computer to solve the problem.[102]

Computability also means that once we have abstracted enough of the most important attributes of a thing into a formalized model - we can map its behavior backward and forwards in time. We can access powerful shortcuts to the things behavior - to diagnose, analyze, and predict.[103] The modern world we have was achieved through computation - from bridges to bombs to games and deep space probes. These are possible because we could build models of the forces of nature, and predict their interactions through logic and mathematics.

3.2 Can the law be computable?

Wolfram argues that the computability of law can flow from the computational character of nature, from which all phenomena (including humans and human institutions from which

[102]The notion of computability is derived from Alan Turing's description of problems that are amenable to an algorithmic solution (to be carried out by a computational model such as a Turing Machine). Alan M. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society* 2.42 (1937), pp. 230–265. DOI: 10.1112/plms/s2-42.1.230.

[103]Without computability, we are confined to recording descriptions of phenomena, and we are limited in our ability to draw insights and make predictions abut a system. Similar to the state of astronomy before Newton developed the formalisms of calculus - without a proper computational model for celestial mechanics, all that could be done was observation and recording.

laws are derived)[104]: The universe itself is built on a computational foundation, and our current computational tools for representing and analyzing knowledge is the latest (and perhaps ultimate) in a series of formalisms for representing and understanding reality.[105] It is not necessary to get into such a fundamental claim. As will be argued below - it should be enough that the computational approach capture what is essential of legal knowledge. Law is not magic - it occurs within the same universe that is, to some extent, discoverable. A premise of the law as a practical profession and an academic field is that it is knowable, and that legal reasoning can be systematized. At a high level of abstraction legal processes can be modeled as if it were a computational process. We have an input (laws, evidence), and we expect an output (in the form of a decision). Formalization and computation promises to make that process purer: free from bias, fatigue, and ignorance.[106]

Of course, modeling the forces acting on a physical system is one thing, but trying to model the behavior of people and institutions under the constraint of law is a different category. As mentioned in the previous section, complexity and incompleteness conspire against us. The open-textured nature of legal concepts like "justice" means that our representations and analytical tools can only go to certain levels of description. Even if we can somehow develop a rich enough toolset to capture legal concepts, Gödel's incompleteness means that there will always be a gap in our formalization.[107]

[104] See Stephen Wolfram. *How to Think Computationally about AI, the Universe and Everything.* Stephen Wolfram Writings. Oct. 27, 2023. URL: https://writings.stephenwolfram.com/2023/10/how-to-think-computationally-about-ai-the-universe-and-everything/ (visited on 12/14/2023).

[105] See generally Stephen Wolfram. *A Project to Find the Fundamental Theory of Physics.* Champaign, Illinois: Stephen Wolfram, LLC, 2020. 770 pp. ISBN: 978-1-57955-035-6.

[106] Christopher Markou and Simon Deakin. "Ex Machina Lex: Exploring the Limits of Legal Computability". In: *Is Law Computable? Critical Perspectives on Law and Artificial Intelligence.* 2020, pp. 31–65, at 33.

[107] Gödel's theorems on the fundamental incompleteness of any axiomatic system impacts mathematics and logic, and ultimately, the capacity of computational formalism to model reality Richard P. Feynman. *Feynman Lectures on Computation.* Ed. by Anthony J. G. Hey and Robin W. Allen. Boca Raton: CRC Press, 2018. 303 pp. ISBN: 978-0-7382-0296-9, at 52.

Some problems are subject to computational irreducibility. That is, even if we can reduce a system's behavior into simple rules, it is still possible for complex behavior to arise from such systems. It may not be possible to make a prediction about a systems state or behavior past a certain point (even if the system's behavior can be modeled algorithmically). Which also means - that if you design those rules instead of discovering for yourself. There is no way to control against unintended circumstances.[108]

This difficulty does not mean that the problem will be intractable. The physicist Stephen Wolfram states that in the teeth of complexity and incompleteness, even the hard sciences are beset by oceans of non-computability. Despite all their progress in theory-making and theory-testing, scientists still have to contend with a universe that largely resists mathematical certainty. And yet, they have found enough islands of computability amidst that ocean to lay the foundations of useful things like engineering, computer science, particle physics.[109]

Our models for law will likely be incomplete and thus inaccurate. But the incompleteness of a model does not mean it will be useless. A map will never be as detailed as the territory that it guides us through, but a good map should have enough information to be useful.

A formal approach can allow smoother, more reliable collaboration and the building of higher "towers of consequences"[110] - systems that will allow more detailed study of legal systems, as well as applications for real world problems that involve the law. For example - the ability to encode legal rules into a computer program may be the key to

[108] Stephen Wolfram. "AI Law and Computational Irreducibility". FutureLaw 2023, Stanford Law School. Apr. 25, 2023. URL: https://www.youtube.com/watch?v=8oG1FidVE2o (visited on 01/14/2024), at 3:53.

[109] See generally Stephen Wolfram. *A New Kind of Science*. Champaign, Illinois: Wolfram Media, 2002. 1197 pp. ISBN: 978-1-57955-008-0.

[110] Wolfram demonstrates how a field can progress through a better formalization and encoding system: Prior to the invention of algebraic notation, problems were described through natural language text (which can be imprecise). A more formal, streamlined method made it easier to share and build off each other's ideas. Wolfram, *How to Think Computationally about AI, the Universe and Everything.*

encoding firm, normative("constitutional") limits on artificial intelligence that can still be read, understood, and edited by humans.

3.3 Confronting objections to logic in law

Computational law requires some role for logic in legal reasoning. A significant goal of computational law is the production of a computer system capable of producing legal advice (as opposed to just textual information). This can only be possible if logic has a place in law. Because if anything, a computer system's only actual capability is demonstrating a logical system.[111] Similarly, only a logical system can be computerized[112]. Even in the long term, understanding and designing AI systems involved in legal reasoning will require a background in logic, since AI applications (even those that seemingly interact through natural language), will have programming that will be undergirded by formal logic.

Lawyers have built a conceptual moat around the field of law, to distinguish it from the hard sciences, claiming that the law, unlike these fields, will always evade a reductionist, logical approach.[113] Thus, its concepts and reasoning are not amenable to computation because these are largely not computationally reducible. Since legal concepts and rules are socially constructed and in flux, they cannot be fully represented into numbers and logical constructs.The objections in the legal literature can fall under the following categories: 1. **Historical arguments**, i.e. that the legal reasoning has developed as a discipline separate

[111] Philip Leith. "Logic, Formal Models and Legal Reasoning". In: *Jurimetrics* 24.4 (1984), pp. 334–356, at 334.

[112] Ibid., at 334.

[113] Jeffrey Goldsworthy. "The Limits of Judicial Fidelity to Law: The Coxford Lecture". In: *Canadian Journal of Law and Jurisprudence* 24.2 (July 2011), pp. 305–325. DOI: 10.1017/S084182090000518X, "The popular impression of legal thinking is that it is logically rigorous. But legal reasoning, whether of judges, advocates or legal scholars, rarely has the clarity and rigour of the best analytical philosophy. Often this is because the subject-matter is simply incapable of being treated as rigorously. But more importantly, legal reasoning in real cases leads to practical decisions that have drastic effects on individual's lives or the welfare of the community, for which judges properly feel some moral responsibility. Consequently, legal reasoning can have a tendentiousness—an almost palpable gravitation towards a desired conclusion—that is lacking in the work of analytical philosophers, pure mathematicians or nuclear physicists."

from logic; 2. **Epistemological arguments**, which rely on fundamental difference between law and logic, not only in substance but in terms of subject matter; 3. Finally, there are the **Practical arguments** that relate to the applicability of logic to real-world legal problems. We confront these objections in the following subsections:

3.3.1 Historical convergence of logic and law

Law and logic during the classical period A profession as steeped in tradition and the weight of history as law may view embedding logic as an unnecessary modernist intrusion. However, the history of law is replete with examples of the convergence of logic and law. For Aristotle, law and logic were one and the same.[114]. Aristotelian logic, or what we now know as classical propositional logic, was derived from analysis and systemization of legal arguments and decisions.[115] This was carried on through the scholastic tradition, which viewed law as a system of rules which can be logically deduced from immutable principles.[116] These principles, in turn, can be discovered by man through a process of reasoning. Great jurists such as Thomas Aquinas, William Blackstone also proceeded along these lines.[117] For the longest time, logic was Aristotelian logic. One of the Aristotelian logic's central theory of the judicial syllogism, where a judicial decision is justified through

[114]Lee Lovevinger. "An Introduction to Legal Logic". In: *Indiana Law Journal* 27.4 (Sum. 1952), pp. 471–522, at 471, citing A Treatise on Government, or The Politics of Aristotle, Book III, c. 16, Elli's translation, 1943.

[115]See Wolfram, "AI Law and Computational Irreducibility", at 14:13; See also Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution", at 145. An intriguing notion propounded by Wolfram is that laws are in fact the original inspiration for logical and mathematical systems. Legal arguments served as the model for the axiomatic approach to geometry defined by Euclid. Later, in the development of scientific thought, the discovery of "natural laws" were viewed as similar to legislation, i.e. These define constraints from God (or nature) instead of a human lawmaker.

[116]See Karlheinz Hülser. "Proculus on the Meaning of OR and the Types of Disjunction". In: *Past and Present Interactions in Legal Reasoning and Logic*. Springer International Publishing, 2015, pp. 7–30, at 8. Emperor Justinian's *Digestae*, in the chapter *De verborum significatione* (On the meaning of words), contains reference to the Letters of Proculus, a distinguished Roman jurist.The passage quoted from Proculus covered his discussion on logical disjunctions (OR). The fragment from Proculus is itself derived from a long tradition of adopting concepts from Stoic logic. Through its adoption in the digests, it continues to inform modern statutory interpretation.

[117]*Ibid.*

a form of syllogistic reasoning, i.e. as an inference from normative and factual premises.[118] This form of legal determination has arguably shaped the notion of separation of powers (i.e. between legislation and adjudication): The legislative creates law as a set of legal norms, and the judge will need to reason through these premises in order to apply them to a particular set of facts.[119]

Law and logic during and after the Renaissance A cornerstone of the 17th century naturalist doctrine (Grotius, Salamanca School, Espinoza) is that the principles of law should be systematized through mathematical methods. Efforts to both define and systematize characterized legal studies and there was the view that certainty of the law was attainable.[120] A crystallization of these ideas can be found in the recently rediscovered works of Gottfried Wilhelm Leibniz. Leibniz is commonly known a a leading figure in mathematics and philosophy. However, before his seminal work in those fields he was a lawyer and a promising legal scholar. His work combines law and philosophy, and proceeds from the premise that some of law's fundamental questions cannot be answered without philosophical thought.[121] Leibniz insisted that law should have a "philosophical basis", without which the law is bound to be an "inextricable labyrinth".[122] His forays into philosophy and law seems to be partially motivated by his numerous attempts at reconciling church doctrines (Protestants v. Catholics), conflicts over which led to the Thirty Years war that

[118] Pablo E. Navarro and Jorge L. Rodríguez. *Deontic Logic and Legal Systems*. New York: Cambridge University Press, Sept. 29, 2014. ISBN: 978-0-521-76739-2. DOI: 10.1017/CBO9781139032711, at ix.

[119] Such a separation of functions assumes law has logical attributes such as: 1. Completeness - that there is always an applicable legal norm that can solve any dispute; 2. Consistency - that there are no incompatible norms applicable to the same case. Judicial decisions rely on at least one of these holding true. ibid., at ix.

[120] Alberto Artosi and Giovanni Sartor. "Leibniz as Jurist". In: *The Oxford Handbook of Leibniz*. Ed. by Maria Rosa Antognazza. Oxford University Press, Dec. 27, 2018, pp. 640–663. ISBN: 978-0-19-974472-5. DOI: 10.1093/oxfordhb/9780199744725.013.38. URL: https://academic.oup.com/edited-volume/34667/chapter/295400716 (visited on 10/10/2023), at xviii.

[121] Matthias Armgardt. "Leibniz as a Legal Scholar". In: *Fundamina* (2014), pp. 27–38, at 28-29, citing *Specimen quaestionum philsophicarum ex jure collectarum, 1664*.

[122] Note that Leibniz was refering to Philosophy in its broader, classical sense, which includes logic and mathematics Artosi and Sartor, "Leibniz as Jurist", at xx.

destroyed Germany. His works on legal reasoning have only been translated and published recently, indicating that he pursued a mathematical-logic approach similar to modern ideas in computational law.[123] Leibniz's first legal dissertation, *Disputatio juridica de condetionibus* used propositional logic, modal logic, and probability logic to the law on conditions, a technical problem under Roman law.[124] His writings indicate that this direction was inspired by classical sources, which requires that law, as the "science of the just and unjust", be built on "the awareness of human and divine affairs".[125]. Leibniz's interest in Roman Law as the basis of a rational legal system is the view (shared by other jurists) that the Roman law tradition is more accepting of the convergence between law and science. Roman law is said to take into account "the working of nature" in order to produce sound and equitable decisions.[126]

The three underlying ideas of Leibniz's legal investigations are:[127]

1. Legal research and problem solving, particularly adjudication requires an interdisciplinary dialogue. The law needs to accept ideas from other disciplines such as philosophy, logic, theology, mathematics, and physics.

2. Law also needs to have an intradisciplinary dialogue, i.e., between the various schools of legal thinking.

3. Law requires a more diverse range of reasoning methods and cognitive tools. Practitioners can select the appropriate tool based on pragmatism, i.e. their effectivity in solving legal problems.

[123]Leibniz'a view on legal certainty rests partly on similarities between geometry and jurisprudence: "Both have elements and both have cases. The elements are simples (simplicia); in geometry figures, a triangle, circle, etc; In jurisprudence an action, a promise, a sale, etc. Cases are complexions (complexiones) of these, which are infinitely variable in either field." Artosi and Sartor, "Leibniz as Jurist", at xxv.

[124]Armgardt, "Leibniz as a Legal Scholar".

[125]Artosi and Sartor, "Leibniz as Jurist", at 5 citing Ulpian, D.1.1.10.2, *De justitia et jure.*

[126]Ibid., at 6.

[127]Armgardt, "Leibniz as a Legal Scholar", at 5.

Leibniz believed that no case, no matter how apparently perplexing, is insoluble *ex jure*. Thus, he applied logic to confront legal puzzles from the classical era, arriving at a classification scheme for apparent and actual legal conundrums and the appropriate analytical device to solve them:[128]

1. Cases of apparent conflict between law and philosophy (which during that time included metaphysics, mathematics, empirical sciences, theology) that often arise from the same terms (but with different meanings) used in law and philosophy.

2. Questions that arise from the assumption that a principle is of universal application, but is in fact justifiable under particular pragmatic conditions, or simply the result of defects in the underlying conceptual frameworks used by lawyers and jurists.

3. Problems that arise from the lack of a deeper logical analysis of a conceptual issue.

4. Actual legal puzzlesm which are cases of doubtful solution because of the convoluted form of dispositions (expressions of intent), or conflict with a priority relationship.

Leibniz himself acknowledged that reasoning through legal problems will require more than propositional logic, since such problems involve uncertainty, possibility, and the passage of time. Although Leibniz's efforts to develop a logical formalism was not successful, these ideas, inspired systems of logic and continue to animate the field of computational law.[129]

The challenge of legal realism The most potent historical challenge to the notion of identity between logic and law comes from Justice Oliver Wendell Holmes Jr.[130] Legal scholars continue to cite this epigram as an embodiment of the school of legal realism: "The life of the law has not been logic, it has been experience".[131]

[128]This position also made him wary of judicial discretion Artosi and Sartor, "Leibniz as Jurist", at ix, xxi.

[129]Ibid., at 11.

[130]See Lovevinger, "An Introduction to Legal Logic", at 472.

[131]Holmes, The Common Law, p. 1, 1881.The full quotation is as follows:"The life of the law has not been logic: it has been experience. The felt necessities of the time, the prevalent moral and

Courts and advocates in the Philippines have cited this quote from Holmes, often without its full context to the point that it has become a slogan, or the legal equivalent of a meme. It can be invoked to defeat a clear interpretation of the law on linguistic and rational grounds in order to introduce extraneous considerations. However, the reflexive invocation of this epigram in order to frustrate the application of logic is misleading.

If one were to read the rest of Holmes' work, one would realize that Holmes was not dismissing the role of logic and rational thinking in law. Instead, Holmes was urging us to include more inputs into what is still a logical process of making a legal determination.

In objecting to what he called "the fallacy of the logical form", Holmes:

1. Acknowledges that as a phenomena contained in the same universe as physical matter, law is ultimately subject to the same underlying rules, such as causation (otherwise, it would be a miracle);

2. Acknowledges that logic permeates through the practice: "The training of lawyers is a training in logic" - since it involves building familiarity with logical tools like analogy, discrimination, and deduction. Holmes also characterizes judicial decision as expressed in the language of logic.

Thus, Holmes objection, and the actual divide between "natural law" and legal realism is not whether or not logic should be applied at all, but to what materials logical processes should work with. For the "natural law" school, they believe that there are transcendent basic principles which can be grasped intuitively, or derived through deduction. On the other hand, "legal realists" reject *a priori* transcendent rules and emphasize an inductive approach from empirical data (or experience).

political theories, intuitions of public policy, avowed or unconscious, even the prejudices which judges share with their fellow-men, have had a good deal more to do than the syllogism in determining the rules by which men should be governed. The law embodies the story of a nation's development through many centuries, and it cannot be dealt with as if it contained only the axioms and corollaries of a book of mathematics..."

Hawkins, through a historical and textual analysis, argues that the statement was never meant as a practical guide for legal reasoning, or the interpretation of constitutional or statutory law. Instead, it is a descriptive view of the development of the common law. The "logic" that the statement describes as being eschewed by the common law tradition is not logic as academically understood or colloquially known, but refers to the "vain attempt to impose consistency on intuitively developed law".[132] To the extent that Holmes's words can serve as a foil to the application of logic in law, Hawkins finds that there is ambiguity as to the scope of his objections, and thus its actual application in a legal: Is it that logical reasoning has no place in law - that lawyers and judges should embrace irrationalism or intuition? Or perhaps, more realistically - was Holmes merely asking for a counterweight against excessive legal formalism?[133] Furthermore, if "experience" defines the content of the law - what constitutes this experience. More pointedly - whose experience matters?

It should also be noted that logic has evolved from Holmes' schoolboy days, when most likely education would only cover classical propositional logic (or syllogistic logic as originally systematized by Aristotle)It can be conceded that classical propositional logic, as formulated during Holmes' time, the logic that most of us are aware of (and the one usually employed in programming) is not the most appropriate tool for representing legal rules. Subsequent sections will discuss the more appropriate logical systems for representing legal rules, such as deontic logic and defeasible logic.

3.3.2 Epistemiological unity between law and logic

Objections to logic often point to a fundamental difference not just in method (structured, formal versus discursive and intuitive), but also to their subjects. The basis of logic was the assumption that valid argument can be based upon the elemental form of the proposition, composed of a subject and a predicate linked by a connective. Any proposition, meanwhile

[132] See generally Brian Hawkins. "The Life of the Law: What Holmes Meant". In: *Whittier Law Review* 33 (Winter Issue 2012), pp. 323–370.
[133] Ibid., at 325.

has a truth value - either it is true or false. There is nothing in between (the law of the excluded middle).[134]. On the other hand, legal propositions are normative rather than fact-stating, and we only have an incomplete picture of the general logic of norms.[135]

Misapprehension of "logic" Synthesizing the arguments of A.G. Guest and other legal philosophers, Summers argues that most objections of this kind is often based on a misuse of the concept of logic. Upon closer inspection, even basic logical propositions do not refer to things in nature, but concepts that may not necessarily be subject to true-or-false evaluation. [136]Summers adds that most likely, these statements are criticisms of the reasoning in particular cases, rather than general arguments against the use of logic in legal reasoning. [137] More directly, the objection can be met by referring to legal pluralism, i.e. the notion that there are other forms of logic that can be used to represent legal reasoning.[138] This includes, as will be discussed below, deontic and defeasible logics.

One problem when we discuss the role of logic in law, is what we mean by logic in the first place - is it the technical, formal sense or are we using logic in the everyday, colloquial sense? Logic in its formal sense relates to whether or not an argument's conclusions follows necessarily from the premises. The latter, "everyday logic", on the other hand, is concerned with whether any legal conclusion "makes sense" based on some informal standard.[139] These senses of the word "logic" are not related to each other. The main purpose of formal logic is to surface "possible forms of argument and conditions of valid argument".[140] On the other hand, everyday logic is prescriptive i.e., it involves the application of beliefs, (ofen grounded

[134]Leith, "Logic, Formal Models and Legal Reasoning", at 336.

[135]Robert S. Summers. "Logic in the Law". In: *Cornell Law Faculty Publications* (Paper 1133 1963), pp. 254–258. URL: `http://scholarship.law.cornell.edu/facpub/1133`, at 254.

[136]*Ibid.*

[137]*Ibid.* In cases where a decision is criticized for an "abuse of logic" (e.g. Whiteley v. Chapel), what may be at fault is the choice of legal premises, and not the (logical) manner in which the judge proceeds from premise to conclusion. Or, more often enough, it may be a problem with semantics.

[138]Leith, "Logic, Formal Models and Legal Reasoning", at 340.

[139]Ibid., at 335-336.

[140]Ibid., at 337-338.Citing McCormick (1982), The Nature of Legal Reasoning: A Brief Reply to Dr. Wilson, Legal Studies, vol. 2, no. 3 286(1982).

in social processes) as to what ought to be.[141]

Halper points out that complaints directed towards logic in judicial reasoning is often actually directed to something other than logic, such as:

1. **Belligerent precisionism** - This happens when the court takes a shortcut by interpreting a word too literally, ignoring its context, history, and the purpose of the rule.

2. **Bad faith** - It may also be the case that the court is simply being disingenuous in order to pervert the law. The use of a seeming use of syllogisms and faulty inferences, however, does not make the bad faith logical.

3. **Misapprehension of scope** - By "logic" critics may mean the simplistic notion that a few express (or otherwise deducible) rules should apply to all situations; When in actuality the rule does not encompass the situation, but is nevertheless characterized as an inconsistency of reasoning.

4. **Maintenance of contradiction** - It may be possible that the Court is accommodating contradictory rules when it upholds a new line of reasoning while allowing a previous case to remain valid.

5. **Simplistic, rote reasoning** - The Court may just be stuck in simplistic, rote reasoning in order to avoid, or exculpate itself from moral or social considerations. And "logic" is equated with this mechanism, operationalizing the fiction of the detached judiciary.[142]

On the incompleteness of formal systems Another aspect of the divide between law and logic is related to the necessary incompleteness of formal systems. The incompleteness of formal systems is a result of Gödel's incompleteness theorems, which states that

[141] Leith, "Logic, Formal Models and Legal Reasoning", at 336.

[142] Thomas Halper. "Logic in Judicial Reasoning". In: *Indiana Law Journal* 44.1 (1968), pp. 33–48, at 33-35.

any non-trivial formal system will contain statements that are true but cannot be proven within the system. This means that there will always be gaps in any formalization of the law, and that there will always be legal questions that cannot be answered through logical deduction.[143] This is a significant challenge to the idea of computational law, as it suggests that there will always be limits to what we can achieve through logical analysis. Without an overall general model for the world, representations in a formalism will always be incomplete. Wolfram asserts, however that an overall scheme is not necessary, and that it would be possible to capture concepts as needed.[144]

3.3.3 Practicality of employing logic

Logic in the adjudicatory process Another, more practical line of argument is that logic has no use for judges and lawyers, since their conclusion are arrived at intuitively, with the reasoning is arrived at *post facto*.[145] The rule deduction skeptics adopt the position that legal decisions do not arise from deduction from existing legal rules. The legal principles that supposedly guide legal reasoning are too vague and subject to so much discretion, that the operation of logical processes is not possible.[146] To this, Halper points to fields of law, (such as real property law) that are devoid of any emotional or intuitive notions, through which systematic generalizations can be derived.[147] The non-logical intuition may thus be based on a judge being so steeped in the deeper overall logic of the law, and it only seems intuitive since he reaches his conclusions first and then justifies them later. Logic may also play a role in how a judge evaluates a particular proposition (whether or not such proposition was arrived at logically or intuitively), in that the judge reasons through the

[143] Rebecca Goldstein. *Incompleteness: The Proof and Paradox of Kurt Godel.* New York, London: Atlas Books, 2005.

[144] See Wolfram, "Computational Law, Symbolic Discourse and the AI Constitution", "At a foundational level, computational irreducibility implies that there will always be new concepts that could be introduced...[C]omputational irreducibility implies that none of them can ever be ultimately be complete".

[145] It is asserted that the formalized, logical form of the decision is uesed to legitimize a decision based on emotion, prejudice, or rote of training. Halper, "Logic in Judicial Reasoning", at 36-38.

[146] Ibid., at 36-38.

[147] Summers, "Logic in the Law", at 255.

actual application of the proposition and considers its implications. Logical deduction is useful in this stage since it involves determining the effect of a proposition on the existing structure of the law. This is often conceived in logical terms, i.e. whether or not there are inconsistencies.[148] There is also the argument that we should not privilege the default ways of thinking in the law. These intuitive, psychological processes are exactly the kind we need to scrutinize with logic for possible inconsistencies. While Halper concedes that legal decision making is not purely logical, and the presence of a clear body of rules will not remove judicial discretion, or eliminate the influence of nonlegal considerations.[149]

Although not couched in formalisms of modern symbolic logic, instances of both deductive and deductive thinking are inherent in legal reasoning:

> In his selection of competing propositions and in his consideration of the propriety of subsuming a particular case under a certain general rule, a judge is not, of course, guided by logic. He is guided by insight and experience. But in his application of the proposition selected, and in his testing of its implications before he adopts it, he uses a deductive form of reasoning in order to discover its potentialities. The directive force of the principle may be exercised along the line of logical progression, and a judge must always keep in mind the effect which his decision will have on the general structure of the law.[150]

Summers criticizes that this is incomplete, i.e., that logic can play a role even in the selection of premises necessary to decide particular cases. Guest also asserts that inductive logic is not applicable to law. However, Summers points out that when lawyers advise clients they often use a form of inductive logic when they make predictions and generalizations from individual cases.[151] In a way, a lawyer already treats legal questions as a computational

[148] Halper, "Logic in Judicial Reasoning", at 36-38.
[149] Ibid., at 36-38.
[150] Anthony G. Guest. "Logic in the Law". In: *Oxford Essays in Jurisprudence*. Oxford: Oxford University Press, 1961, pp. 176–197, at 188.
[151] Summers, "Logic in the Law", at 255-256.

problem, having his own estimation function based on past data such as the history of the controversy, the applicable law, and the court's past decisions.

Logic against "Judicial subterfuge" There is often a perceived tension between "rule of law" defined as "strict adherence to legal norms and their logical implications", and the aspiration to "do justice", often in the form of providing a "happy ending" for the individuals before them. This leads to judicial subterfuge, in the form of spurious interpretation of the law.[152] Goldsworthy acknowledges that there are hard cases characterized by indeterminate law. In which case judges must exercise creativity and in effect create new law. The problem lies in judges allowing considerations outside of the law in order to supplant determinate law. Often, the first step to this is engaging in the pretense that an otherwise determinate law is indeterminate, and thus the appropriate opportunity for deploying judicial creativity.Courts can delude themselves as to the content of the law, based on their long immersion in legal culture - which results in *post facto* legal rationalization of their intuitive convictions as to the proper legal solution. There is no evidence that a judge's intuitions as to practical consequences should be privileged over sound legal reasoning, and the preference for intuitive solutions, while appealing for the immediate case may erode the rule of law over the long term. The use of logic can provide a practical constraint on legal interpretation, bolstering it against judicial subterfuge.[153]

[152] Goldsworthy, "The Limits of Judicial Fidelity to Law: The Coxford Lecture", at 307; See also Pound's less generous characterization of spurious interpretation as an act that "puts a meaning into the text as a juggler puts coins, or what not, into a dummy's hair, to be pulled forth presently with an air discovery" Roscoe Pound. "Spurious Interpretation". In: *Columbia Law Review* 7.6 (June 1907), p. 379. ISSN: 00101958. DOI: 10.2307/1109940. JSTOR: 1109940. URL: https://www.jstor.org/stable/1109940?origin=crossref (visited on 04/28/2024), at 382.

[153] Logic can also help prevent a related shortcoming of the judicial process, that of "well-meaning sloppiness of thougt" - characterized by undefined or poorly defined concepts, failing to interrogate the rigor of arguments Goldsworthy, "The Limits of Judicial Fidelity to Law: The Coxford Lecture", at 318.

3.4 Modern Approaches to Law and logic

Computer scientists and philosophers have made many attempts to use logical tools to represent the intricacies of legal language and legal reasoning. This stream of work is based on the assumption that logic is a component of legal reasoning.[154]

In legal theory (as well as AI research into the law domain), the logical aspects of legal reasoning is divided into two principal approaches:[155]

First, the formal approach - where legal decisions (e.g. the judge's justification) are arrived at through a mainly deductive process. Deductive reasoning draws conclusions from a set of general principles or premises that are given or established. This is related to formal symbolic logic.

Second, the dialectic (or argument theory) approach, which views legal justification as arising from an adversarial process, where parties use discretion to evaluate between reasonable alternatives.The approach borrows much from so-called "informal logic".

The logical and dialectic approaches are seen as divergent, incompatible modes of legal reasoning, and for a long time have gone on separate tracks of development and application. The logical approach was seen as a tool for the legislative process, advancing the goal of representing laws as a set of consistent statements. Meanwhile, the dialectic approach was often applied to case-based problems that characterized litigation and judicial decision making - legal justifications derived from a process of presenting and evaluating pro and contra cases.[156] Nevertheless, Advancements in both legal theory and technology may

[154]See generally Matthias Armgardt, Patrice Canivez, and Sandrine Chassagnard-Pinet, eds. *Past and Present Interactions in Legal Reasoning and Logic*. Vol. 7. Logic, Argumentation & Reasoning. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-16020-7 978-3-319-16021-4. DOI: 10.1007/978-3-319-16021-4. URL: https://link.springer.com/10.1007/978-3-319-16021-4 (visited on 03/16/2024).

[155]Henry Prakken and Giovanni Sartor, eds. *Logical Models of Legal Argumentation*. Netherlands: Kluwer Academic Publishers, 1997. ISBN: 0-7923-4413-8. DOI: 10.1007/978-94-011-5668-4, at 1.

[156]*Ibid.*

allow for the unification of the divergent approaches (of logic and dialectics). Within the case-based reasoning that defines the dialectic approach, there is acknowledgement that consistent logical rules can be formalized. Within the logic approach, on the other hand, researchers have developed models that take into consideration the incomplete and defeasible nature of legal argumentation. [157]

The foregoing analysis will cover debates covering the first approach. Much of the work in the field has emphasized the deductive approach, due to its seeming ubiquity in legal reasoning. The deductive approach is viewed as essential to legal interpretation and application: Lawyers will analyze the text, structure (and history) of a statute to determine meaning and intent. These will then serve, along with a background of other established rules, as premises for determining applicability to specific cases.[158]

[157] *Ibid.*

[158] Jaap Hage. "A Theory of Reasoning and a Logic to Match". In: *Artificial Intelligence and Law* 4.3-4 (1996), pp. 199–273.

4 Overview of encoding and analysis approaches - Ontologies and Descriptive Logic

The proposed work is based on restating the problem of competition impact analysis in computational terms:

1. *The Relevance Problem* - Given a law, is it **relevant** to the sector for which the assessment is being made?

2. *The Threshold Testing Problem* - Given a rule within a relevant law, is the rule **compliant** with the norms laid out by the threshold test?

From a computational point of view, the problem of competition impact assessment is a problem of logical comparison and evaluation. It involves comparing the provisions of the law that cover a sector with a set of standards, and then evaluating the extent to which the law complies with the standards. The standards can refer to the OECD threshold tests (and are further elaborated in the economics literature, usually based on models of a competitive market). In order to proceed with automating this evaluation, a computational law system will require: 1. A system for encoding the content of legal text, as well as 2. Algorithms that can process these encodings.

Based on the previous chapter, we are proceeding from the notion that law and questions of law are largely computable problems.[159] Facilitating computation of law requires encoding systems for both problems: First to represent, then to analyze these representations(determine relevance, and evaluate for compliance). These appear to be distinct problems and require different encoding systems. The encoding methodology for this study uses two divergent approaches, each applicable to a different aspect of the law. The first approach aims to capture the semantic content of the law through ontologies, which are used to model the entities and relationships in a domain. The second approach is concerned with

[159] A computable question is one that can be computed by a sufficiently powerful "Turing machine".

Table 1: Encoding and Analysis Approaches

Problem	**Encoding**	**Analysis**
Relevance Testing: Does the law map with the industry being assessed? (Actors, transactions)	Ontologies (Ontology Web Language)	Reasoning engines to determine relationships: - No mapping? - Identity? - Classification? - Mereological? - Inference?
Threshold Testing: Given a specific rule within a relevant law - How does this rule relate to the norm of the threshold test?	Inference rules (Prakken, Sartor) - LegalRuleML	Argumentation Frameworks Propositional networks

representing the normative constraints contained in the law as a set of defeasible inferential statements in deontic logic.[160] This chapter provides an overview of both approaches, with a focus on how they can be applied to the domain of competition law.

Since every modern computer language is Turing complete (i.e. it can fully implement a Turing machine), these programming languages are capable of computing legal questions. The only constraints will be time, memory, and computing power. Andersson (2014) asserts that most software tools (general purposes, modern languages) are overkill for implementing the requirements of a computational law system. It would be more efficient (cost-benefit wise) to develop and use domain-specific languages for computational law.[161] However, it is very difficult to come up with domain specific languages specific to law - this may be a function of few lawyers knowing how to program, and few programmers understanding law.

[160]It may be possible to combine both the semantic and normative aspects. Both ontologies and inference statements are based on logic and can be arranged into network structures. In the future, machine learning may be used to automatically translate rules into logical formalisms. Meanwhile, the exercise will be undertaken by humans.

[161]See Andersson, "Computational Law: Law That Works Like Software", at 21.

4.1 Ontological Representation of Legal Semantics

4.1.1 Definition and benefits

Law provides a description of the world - which can be made legible as a configuration of entities and relationships. The entities are the actors, transactions, and objects that are the subjects of the law. The relationships are the connections between these entities, and the attributes that describe them. This aspect of the law can be encoded as an ontology. An **ontology** is a formal, explicit description of concepts that are part of a domain.[162]. It consists of: 1. **classes** that represent concepts; 2. **properties** that describe features of these concepts, including their relationship with each other; and 3. **restrictions** to the way these classes and attributes are defined.[163] An ontology of classes, along with specific instances of these classes, constitute a **knowledge base**, although as a practical matter there can be little to distinguish this from an ontology.[164] Ontologies can be used to make web pages (or other electronic resources) more "understandable" to electronic agents. Many disciplines are developing standardized ontologies used by experts to encode, annotate, and share knowledge in their respective fields,providing a common vocabulary researchers and a source of machine-readable definitions.[165]. Noy (2001) suggests that for extensive domains of knowledge, ontologies can provide the following benefits:

[162]Natalya F Noy and Deborah L McGuinness. "Ontology Development 101: A Guide to Creating Your First Ontology". In: *Stanford Medical Informatics Technical Report* (SMI-2001-0880 Mar. 2001). URL: `http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html`, at 3. The term ontology originally referred to a branch of philosophy concerned with the study of being. It was borrowed by computer science to refer to the formal definition of objects in a domain, and the relationships between these objects. See Lamy Jean-Baptiste. *Ontologies with Python: Programming OWL 2.0 Ontologies with Python and Owlready2.* Berkeley, CA: Apress, 2021. ISBN: 978-1-4842-6551-2 978-1-4842-6552-9. DOI: `10.1007/978-1-4842-6552-9`. URL: `http://link.springer.com/10.1007/978-1-4842-6552-9` (visited on 04/03/2024), at §3, p. 61.

[163]Michael De Bellis. *A Practical Guide to Building OWL Ontologies.* Oct. 8, 2021. URL: `https://www.michaeldebellis.com/post/new-protege-pizza-tutorial` (visited on 01/31/2024), at 6.

[164]Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology".

[165]For medicine, for example, there is SNOMED (Price and Spackman, 2000) and the Unified Medical Language System (Humphrey and Lindberg, 1993); For describing products and services for the purpose of trade regulation, see the United Nations Standard Products and Services Code(UNSPC), at https://www.unspsc.org/

1. *Sharing and collaboration* Experts and practitioners can represent their shared understanding.

2. *Enabling reuse* Users can build on existing ontologies - extending or refining them as needed.

3. *Making assumptions explicit* Assumptions can become explicit in the design of an ontology, making it easier to question and resolve them as necessary.

4. *Separating domain knowledge from operational knowledge* We can analyze a class of concepts in the abstract, independent of particular instances.

5. *Analyzing domain knowledge* Once a representation is available, it can be subjected to formal analysis.

An ontology can be formally expressed in a computer language. This work will use the Web Ontology Language (OWL) to express ontologies. The choice is largely based on the OWL's broad adoption, and the availability of supporting software and documentation. OWL is a language that is based on Description Logic, a subset of first-order logic that is used to represent knowledge in a structured and formal way.[166][167] For protoyping and visualization purposes, the author will use the Protégé ontology editor, which is a widely used tool for creating and editing ontologies in OWL.[168]

[166]See Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 3.

[167]OWL is a standard that is maintained by the World Wide Web Consortium (W3C), the organization that sets standards for the web. It is used to represent knowledge in a way that is machine-readable and can be processed by computers. OWL is based on the Resource Description Framework (RDF), a standard for representing information on the web. RDF is used to represent information in the form of triples, which consist of a subject, a predicate, and an object. OWL extends RDF by providing a way to represent classes, properties, and relationships between classes and properties. It also provides a way to represent restrictions on classes and properties, such as cardinality constraints and value constraints. OWL is used in a wide range of applications, including the Semantic Web, data integration, and knowledge representation. It is a powerful language that can be used to represent complex knowledge in a structured and formal way.

[168]See Mark A. Musen. "The protégé project: a look back and a look forward". In: *AI Matters* 1.4 (2015), pp. 4–12. DOI: `10.1145/2757001.2757003`. URL: `https://doi.org/10.1145/2757001.2757003`.

Managing and retrieving data from ontologies is more efficient and cost-effective compared to Large Language Models (LLMs). To make corrections, one simply needs to identify and modify the specific entity and attribute. This approach is more appropriate for making precise factual determinations where accuracy is prioritized over expressiveness. The use of ontologies is also more transparent and interpretable compared to LLMs. The structure of the ontology can be visualized and understood by humans, and the reasoning process can be traced and explained. This is important for legal applications, where the reasoning process must be transparent and understandable to the parties involved.

4.1.2 Ontology components:classes and properties

Classes are the primary focus and building blocks of an ontology. These describe concepts in a domain.[169] Since we are concerned with modelling entities that interact with each other and the law, our ontology can have a `Person` class that represents the legal definition of a person - an individual or entity that has the capacity to enter into legal relations. A class can have **subclasses** that represent more specific concepts.[170] For example, the `Person` class can have subclasses such as `Natural_Person` to represent a human individual and `Juridical_Entity`, such as a corporation. Individuals (or *instances* of these classes) are the actual objects in the domain of interest.[171] For example, the `Natural_Person` class can have instances such as `Alice` and `Bob`.

Properties and inheritance describe the attributes of and relationships among classes and instances. The class definition of the `Person` class can have `has_name` property that describes the name of a person, which can be provided for an instance of that class. Properties can also be used to describe the relationships between classes. For example, the `Person` class can have a `has_child` property that describes the relationship between a parent and

[169] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 3.

[170] Ibid., at 3.

[171] See De Bellis, *A Practical Guide to Building OWL Ontologies*, at 7. At the same time, classes can be thought of as sets that contain individuals.

a child. The `has_child` property can be used to connect a `Natural_Person` instance to another `Natural_Person` instance that is their child. Properties can also have restrictions that define the cardinality of the property, the value of the property, or the relationship between the property and other properties. For example, the `has_child` property can have a restriction that specifies that a child can have at most two parents. Subclasses inherit the properties of their parent classes, and can have additional properties that are specific to them.[172]

4.2 Ontology construction

There is no one "right" methodology for constructing an ontology. Noy(2001) proposes an iterative approach: With a rough, initial pass, filling details along the way. It is a question of what is most appropriate for the applications in mind and the developments anticipated for the ontology. There should at least be a sense of isomorphism, or closeness, between an ontology and the common understanding of the domain.[173] This can be achieved by reflecting on the statements that describe the domain. The nouns correspond to the classes/instances, while the verbs and adjectives correspond to the attributes.

The ontology to be used for this work shall be designed based on the following process outline in Noy(2001), with some details provided by DeBellis (2021):

1. Determine the domain and scope of the ontology
2. Consider reusing existing ontologies
3. Enumerate important terms
4. Define the classes and class hierarchy
5. Define the internal structure of classes

[172] See De Bellis, *A Practical Guide to Building OWL Ontologies*, at 7.

[173] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 4.

6. Define the restrictions of attributes

This short tour of the design process will also serve as an opportunity to describe how ontologies can model the semantic content of the relevant competition law, as well as some of the early design decisions taken.

STEP 1: Determine domain and scope of the ontology The first step requires us to specify the domain of interest, as well as the contemplated uses of the ontology.This study is concerned with several domains, each of which can be modelled through separate ontologies: 1. The entities and transactions in the digital payments market in the Philippines, as described by the relevant laws; and 2. The idealized configuration of entities and transactions in a competitive market, as described by the OECD threshold tests. This work will focus on the OECD Guidelines since it has become the *ad hoc* basis of the Philippine's competition impact assessment regime. It is also the most comprehensive and most updated resource of this type available to the public. Other similar guidelines, such as those issued by the International Competition Network, the Asia Pacific Economic Cooperation, and the UK's Competition and Markets Authority will be used to supplement our understanding of the norms applicable to competition impact assessment. The primary purpose of the ontology is to enable the evaluation of laws governing a particular sector for competition effects. For this chapter, we will use the first OECD threshold test standard as an example. The OECD tests for the following competition concerns:

For this demonstration of the design process we are only concerned with A1, which flags a law as having competition issues if it "Grants exclusive rights for a supplier to provide goods or services". Note that although the header for Section A by itself is not a threshold test, and its general normative requirement (i.e., that it not "limits the number or range of suppliers"), it is considered part of the domain since it may still provide information as to required classes and properties.

Figure 1: The OECD Threshold Test Checklist

COMPETITION ASSESSMENT CHECKLIST

Competition assessment should be conducted if a legal provision has any of the following effects:

A Limits the number or range of suppliers

This is likely to be the case if the provision:

- ☐ **A1** Grants exclusive rights for a supplier to provide goods or services
- ☐ **A2** Establishes a license, permit or authorisation process as a requirement of operation
- ☐ **A3** Limits the ability of some suppliers to provide goods or services
- ☐ **A4** Significantly raises cost of entry or exit by a supplier
- ☐ **A5** Creates a geographical barrier for companies to supply goods, services or labour, or invest capital

B Limits the ability of suppliers to compete

This is likely to be the case if the provision:

- ☐ **B1** Limits sellers' ability to set prices for goods or services
- ☐ **B2** Limits freedom of suppliers to advertise or market their goods or services
- ☐ **B3** Sets standards for product quality that provide an advantage to some suppliers over others, or are above the level that some well-informed customers would choose
- ☐ **B4** Significantly raises costs of production for some suppliers relative to others (especially by treating incumbents differently from new entrants)

C Reduces the incentive of suppliers to compete

This may be the case if the provision:

- ☐ **C1** Creates a self-regulatory or co-regulatory regime
- ☐ **C2** Requires or encourages information on supplier outputs, prices, sales or costs to be published
- ☐ **C3** Exempts the activity of a particular industry, or group of suppliers, from the operation of general competition law

D Limits the choices and information available to customers

This may be the case if the provision:

- ☐ **D1** Limits the ability of consumers to decide from whom they purchase
- ☐ **D2** Reduces mobility of customers between suppliers of goods or services by increasing the explicit or implicit costs of changing suppliers
- ☐ **D3** Fundamentally changes information required by buyers to shop effectively

STEP 2: Consider reusing existing ontologies The knowledge base can be also be based on existing ontologies that have already been developed for some knowledge domains or specific activities. For example, the financial sector is already covered by the Financial Industry Business Ontology (FIBO), a knowledge graph that models the entities and transactions in the financial sector.[174]It is a standard that is already being used by financial institutions, regulators, and other stakeholders. For concepts related to law, we may derive from the design of LegalRuleML[175].Finally, the Wikidata project is a knowledge base that models data that can be found in the open web.[176] Whenever appropriate, we can use these ontologies directly, or design our ontology to be compatible with them.

STEP 3: Enumerate important terms We next proceed to listing the important terms that the ontology needs to describe and explain, as well as their relevant properties - **property attributes** can qualify classes (i.e. what they are "like"), while **functional attributes** can describe what the classes can do, or what can be done to them.[177]. The rule of thumb is to consider the nouns of statements as the classes of the ontology, while adjectives and verbs can be considered as the properties. For the competition impact assessment ontology, we can start with the following terms (with implied terms in parentheses):

The ontology designer should also take note of any term that may be in the statement being modelled, but are nevertheless implied by the other terms. For example, since the standards mention a `Supplier`, it can be inferred even at this point that we need to model the ultimate recipient of the goods and services supplied - a `Consumer`. Both `Supplier` and `Consumer` are subclasses of `Person`, which we will also need to define and elaborate later on. Finally, since the standards in the threshold test are meant to apply to laws - hence the

[174]See EDM Council. *The Financial Industry Business Ontology.* FIBO. URL: `https://spec.edmcouncil.org/fibo/` (visited on 01/18/2024).

[175]Oasis Open. *LegalRuleML Core Specification Version 1.0.* Aug. 30, 2021. URL: `http://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/legalruleml-core-spec-v1.0.html` (visited on 10/06/2023), See also See https://www.gecad.isep.ipp.pt/ieso/contract/v1.0.0/#description for a basic contract ontology.

[176]See Wikimedia Foundation. *Wikidata.* URL: `https://www.wikidata.org/wiki/Wikidata:Main_Page` (visited on 01/18/2024).

[177]Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 6.

Table 2: Example terms for the ontology

Nouns (Classes)	**Verbs or Qualifiers (Attributes)**
Right	
(Person)	limit
Supplier	number
(Consumer)	range
Good	grant
Service	provides
(State)	exclusive
(Law)	

need for a `Law` class. The `State` class is also implied, as the standards assume that there is a state that is enacting and enforcing the law.[178]

STEP 4: Define the classes and class hierarchy Several approaches are open to determining the classes and their place in the hierarchy (i.e. the subclass-superclass relationship). There is the **top-down approach** which is to start with the most general concepts, and then proceed to the more specific cases. Alternatively, one can also take a **bottom-up approach**, which means to start with defining the most specific classes, then determine if these can be grouped into general concepts (i.e. generate common superclasses). The more realistic approach is a combination of both, i.e. define the salient concepts and then generalize or specialize as needed. No method is best - it would depend on the circumstances surrounding the modeling, i.e. if a general view is available, if data is granular enough to describe specific cases.[179] To determine which terms can be classes or subclasses, a good rule of thumb is that objects that are capable of independent existence (rather than descriptions of other objects) can be the principle classes in a class hierarchy.Once classes are identified and defined, arrange them hierarchically into a taxonomy. This can be done

[178] Although the text of the OECD tests refers to some concepts in the plural (e.g. "Goods"), the naming convention will use the singular form. Classes represent sets and can contain multiple instances. Thus, it is not necessary to define singular forms of classes as subclasses.

[179] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 6-7.

by asking for each class, whether it can be an instance of the same class.[180]

For the competition impact assessment ontology, we can start with the following configuration of classes and subclasses:

- `Person` - An individual or entity with legal capacity. This can have the following subclasses:
  - `Natural_Person` - A human individual, to which the class of `Consumer` belongs.
  - `Juridical_Entity` - A legal entity, which can include a `Corporation` - which in turn is the superclass of any `Supplier` object (an entity that provides a `Good` or a `Service`).[181]
- `Right` - A legal entitlement (or permission, in deontic terms) that can be granted or limited by the `State` through a `Law`. The right concerns the ability to offer and enter into a contract concerning a `Provision`, which can have the following subject matters:
  - `Good` - Physical objects that can be supplied by a `Supplier`. Can refer to any tangible object that can be bought or sold.
  - `Service` - Intangible objects that can be supplied by a `Supplier`. Can refer to any contractual performance.[182]

Class hierarchies show how concepts are related. They use terms like "is-a" or "kind-of" to show these connections. When one class is a subclass of another, it means the subclass

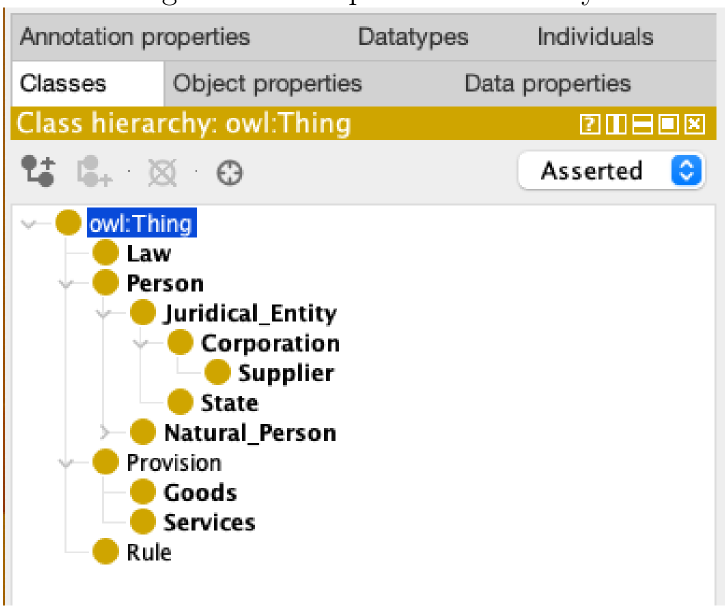
[180] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 7-8.

[181] Note the simplifying assumptions that we are holding for now in order to facilitate the design of the ontology. In the real world, a corporation can be a consumer, and a natural person can be a supplier. The artificial distinction however, may be "true enough" for the purposes of our ontology.

[182] The classes of Good and Service can be bound by reference to another ontology, such as the United Nations Standard Products and Services Code (UNSPC).

represents a more specific type of the general concept represented by the main class.[183] A subclass relationship is transitive, i.e. ”If B is a subclass of A and is a subclass of B, the C is a subclass of A”.[184]. It may also be useful to determine at this point which classes are *disjoint*,i.e., that no individual can be an instance of more than one of those classes.[185] In our example, the `Natural_Person` and `Juridical_Entity` classes are disjoint. Objects that are instantiated as either of those classes can only belong to one class or another.The class hierarchy, as constructed in Protégé can be visualized as shown in the following figure:

Figure 2: Example class hierarchy



Note that in OWL, all classes are subclasses of a root class called `owl:Thing`, the class that represents the set containing all individuals. All empty ontologies still contain one class called `owl:Thing`.[186]

[183] Noy and McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, at 12.
[184] Ibid., at 13.
[185] Ibid., at 16.
[186] De Bellis, *A Practical Guide to Building OWL Ontologies*.

STEP 5: Define the internal structure of classes The internal structure of the classes can be defined through its properties or attributes.[187]. For every class in our ontology, we are concerned with both intrinsic and extrinsic properties:[188]

- **Intrinsic properties** - There are the essential, or inherent to the class itself. These properties are essential to the identity and nature of the class, independent of external factors or contexts. They are characteristics that an instance of the class possesses purely by being an instance of that class. For the class `Person`, intrinsic properties might include a `has_name` - since each legal person, whether an individual human being or a corporation, has a name.

- **Extrinsic properties** - These are context-dependent, relational attributes of a class. Extrinsic properties are those that depend on external factors or the context in which an instance of the class exists. These properties are not essential to the identity of the class and can change depending on the environment, relationships, or interactions with other entities. For the class `Person`, extrinsic properties might include the person's current location, occupation, marital status, or the clothes they are wearing.[189]

Subclasses inherit the properties of their parent classes, and can have additional properties that are specific to them.[190] For example, the `Natural_Person` class can inherit the `has_name` property from the `Person` class, and can have additional properties such as `has_age` and `has_address`. The `Juridical_Entity` class can inherit the `has_name` property from the `Person` class, and can have additional properties such as `has_registration_number` and `has_legal_address`.

[187] Also called slots in earlier documentation

[188] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 8.

[189] A form of extrinsic properties that relate the class to other classes are mereological properties, i.e. a class can also have can have physical and abstract parts (e.g. the parts of an engine or the courses of a meal)

[190] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 9.

STEP 6: Define the attribute restrictions The properties of a class can have restrictions that define the cardinality of the property, the value of the property, or the relationship between the property and other properties.[191] We can define the cardinality of an attribute - how many values a property can have. The `has_name` for a person can have a single cardinality - that is, a person is allowed to only have one legal name. Other properties can have multiple cardinality. For example, the `has_child` property of a `Natural_Person` class can have a restriction that specifies that a child can have at most two parents, or several friends. We can also define restrictions for acceptable values that can be entered for each property: The `has_name` property can have a restriction that specifies that the value of the property must be a string(i.e. a series of text characters), or that the `has_age` property can have a restriction that specifies that the value of the property must be a positive integer. By specifying the domain and **range** of an attribute, we can place restrictions on the relationships of classes. The **domain** of a property refers to the set of all objects that can have that property asserted about it.[192] The **range** of a property, on the other hand, the set of all objects that can be the value of the property.[193] For example, the fact that a `Law` can contain many `Rules` can be modelled by the `has_rule` attribute. The `has_child` property can also have a restriction that specifies that a child must be a `Natural_Person` instance.

When defining a domain or range of an attribute, Noy(2001) recommends finding the most general classes or class that can serve the purpose. Nevertheless the domain or the range should not be too general, i.e. the classes in the domain of an attribute should be described by the attribute, and the instances of all the classes in the range of an attribute should be potential values for the attribute.[194]

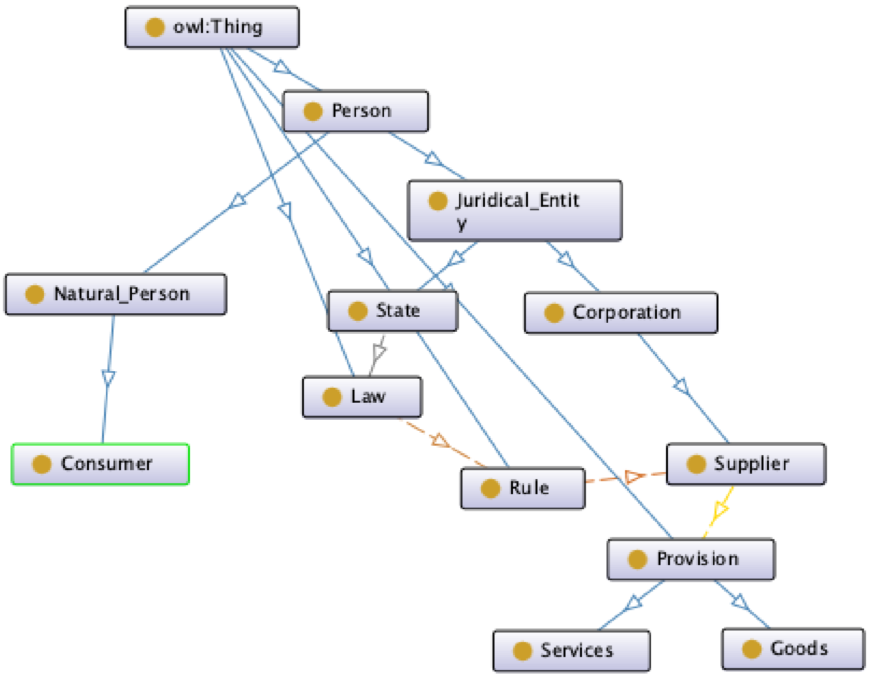
[191] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 9.

[192] De Bellis, *A Practical Guide to Building OWL Ontologies*, at 26.

[193] Ibid.

[194] Noy and McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", at 10.

Figure 3: Initial class diagram



4.3 Representation of normative constraints

4.3.1 Inference rules

The threshold test of competition impact assessment can be stated more formally as follows: Given a set of rules (i.e., the rules that cover an industry) - does it comply with or diverge from the idealized norm of the threshold test? In previous assessment exercises, to make things manageable logistically, the author has proposed making individual provisions the unit of analysis. However, even a provision can still express several rules, each of which can be independently evaluated. Therefore rules will serve as our unit of analysis.

Ontologies only give us a part of the picture. Besides the entities, their attributes and interactions - all these are subject to constraints and transformations based on law. These only provide static data about the semantics of entities and their interactions - but these do not reflect the legal constraints that act upon those objects, and how the semantics could be qualified, transformed, or annulled by such constraints. Another way of putting it is that knowledge graphs reflect only the whats and the who's, not the oughts and ought nots that contained in legal knowledge.

Legal provisions may be restated into atomic inference rules, which have the structure - If P then Q. It is also possible to state a rule categorically as simply Q, but this should be rare in operation.[195]

Take for example the simple rule "If a lane is designated as a bus lane, then only buses can drive through it". This can be broken down to several inference rules:

- If [Lane has Bus Only Markings] then [Lane is Designated]
- If [Lane is Designated] then ¬[Driver Enters]
- If [Driver Enters] then [Violation]

Once we have formal representations, the next step would be to apply analytical methods grounded in logic. We can trace the chain of inferences (via *modus ponens*), discover other rules, even look for potential inconsistencies.

4.3.2 Deontic Logic

Legal statements are for the most part, not composed of factual statements. They do not describe the state of the world as it is, but how it ought to be. They can't be assessed for truth values. Furthermore, legal conclusions are arrived at under an informational environment marked by incompleteness, uncertainty, and inconsistency.[196] Logicians have since

[195] See Giovanni Sartor. "A Formal Model of Legal Argumentation". In: *Ratio Juris* 7.2 (July 1994), pp. 177–211. ISSN: 0952-1917, 1467-9337. DOI: `10.1111/j.1467-9337.1994.tb00175.x`. URL: `https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9337.1994.tb00175.x` (visited on 06/20/2023).

[196] See Kathleen Freeman and Arthur M. Farley. "A Model of Argumentation and Its Application to Legal Reasoning". In: *Artificial Intelligence and Law* 4.3-4 (1996), pp. 163–197, at 165.

developed a form of logic, called Deontic Logic, which is not concerned with True or False, but oughtness: Whether certain acts or states of the world are: Obligatory, Prohibited, or merely Permitted.[197]

Deontic Logic was influenced by modal logic (which concerns modalities, or expressions that qualify the truth of propositions, i.e., necessity and probability) Although notions of Deontic Logic have been explored in fourteenth century Europe as well as Islamic thought (in the 10th century), its modern version grounded in symbolic logic is based on the work of Von Wright (1951).[198]

Instead of the binary values of True or False, Deontic Logic accommodates six normative states:

1. It is obligatory that (OB)
2. It is permissible that (PE)
3. It is impermissible that (IM)
4. It is omissible that (OM)
5. It is optional that (OP)
6. It is is non-optional that (NO)

Recasting the earlier example under the Deontic mode:

- If [Lane has Bus Only Markings] then [Lane is Designated] - No changes because this is actually a factual statement.

[197] See G.H. von Wright. "Deontic Logic". In: *Mind* 60.237 (Jan. 1951), pp. 1–15, for the original use of the term and the first modern systemization of the field; See also Paul McNamara and Frederik Van De Putte. "Deontic Logic". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2022. Metaphysics Research Lab, Stanford University, 2022. URL: `https://plato.stanford.edu/archives/spr2022/entries/logic-deontic/` (visited on 06/08/2022), for an updated overview.

[198] McNamara and Van De Putte, "Deontic Logic", at § 1.

- If [Lane is Designated] then ¬[Driver Enter] becomes: O([Lane is Designated] $\rightarrow\neg$ [Driver Enters]) - The inference is neither true nor false, but has the deontic modality of Obligation (O).

- If [Driver Enters] and [Lane is Designated] then [Violates] becomes O([Driver Enters] $\wedge$ [Lane is Designated] $\rightarrow$ [Violation]) - That is, if the car enters the lane when the lane is designated as a bus lane, then we must find a violation

4.3.3 Defeasibility and argumentation

Another attribute of legal propositions is that they are **defeasible**. This means that they are tentative - accepted until some other proposition - a new fact that activates an exception, better evidence, or even a higher law - defeats our original proposition.[199]

Legal conclusions are arrived at based on knowledge that is incomplete, uncertain, and inconsistent. Despite this, an adequate theory of legal reasoning should provide a sound basis of what to believe. Argumentation theory is suited to the problem because it takes into consideration contrasting claims under an environment of uncertainty and inconsistency.[200]

The model proposed by Freeman views argument in the following ways:

1. As a structure for supporting explanation - It consists of discrete units of arguments that connect claims with data

2. As a dialectical process - It consists of a series of moves by opposing parties that either support or attack a given claim[201]

[199] See generally Giovanni Sartor. "Defeasibility in Legal Reasoning". In: *Rechtstheorie* 24.3 (1993), pp. 281–316.
[200] Freeman and Farley, "A Model of Argumentation and Its Application to Legal Reasoning", at 163-164.
[201] Ibid., at 167.

Freeman's model integrates the notion of burden of proof - the level of support necessary for any one party to "win" the argument. This serves as filter, turntaking mechanism, and termination criteria. The process enables the generation of decisions that could fall anywhere within the continuum of skeptical and credulous.[202]

4.4 Automated analysis and evaluation

The canonical approach requires evaluation of the relevant laws for features that match a predetermined list of factors (usually based on the economics literature). It relies on both a reading of the text, and the lawyer's training on how the text is most likely interpreted and enforced. What usually happens, based on the recommendations of these guides, is an appeal to the lawyer's intuition as to the intent and consequences of the legal text. Some of these guides suggest, to balance out the inherent subjectivities in that determination: Consulting other stakeholders (regulators and industry stakeholders). While this cross analysis might go a long way towards making the conclusions less stilted, there is still no proof of work that can be shared and independently studied, changed, and evaluated. We should be able to rely on a transparent chain of reasoning proceeding from plausible assumptions into consistent propositions, that can be shared, analyzed, built on top of each other.

Once we have the rules encoded, the goal is to perform automated evaluations. We can look for internal inconsistencies, or gaps in the coverage of industry entities and transactions. Then we can compare one set of rules - such as the legislation under competition impact assessment, with the standards set by the economic literature, or the competition authority, or international organizations. Once law is reduced to a formalized structure, then it becomes amenable to direct comparison - for finding difference and inconsistency. Unlike intuitive assessments, though, the reasoning process is exposed from the start - the assumptions are provided (or at least very easy to look up), and each step towards the

[202] *Ibid.*

conclusion is available for proof.

Ontologies and inference rules can be combined into network structures, and it is possible to compare network structures - i.e. to what extent these structures are similar or different. But beyond some of the more obvious methods, this work will explore two pathways that will enable computers to compare and evaluate the encoded rules: 1. Argumentation frameworks and 2. Propositional networks.

The first takes into account the dialectic nature of arriving at a legal determination. Conclusions about law are often only arrived at after an argument - one side presents a plausible reading of the law, another counters with a supposedly better reading of the law, or evidence of factual circumstances that would make the law inapplicable, or a higher law.[203] The initial proponent could counter, and on and on until the arguments are exhausted and a decision has to be made by some process and standard. In the computational law field, there are so called argumentation frameworks. These are tools for modeling both rules and facts into arguments. Normative claims can be encoded just like rules, while the facts embodied in knowledge graphs can serve as evidence, or a warrant that either supports or undercuts a claims. In order to be processed an argumentation framework, we need to add information as to how all the claims and warrants relate to each other - either supporting or attacking. A reviewer can set the burden of proof, the weight of different kinds of evidence, and the standard required for an argument to prevail over the other.

Another method to be explored is through propositional networks. Propositional networks are an extension of game theory.[204] It is used in artificial intelligence, used for playing games and programming logic. Under this approach, entities and transactions can be modeled

[203] See generally Frans H. Van Eemeren et al. *Handbook of Argumentation Theory.* Dordrecht: Springer Netherlands, 2014. ISBN: 978-90-481-9472-8 978-90-481-9473-5. DOI: 10.1007/978-90-481-9473-5. URL: https://link.springer.com/10.1007/978-90-481-9473-5 (visited on 06/20/2023).

[204] See Michael Genesereth and Michael Thielscher. *General Game Playing.* Red. by Ronald J. Brachman, William W. Cohen, and Peter Stone. Synthesis Lectures on Artificial Intelligence and Machine Learning 24. Morgan & Claypool Publishers, 2014.

as they are in a knowledge graph - related to each other through states, attributes, and transactions. Unlike the static representation of knowledge graphs, however, propositional nets allow us to model transitions in both entities and relationships that can be caused either by constraints or actions - which can be provided by law. Propositional networks can be used to model the behavior of entities and transactions over time, and how they interact with each other.

The approach should combine the norms in our deontic propositions with the structured information in a knowledge graph, such that the norms can interact with the semantic information. Because the law can assume that the [Driver] is an adult and is licensed, and if neither of those are true, then a different set of norms apply. At the same time, a state of [Violation] would mean that the status of [Driver] could be modified i.e., suspended or annulled.

5 Overview of Encoding and Analysis Approaches - Normative Component

In order to carry out automated reasoning of law, we have to encode legal norms into computational forms. In the previous section, ontologies and description logic, helped us define the descriptive component of the legal knowledge (in this case the OECD Competitive Impact Assessment tests) that we seek to encode. The analysis that can be performed on an ontology-based data structure can reveal implicit relationships between entities (such as inheritance,equivalence), as well as inconsistencies. However, ontologies only give us a part of the picture. Besides the entities, their attributes and interactions - all these are subject to constraints and transformations based on law. These only provide static data about the semantics of entities and their interactions - but these do not reflect the legal constraints that act upon those objects, and how the semantics could be qualified, transformed, or annulled by such constraints. Another way of putting it is that ontologies reflect only the whats and the whos, not the oughts and ought nots that are contained in legal knowledge.

In this chapter, we shall cover the requirements of a logical system for representing important normative features of a body of rules: First, that it should capture the conditional nature of legal inferences; Second, it should involve modalities other than True or False - that is, it should work on normative states (for example, whether propositions are permitted, forbidden, or obligatory); Finally, it should also allow for the possibility of inferences being defeated by additional information. The chapter shall describe these features in turn, and propose Reified IO Logic as an encoding system that integrates all these requirements.

The choice of encoding system is based on Robaldo (2020)'s description of a computational knowledge base for legal rules,[205] which accomodates several levels of encoding:

[205] Livio Robaldo, Cesare Bartolini, and Gabriele Lenzini. "The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML". in: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020).* Marseille, May 11–16, 2020, pp. 5688–5697, at 5688-5689.

1. **Legal text** - Written in a human readable language but tagged and structured through an XML-based markup (such as LegalDocML, an OASIS standard for legal markup). At this level, the system designer encodes the law as is, but provides markup for some sections in order to signal the structure of the document, as well as highlight concepts that are relevant to the ontology and logic layers. This allows systems to associate these elements with subsequent logical encodings the represent their meaning. This will allow components of the text to be linked to the subsequent encodings and support automated processing.

2. **Legal ontology** - This consists of the formalized naming and definitions of concepts that are contained in the human-readable rules, as described in the previous chapter. Concepts and relationships are encoded in OWL, and will serve as the predicates to be used in Thehe normative logic layer. The ontology can also be described in terms of Description Logic, and can support some analysis. Howeverm this ontology layer alone is not fit for legal reasoning, as it does not account for deontic aspects of the rules, or accounts for their defeasibility.

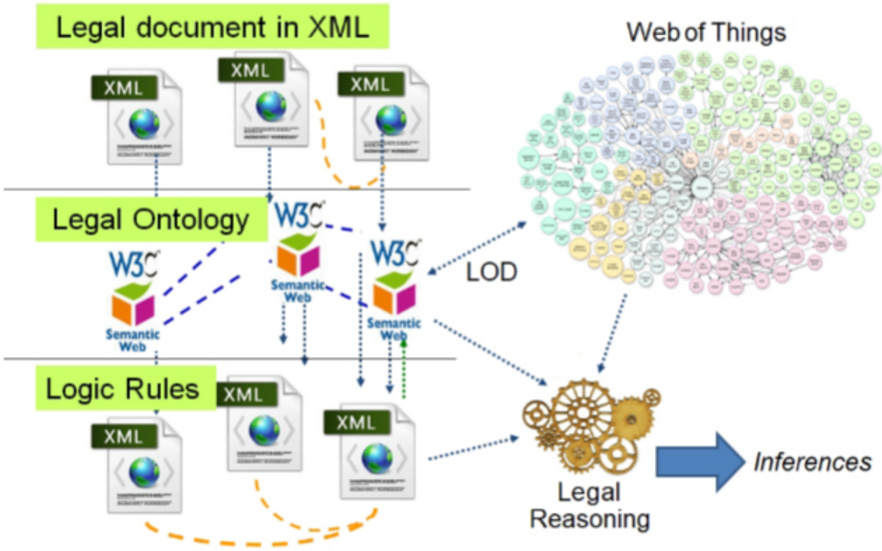
3. **Normative logic** - This layer represents the normative content of the rules, represented as logical formulae. This logic layer is formalized in a defeasible form of deontic logic and then encoded in LegalRuleML.

5.1 Availability of Multiple Logical Systems

In stating that we will translate legal rules into a logical encoding, we mean "logic" as a formal method that can support deductive reasoning. That is, proving a conclusion by means of at least two other propositions.[206] The term includes not just Aristotelian syllogism, but can accomodate other forms of deductive inferences, such as the logic of alternatives, compound propositions, and of relationships, and the study of propositions

[206] Ruggero Aldisert, Stephen Clowney, and Jeremy Peterson. "Logic for Law Students: How to Think Like a Lawyer". In: *University of Pittsburgh Law Review* 69 (2007), pp. 1–22, at 2.

Figure 4: Layered Architecture of Encoding(Robaldo, 2020)



themselves.[207] Burgin (2022) provides a high-level overview of the evolution of logical systems: From loose collection of rules related to belief systems to more modern, formalized logics.[208] The diversity of logics can provide tools for the representation of various aspects of knowledge. Each logic can capture and emphasize a certain level of description, or comprehend specific problems.

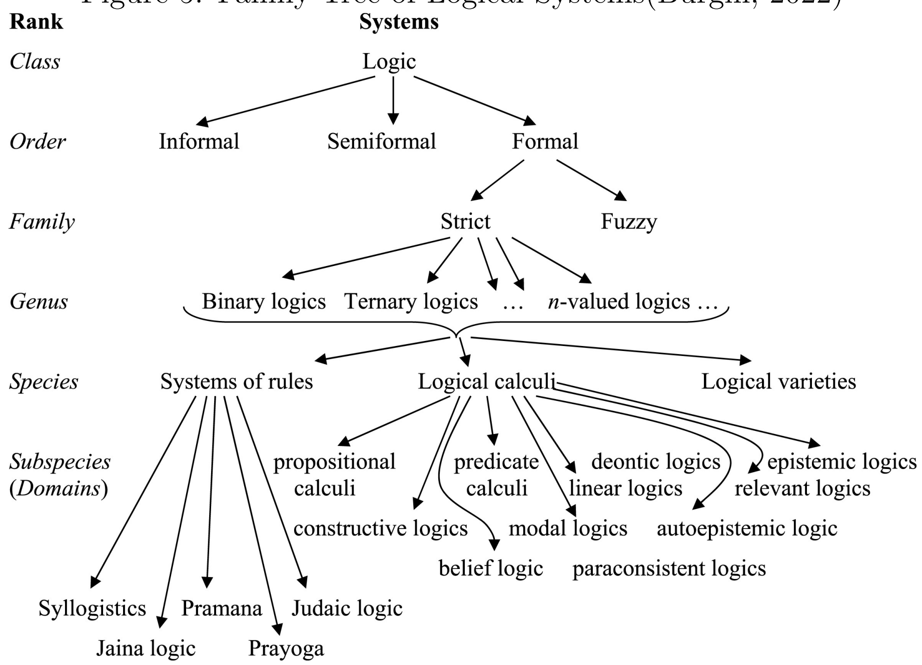
The development of novel logical systems now allow us to have a more focused view of a problem.[209]. For example: Triadic logics which allows for intermediate truth values, rejecting the law of the excluded middle; A fuzzy logic, which has instead of True or False, an infinite continuum of possible values, along with a more informal process of inference.

[207] Guest, "Logic in the Law".

[208] See generally Mark Burgin. "Evolution of logic as an information processing mechanism in advanced biological systems". In: *Bio Systems* 221 (2022), p. 104758. ISSN: 0303-2647. DOI: 10.1016/j.biosystems.2022.104758. URL: https://www.sciencedirect.com/science/article/pii/S0303264722001393.

[209] Susan Haack. "On Logic in the Law: "Something, but Not All"". In: *Ratio Juris* 20.1 (2007), pp. 1–31. ISSN: 0952-1917, 1467-9337. DOI: 10.1111/j.1467-9337.2007.00330.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9337.2007.00330.x (visited on 03/26/2024), at.

Figure 5: Family Tree of Logical Systems(Burgin, 2022)



Then there are deontic logics, with new operations such as "obligatory", "permitted", and "forbidden".[210]

5.2 Legal Norms as Conditional Inferences

In more conventional forms of logic, we can readily represent factual statements. Take for example the proposition, earlier made explicit as a fact in the ontology, that corporations are also persons:

$$\text{All Corporations are Persons(All } S \text{ is } P) \tag{1}$$

[210]Deontic logics in particular have attracted legal theorists as a way to make formal, rigorous representations of the structure of legal orders. A variation of the idea of deontic modes is a component of the encoding sytem proposed later in this chapter Haack, "On Logic in the Law", at 11-12.

The predicate Persons is descriptive of the subject Corporations. If the proposition is admitted, it follows that no Corporations are not Persons. If the proposition is denied, then it follows that some Corporations are not Persons.

However, let us take a normative or legal statement, “Any person who shall abuse its market dominance shall be guilty of a criminal offense”. The predicate “guilty of a criminal offense” is not necessarily descriptive of the subject ”person who abuses market dominance”. The relationship between the components of the proposition hinges on the injunctive ”shall”, which is not descriptive of what is, but instead denotes what ought to be under certain contingencies.[211]. Normative propositions are more comparable to the structure of causal inferences: If p then q. Instead of making factual predictions, however, they are statements of what ought to be.[212] Sartor states that legal provisions may be restated into such atomic inference rules, with the basic if p then q structure.[213] It may also be possible to state a rule categorically as simply q, but this should be rare in operation.[214].

Take for example rule A1 in the OECD Guidelines, that a provision should be flagged if it “Grants exclusive rights for a supplier to provide goods and services”. This can be broken down to several inference rules:

[211] Guest, “Logic in the Law”, at 183-184.

[212] Guest also clarifies that these are not necessarily imperative statements or commands. ibid., at 184.

[213] The consequent of each rule is a litera;. and the antecedent is a conjunction of literals. A literal is an atomic formula or the negation thereof. A positive literal has the form ‘p(x)‘ where ‘p‘ is a predicate symbol and ‘x‘ is a list of terms. On the other hand, a negative literal has the form ‘not p(x)‘ where ‘not‘ is a logical negation. The complement $\bar{q}$ of a literal q denotes the literal opposed to q : If q is a positive literal p, then $\bar{q}$ represents the negative literal not p ; if q is the negative literal not p, then $\bar{q}$ represents the positive literal p. Sartor, “A Formal Model of Legal Argumentation”, at 179.

[214] Sartor’s formalization also admits of these so-called “degenerate inference rules”. These enable unconditional derivation of any instance of their conclusion A. These categorical inference rules can be used to express some forms of ungrounded assertion, such as:1. Statements of undisputed empirical evidence (the facts that justify a law or court decision); 2. Basic (and very general) normative postulates; 3. Tentatively advanced propositions for which no ground is currently available ibid., at 179.

$$\text{If [Law Requires Single Supplier] then [Rights are Exclusive]} \tag{2}$$

$$\text{If [Rights are Exclusive] then [Flagged]} \tag{3}$$

Where [Rights are Exclusive] stands for p and [Flagged] stands for q.

Initially, we can think of rules of substantive law as statements of the specific factual conditions upon which specific consequences depend. The applicability of the condition can be contingent on several conditions, as well as the absence of exceptions.[215] Thus:

$$\text{If events } x_1 \ldots x_n \text{ is the case, and unless there are } y_1 \ldots y_n, \text{then } z \text{ is the case.} \tag{4}$$

A legal system can be represented as a body of such propositions that can be evaluated not based on truth or falsity, but by some other normative standard (such as social benefit, or compliance with other higher rules). Once we have formal representations, the next step would be to apply analytical methods grounded in logic. We can trace the chain of inferences (e.g. via *modus ponens*), discover other rules, even look for potential inconsistencies.[216]

[215] Jerome Michael and Mortimer J. Adler. "The Trial of an Issue of Fact: I". in: *Columbia Law Review* 34.7 (Nov. 1934), pp. 1224–1306. ISSN: 00101958. DOI: `10.2307/1116103`. JSTOR: `1116103`. URL: `https://www.jstor.org/stable/1116103?origin=crossref` (visited on 12/11/2024), at 1241.

[216] Inference rules are mono-directional, to be used/understood only forward (*modo ponente*) and not backward (*modo tollente*). The consequent q can be derived whenever the antecedent p is satisfied. However, the negation of p cannot be derived when q is assumed to be false. The "if" connective in inference rules is not the same "if" in logical conditionals. Sartor, "A Formal Model of Legal Argumentation", at 179.

5.3 Deontic Logic

Legal statements are for the most part, not composed of factual statements. They do not describe the state of the world as it is, but how it ought to be. They cannot be assessed for truth values. Logicians have since developed a form of logic, called Deontic Logic, which is not concerned with True or False, but oughtness. Although notions of Deontic Logic have been explored in fourteenth century Europe as well as Islamic thought (in the 10th century), its modern version grounded in symbolic logic is based on the work of von Wright (1951).[217]. Under von Wright(1951)'s classic formulation, it is concerned with the following modes of obligation:[218]

- **Obligatory** - That which we ought to do
- **Permitted** - That which we are allowed to do
- **Forbidden** - That which we must not do

For von Wright the starting point of his deontic system is the concept of "Permitted" as the basic operator - e.g. a proposition can ϕ is Permitted, $P\phi$. Other operators can then be defined in terms of P:[219]

$$F\phi =_{df} \neg P\phi \text{ - something Forbidden is not Permitted} \quad (5)$$

$$O\phi =_{df} \neg P\neg\phi \text{ - something Obligatory is something not Permitted not to do} \quad (6)$$

Deontic logic was later axiomatized and developed to what is now known as Standard Deontic Logic (SDL). Under SDL, the primary operator is Obligation, denoted as by the

[217] McNamara and Van De Putte, "Deontic Logic", at § 1.

[218] Von Wright, "Deontic Logic", at 1.

[219] Donald Nute, ed. *Defeasible Deontic Logic*. Dordrecht: Springer Netherlands, 1997. ISBN: 978-90-481-4874-5 978-94-015-8851-5. DOI: `10.1007/978-94-015-8851-5`. URL: `http://link.springer.com/10.1007/978-94-015-8851-5` (visited on 11/23/2024), at 2.

symbol $\bigcirc$ (or "ought"). The Permitted operator can be defined as:[220]

$$PE\phi =_{df} \neg \bigcirc \neg\phi \tag{7}$$

That is, ϕ is Permitted if and only if it is not am Obligation that not ϕ. We can thus construct all the other operators in terms of $\bigcirc$ (See Table 3 at 85 below).[221]

Table 3: SDL Definitions of Deontic Operators

Definition	**Implication**	**Example**
$\bigcirc$(OB)	A proposition is obligatory if it must occur	It is OBligatory to pay taxes
PE $\phi =_{df} \neg \bigcirc \neg\phi$	A proposition is permissible iff (if and only if) its negation is not obligatory	It is PErmitted to drive a car
IM $\phi =_{df} \bigcirc\neg\phi$	A proposition is impermissible iff (if and only if) its negation is obligatory	It is IMpermissible to smoke in a restaurant
OM $\phi =_{df} \neg \bigcirc \phi$	A proposition is omissible iff it is not obligatory (can be omitted or not done without violating a norm)	It is OMissible to attend that party (you can attend or not attend)
OP $\phi =_{df} (\neg\bigcirc\phi \wedge \neg\bigcirc\neg\phi)$	A proposition is optional iff neither it nor its negation is obligatory	It is OPtional to work from home
NO $\phi =_{df} (\bigcirc\phi \vee \bigcirc\neg\phi)$	A proposition is non-optional iff it is either obligatory or impermissible	It is NOn-optional to wear a seatbelt while driving

[220] Nute, *Defeasible Deontic Logic*, at 2.
[221] McNamara and Van De Putte, "Deontic Logic", at § 1.2.

Recasting the earlier example under the Deontic mode:

$$\text{If [Law Requires Single Supplier] then [Rights are Exclusive]} \tag{8}$$

$$\text{If [Rights are Exclusive] then } \bigcirc \text{ [Flagged]} \tag{9}$$

The inference that the law should be flagged is neither True nor False, but has the deontic modality of Obligation. Once represented formally, it may be possible to evaluate a specific statement based on the axioms and theorems of the chosen system of deontic logic. For example, in von Wright's classical system, there exists the Principle of Deontic Distribution which provides that: "If an act is the disjunction ("or") of two other acts, then the proposition that the disjunction is permitted is equivalent to the disjunction of the propositions that the first act is permitted and the proposition that the second act is permitted":[222]

$$P(\phi \vee \psi) \leftrightarrow P\phi \vee P\psi \tag{10}$$

Applying this to the proposition that a Supplier is permitted to supply goods or services, then the permission is distributed individually to the supply of goods as well as the supply of services. While this distributive property is a feature of this particular system of deontic logic, it is not universally accepted, and we can discard this axiom if it conflicts with our normative intuition.

[222]Nute, *Defeasible Deontic Logic*, at 2.

5.4 Defeasibile Deontic Logic

Another attribute of legal propositions is that they are **defeasible**. This means that they are tentative - accepted until some other proposition - a new fact that activates an exception, better evidence, or even a higher law - defeats our original proposition.[223]

Substantive legal provisions often have a **positive condition**, the event or circumstance that must obtain for the purported legal consequence to be arrived at. At the same time, these conditions are most likely subject to exceptions - elements that according to some antecedent norms has to be absent in order for the legal consequence fo apply: The sum of positive conditions embody the determination of the legislator of what circumstances should normally give rise to the legal consequences. On the other hand, the exceptions represent special circumstances that can override the positive conditions, making the legal consequences not applicable. Legal conclusions are often subordinated structures: The presence of other legal provisions (that are of equal or higher priority in a hierarchy of norms), which may provide (or negate) conditions and exceptions.[224] The goal of legal reasoning in actual cases is to show that certain acts, claims, decisions comply or does not comply with the law. This requires demonstrating that the presence (or absence) of conditions and exceptions.[225] Thus, legal conclusions are arrived at based on knowledge that is incomplete, uncertain, and inconsistent[226] - on plausibility rather than truth. Despite this, an adequate formalization of defeasible reasoning should provide a sound basis of what to believe.

[223] See generally Sartor, "Defeasibility in Legal Reasoning".

[224] This arises from what Stuart Hampshire calls the "inexhaustability of description" Any situation can embody an inexhaustible set of features, but we can only confront and understand part of it at any given time. See Juan Carlos Bayon. "Why Is Legal Reasoning Defeasible?" In: *Diritto & Questioni Pubbliche* 2 (2002), pp. 1–18, at 3; Citing Stuart Hampshire, ed. *Public and Private Morality.* Cambridge: Cambridge University Press, 1991. 143 pp. ISBN: 978-0-521-22084-2 978-0-521-29352-5, at 30.

[225] Bayon, "Why Is Legal Reasoning Defeasible?", at 3.

[226] See Freeman and Farley, "A Model of Argumentation and Its Application to Legal Reasoning", at 165.

Various such formalizations have been developed to embody defeasibility of reasoning. For our purposes, a system of defeasible reasoning should allow for the representation of various propositions and their attributes: 1. Atomic ”facts” that are taken as a given; 2. Rules (whether or not they are subject to exceptions); 3. Defeating propositions and/or superiority relationships; 4. In the case of legal statements especially, their deontic modal values (Obligatory, Permitted, or Forbidden). A system of defeasible reasoning should also enable operations on these propositions, such as resolving conflicts and making plausible inferences. For example - through prioritization of certain rules and/or the evaluation of supporting or undercutting evidence.[227]

In Defeasible Deontic Logic (DDL), legal norms are the positive conditions that prescribe behavior through Permission, Obligation, and Prohibitions. These norms may be subject to exceptions (which are also expressed as norms).[228] DDL allows for the representation of facts, defined as whatever can be considered as conclusive unambiguous statements. Facts can include: either a state of affairs or actions already performed (both considered to always hold true). Based on our ontological definitions, we can state that ”Acme Inc. is a corporation” through:

$$\text{Corporation(Acme Inc.)} \tag{11}$$

[227] Hanif Bhuiyan et al. “Traffic Rules Encoding Using Defeasible Deontic Logic”. In: *Frontiers in Artificial Intelligence and Applications*. Ed. by Serena Villata, Jakub Harašta, and Petr Křemen. IOS Press, Dec. 1, 2020. ISBN: 978-1-64368-150-4 978-1-64368-151-1. DOI: 10.3233/FAIA200844. URL: http://ebooks.iospress.nl/doi/10.3233/FAIA200844 (visited on 11/23/2024), at 9; See also Sanjay Modgil and Henry Prakken. “The ASPIC+framework for Structured Argumentation: A Tutorial”. In: *Argument & Computation* 5.1 (Jan. 2, 2014), pp. 31–62. ISSN: 1946-2166, 1946-2174. DOI: 10.1080/19462166.2013.869766. URL: http://content.iospress.com/doi/10.1080/19462166.2013.869766 (visited on 11/27/2023).

[228] Hanif Bhuiyan et al. “A Methodology for Encoding Regulatory Rules”. In: *Proceedings of the 4th International Workshop on MIning and REasoning with Legal Texts Co-Located with the 32nd International Conference on Legal Knowledge and Information Systems (JURIX 2019)*. International Workshop on MIning and REasoning with Legal Texts 2019. Vol. 2632. Madrid, Spain: Rheinisch-Westfaelische Technische Hochschule Aachen, Dec. 11, 2019, at 2.

A rule in DDL is a relationship between a set of antecedents or premises (clauses), represented as $X_1, ..., X_n$ and the consequent conclusion or conclusion (effect) of the rule, is represented as Y. The strength of the relationship between the premises and conclusion allows us to differentiate between strict rules, defeasible rules, and defeaters:[229]

Strict rules (encoded as $X_1, ..., X_n \rightarrow Y$) are inferences in the classical propositional sense. If the premise is indisputable, then so is the conclusion. E.g. "A Corporation is a Supplier":

$$\text{Corporation(Acme Inc.)} \rightarrow \text{Supplier(Acme Inc.)} \tag{12}$$

Defeasible rules (encoded as $X_1, ..., X_n \Rightarrow Y$) are inferences that are generally true, but can be defeated by other information. An example in the guidelines is that a Supplier cannot be an exclusive provider unless the economic sector allows for a natural monopoly:

$$\text{Corporation(Acme Inc.)} \Rightarrow \text{ExclusiveSupplier(Acme Inc.)} \tag{13}$$

From this, we can conclude that a corporation can be an exclusive supplier, unless there is evidence to the contrary.

Defeaters (encoded as $X_1, ..., X_n \rightsquigarrow Y$) are rules that can prevent a conclusion. Building on the previous example, we can maintain that:

[229] Bhuiyan et al., "A Methodology for Encoding Regulatory Rules", at 8-9.

$$\neg(\text{Sector\_AllowsNaturalMonopoly(Acme Inc.)}) \leadsto \neg\text{ExclusiveSupplier(Acme Inc.)} \tag{14}$$

Defeasible logic can resolve conflicting information by allowing the prioritization of rules through the superiority ($\succ$) relation. E.g. $r1 \succ r0$ means that rule $r1$ takes precedence over rule $r0$. This can be used to resolve conflicts between rules, or to determine the applicability of a rule in a given context.

Finally, DDL takes into account deontic properties such as Obligation (O), Permission (P) and Prohibition (F) and their relationships in SDL. For example, as to the attribute `ExclusiveSupplier`, the Prohibition against acting an exclusive supplier is equivalent to the Obligation not to act as an exclusive supplier.

$$[\text{F}]\text{ExclusiveSupplier} \equiv [\text{O}]\neg\text{ExclusiveSupplier} \tag{15}$$

Thus. the rule that disallows exclusive suppliers (subject to the exceptions for natural monopolies) can expressed as:

$$\emptyset\text{(Empty Set)} \Rightarrow [\text{F}]\ \text{ExclusiveSupplier} \tag{16}$$

$$(\text{Sector\_AllowsNaturalMonopoly(Acme Inc.)}) \Rightarrow [\text{P}]\ \text{Exclusive Supplier} \tag{17}$$

5.5 Next:Encoding Into LegalRuleML

All the logical systems discussed above build on each other and allow us to have a formalized representation of legal norms, enabling various operations and evaluations on these norms. Efficient, automated reasoning with these norms can be achieved by applying the logical model into a machine-readable format. The interest from the Artificial Intelligence and Law communities computational representation of norms has led to the development of digital formats for encoding the logical aspect of legal texts, such as the Rule Markup Language (RuleML),[230] Semantic Web Rule Language (SWRL), Rule Interchange Format (RIF), and the Legal Knowledge Interchange Format (LKIF).

LegalRuleML, an XML-based standard developed and maintained under the auspices of the Organization for the Advancement of Structured Information Standards (OASIS),[231] represents a convergence of many of these previous efforts, with broad support from both industry and academic communities.[232] LegalRuleML allows for the modelling of both constitutive rules and prescriptive rules as if-then statements (antecedent and consequent) with deontic effects, as well as properties and operations related to defeasibility. Detailed discussion of LegalRuleML's features (e.g. reification, temporal management, ontology references) will be provided as they are implemented in encoding the competition impact assessment guidelines. Besides the rich set of modern features, and a design approach that can accommodate multiple theories of logic and norms, there are practical advantages to employing LegalRuleML:1. It is an open standard, with the full specification and documentation available online; 2. It has broader support compared to other formats, leading to a larger codebase of examples and related applications; 3. As an XML-based format, it can be

[230]W3C. *RuleML - W3C RIF-WG Wiki.* 2005. URL: https://www.w3.org/2005/rules/wg/wiki/RuleML (visited on 01/12/2025).

[231]Oasis Open, *LegalRuleML Core Specification Version 1.0.*

[232]Tara Athan et al. "OASIS LegalRuleML". in: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law.* ICAIL '13: International Conference on Artificial Intelligence and Law. Rome Italy: ACM, June 10, 2013, pp. 3–12. ISBN: 978-1-4503-2080-1. DOI: 10.1145/2514601.2514603. URL: https://dl.acm.org/doi/10.1145/2514601.2514603 (visited on 12/16/2024).

connected to any ontology, readily providing the rules with semantics.

6 Description of the Encoded Knowledgebase

This chapter covers a description of a knowledgebase,[233] derived from a computational encoding of item A1-5 of the OECD Competition Impact Analysis Guidelies ("OECD Guidelines"). Focusing on A1 as an example, we will discuss the ontological account of the encoding, followed by a normative account.

6.1 Logical Account of the Encoding

As a software artifact, the knowledgebase is composed of an ontology layer encoded in the Object Web Language (OWL) and the rules layer encoded in LegalRuleML. The source code is freely accessible online and will be included as an annex to this work. Focused discussion of the knowledgebase will be carried out through two forms of related logical notations: A form of Description Logic for the ontology, and Reified I/O Logic for the rules layer. Both logical notations are based on First Order Logic and have been applied to the problem of encoding legal norms. This will allow for a consistent syntax and symbol set for the discussion. The approach also preserves the distinction between two kinds of logical formulae:[234]

1. TBox - the terminological declarative statements, encoded as any formula in DL; and
2. ABox - the assertive contextual statements encoded as flat conjuctions of atomic predications in Reified I/O Logic.

6.1.1 Description Logic

Description Logic (DL) is a term for a family of knowledge representation languages. A decideable fragment of DL called $\mathcal{SROIQ}$ is the basis for the Object Web Language

[233] OWL source code of the knowledgebase is accessible through https://github.com/emersonbanez/dissertation

[234] Robaldo, Bartolini, and Lenzini, "The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML", at 5690.

and will be used to notate the ontology formulae in this section.[235] DL is a subset of First Order Logic (FOL), predicate logic, or first order predicate calculus. It extends propositional logic, representing valid arguments and logical truths that depend on the internal structure of propositions.[236]DL is used in mathematics, computer science to express statements about objects and their relationships. DL enables the representation of atomic concepts (also called classes and subclasses) representing basic concepts in a domain[237] (such as $Person$, $JuridicalEntity$), and define other concepts in terms of these atomic concepts, e.g. A $Natural\_Person$ should be any person that is not a juridical entity ($Person \sqcap \neg JuridicalEntity$). In DL, nouns in the legal text can be mapped into atomic concepts, while verbs and descriptors into atomic roles, with subsequent terms defined by combining these components into through logical operators. For example, a $Buyer$ is defined as an $Actor$ that is also a $Role\_in\_a\_Legal\_Matter$: $Buyer \sqsubseteq Actor \sqcap Role\_in\_a\_Legal\_Matter$. This can also be further defined in terms of what it is not (A $Buyer$ cannot be a $Seller$): $Buyer \sqsubseteq Actor \sqcap \neg Seller$. DL also supports atomic roles (or properties in OWL)that can define the relationships between concepts. If the knowledge domain for example requires that every juridical entity should have a director, then an ontology with a `Juridical_Entity` class is bound to a `Director` class through the $hasDirector$ role: $Juridical\_Entity \equiv \exists hasDirector.Director$. That is, one definition of a juridical entity is that it belongs to the set of objects that is related to the set of objects called $Director$ through the $hasDirector$ role. We can further qualify the relationship by imposing domain restriction to the role. For example, we can specify that only adults, or $Adult \equiv Natural\_Person \sqcap hasAge.(\geq 18)$ can serve as directors. Role restrictions, such as those pertaining to cardinality, can help us define the requirement that a juridical entity should at least have 5 directors, for example:

[235]Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. "A Description Logic Primer". In: *IEEE Intelligent Systems* 29.1 (2014), pp. 12–19. DOI: `10.48550/arXiv.1201.4089`. arXiv: `1201.4089 [cs]`. URL: `http://arxiv.org/abs/1201.4089` (visited on 06/20/2025), at 1.

[236]This analysis of internal structure is carried out through quantifiers (i.e. "for all x", "for some x"), singular terms or names (e.g. "a", "b", etc) and predicates, i.e. "for all x, if Fx then Gx". Haack, "On Logic in the Law", at 11.

[237]Daniele Nardi and Ronald J. Brachman. "An Introduction to Description Logics". In: *The Description Logic Handbook*. Cambridge University Press, 2007, pp. 1–43, at 8.

$Juridical\_Entity \equiv \geq 5 hasDirector.Director$

6.1.2 Reified I/O Logic

Reified I/O logic is a formalism for representing norms in textual form. It combines I/O logic from Markinson and Van der Torre (2000) with a reification-based logic (Hobbs and Gordon 2017)[238]. In encoding the components of a normative statement, the formalism relies on reification. Reification formalizes events and states to correspond to First Order Logic (FOL)terms (constants, variables, functional terms).[239] States and events denoted by these terms are treated like things in the world. Thus, "eventuality" denotes both the reification of a state as well as that of an event or action.[240] In their system every FOL predication, e.g. (*dominant* Meta), which asserts that Meta is dominant firm may be associated with another FOL predication, ($dominant'$ e_d Meta) where e_d is a reified eventuality. The term e_d is the reification of Meta's status as a dominant firm.[241] From the reification of the state of being dominant, other predications may be applied to e_d, and then recursively reified to express more elaborate semantics. For example, the statement that Meta intends to be dominant can be encoded as:

[238] Robaldo, Bartolini, and Lenzini, "The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML", at 5689; Gordon et al has built an extensive index of predications (possible modifiers of subjects), encoding them based on the ISO/IEC 2407 the Common Logic Standard. Andrew S. Gordon and Jerry R. Hobbs. *A Formal Theory of Commonsense Psychology: How People Think People Think*. 1st ed. Cambridge University Press, Aug. 31, 2017. ISBN: 978-1-107-15100-0 978-1-316-58470-5. DOI: 10.1017/9781316584705. URL: https://www.cambridge.org/core/product/identifier/9781316584705/type/book (visited on 12/05/2024), pp. 54–55.

[239] Eventualities can be treated as if they were objects of human thoughts. As reified eventualities can act as parameters for predicates like *believe*,*think*, *want* Livio Robaldo and Xin Sun. "Reified Input/Output Logic: Combining Input/Output Logic and Reification to Represent Norms Coming from Existing Legislation". In: *Journal of Logic and Computation* 27.8 (Dec. 1, 2017), pp. 2471–2503. ISSN: 0955-792X, 1465-363X. DOI: 10.1093/logcom/exx009. URL: http://academic.oup.com/logcom/article/27/8/2471/3098296 (visited on 12/16/2024), at 10-11.

[240] Ibid., at 3.

[241] Hobbs logical framework contemplate two sets of logical predicates working in parallel: primed and unprimed. 1) Unprimed refers to the standard FOL predicates, e.g. (*give* a b c) asserts that a a gives b to c; Primed represents reification of the corresponding unprimed relation, e.g. ($give'$ e a b c) says that e is a giving event by a of b to c. ibid., at 10.

$$(intend'\ e_i\ Meta\ e_d) \wedge (dominant'\ e_d\ Meta)$$

The fact of Meta's intention represented by e_i. This predication is applied to the previously defined e_d, the reified eventuality of being dominant. We can then instantiate the eventuality of Meta intending to obtain a dominant position through a special predication called $RexistAtTime$, providing for its real existence within a time context t:

$$(RexistAtTime\ e_i\ t)$$

This allows us to make a distinction between actually being dominant, and only intending to be so - since only e_i has a real existence for a given time. The final representation of the statement that Meta intends to be dominant, without making any inference about it being actually dominant, can be written as follows:

$$(RexistAtTime\ e_i\ t) \wedge (intend'\ e_i\ Meta\ e_d) \wedge (dominant'\ e_d\ Meta)$$

The extent of the use of reification is such that even boolean operators can be represented as eventualities. For example, $(not'\quad e_1\quad e_2)$ is used to assert that e_1 is the eventuality of e_2's non-existence.[242]

These statements based on reified logic can then be used to model normative statements with the standard structure of I/O logic. In I/O logic, each normative statement is represented by input/ouput pairs (x, y) with x at the left hand side of the comma representing the precedent, and the right hand side y representing the consequent of the normative conditional.[243] The output is then considered as an input to further processing as part of any

[242] not' is not a boolean operator, but is an FOL predicate that describe (through relations) 2 eventualities, one of which is the negation of the other. Robaldo, Bartolini, and Lenzini, "The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML", at 5689.

[243] Livio Robaldo et al. "Formalizing GDPR Provisions in Reified I/O Logic: The DAPRECO Knowledge Base". In: *Journal of Logic, Language and Information* 29.4 (Dec. 2020), pp. 401–449.

of sets C, O and P (constitutive rules, obligations, and permissions).[244]

Combining these insights, we can now model a hypothetical rule which states that: "Firms that are dominant and abuse their dominant position must be fined". In normative conditional form:

> IF a firm x is dominant and is abusing its dominant position THEN x paying a fine is OBLIGATORY.

The corresponding Reified I/O formula, with generalizations based on First Order Logic, is as follows:

$$
\begin{aligned}
\forall_x \forall_t (\\
\exists_{e_a} \exists_{e_{ab}} \exists_{e_d} [(RexistAtTime\ e_a\ t) \wedge \\
(and'\ e_a\ e_{ab}\ e_d) \wedge (abuse'\ e_{ab}\ e_d) \wedge \\
(dominant'\ e_d\ x)], \\
\exists_{e_p} [(RexistAtTime\ e_p\ t) \wedge \\
(pay'\ e_p\ x\ Fine)]) \in O
\end{aligned}
$$

6.2 Design Overview

6.2.1 Semantics from OECD Guidelines

The ontology, its terms, structure, semantics, as well as norms to be encoded, is based on the OECD guidelines as a starting point, but it can be extended to account for other normative frameworks. The programmatic account of the ontology outlined here are usually

ISSN: 0925-8531, 1572-9583. DOI: `10.1007/s10849-019-09309-z`. URL: `http://link.springer.com/10.1007/s10849-019-09309-z` (visited on 11/25/2024), at 6.

[244] It is based on established distinctions between regulative norms and constitutive norms. (Searle, 1995). 1. Regulative norms - are obligations and prohibitions (the deontic statements) while 2. Constitutive norms - definitions of the concepts used in regulative norms Robaldo, Bartolini, and Lenzini, "The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML", at 5689; John R. Searle. *The Construction of Social Reality.* New York: The Free Press, 1995.

implemented as fragments of the FOLIO and FIBO ontologies. Whenever possible, the content and structure of the ontologies will be combined and harmonized. For the ontology we are constructing, whenever possible we will avoid the complications of the Open World Assumption.[245] By default (and unless specified) classes with a common superclass will be set as disjoint (i.e. they do not have individuals in common) This convention will also facilitate data validation requirements for later analysis.[246]

As much as practicable the content of the ontology will be restricted to what is necessary to express the semantics in the OECD guidelines. This means that certain classes and relationships will be defined partially, and may not be fully valid based on other domains, such as legal theory or economics.The encoding takes into consideration:

1. The text of the rule in the OECD guidance;
2. Annotations in the guidelines;
3. When applicable, sources from competition law and economics

6.2.2 Use of LKIF Ontology for Foundational Concepts

The Legal Knowledge Interchange (LKIF) Core ontology is basic library of primitive legal concepts, meant to be re-used by end-users (citizens, lawyers, legal scholars) who can provide the concepts it contains further definition. Thus, for semantics related to legal concepts, the ontology also adopts the encoding in LKIF. This includes an appropriate conception of an actor in the legal domain, as well as what constitutes a legally valid norm. The top layer of LKIF-core is relied upon to provide the framework for instantiating legal concepts (which can be abstract) and legal subjects (both physical or abstract) in a shared

[245] The Open World Assumption - is the assumption that the knowledge in the ontology is incomplete, and there might be other individuals or relationships not yet explicitly stated. Thus if something is not explicitly stated in an ontology - it is not assumed to be false; it is simply unknown.

[246] De Bellis, *A Practical Guide to Building OWL Ontologies*, at 76.

conception of universe that is still subject to constraints of location, time, parthood, and change.[247]

Agents in LKIF LKIF adopts the premise that legal reasoning is based on the behavior of rational agents that can be effectively influenced by the law. These agents are represented by the class of `Agents` that, while in a particular `Role`, perform `Actions`. Agents are capable of maintaining internal mental states (through `Intention` and `Belief`) as well as communication with other agents through `Expressions`.Agents can be subdivided into individual agents called `Persons` (e.g. Pope Leo XIV, or Jean Tirole). and `Organizations` which are aggregates of persons (or other organizations) - to which we can subscribe a unified internal mental state, and to which we can attribute actions.[248] In LKIF, any agent is a vector of an intended outcome of an action. That is, all actions are intentional. `Actions` are `Processes` in that they result through some causal necessity, in some `Change`.[249] An `Agent`'s subclass can include `Person` - which can be a `Natural_Person` or a `Legal_Person`. The LKIF concept of `Actor` contemplates that it can take on a particular role in a legal matter or transaction. For our purposes, such role can be as `Seller` or `Buyer` in a market. The LKIF connects `Agents` and `Actions` through the concept of a `Role`. A role is not just a classification scheme (i.e some actions relate to only some rules) Roles also convey normative information, in the sense that they can specify what the standard is for certain behaviors. Roles can thus also be the basis of how other agents are expected to behave. The LKIF connects `Agents` and `Actions` through the concept of a `Role`. A role is not just a classification scheme (i.e some actions relate to only some rules) Roles also convey normative information, in the sense that they can specify what the standard is for certain behaviors. Roles can thus also be the basis of how other agents are expected to behave.[250]

[247] Rinke Hoekstra et al. "The LKIF Core Ontology of Basic Legal Concepts". In: *Proceedings of LOAIT 07*. II Workshop on Legal Ontologies and Artificial Intelligence Techniques. 2007, p. 43–63, at 48.

[248] Ibid., at 53.

[249] Ibid., at 52.

[250] Ibid., at 54.

Legal Norms in LKIF LKIF also contains a selection of prebuilt classes that correspond to concepts in the legal domain - legal agents, actions, rights, and powers.[251]A norm applies to a situation, connecting it to it certain qualifications. I.e. A situation can be obligatory, allowed, or prohibited.[252]

6.3 Encoding of CIA Guidelines - Checklist A

The rules in Checklist A are concerned with legal constraints that limit the number and range of suppliers, which creates the risk that market power will be created and reduce competitive rivalry.

6.3.1 A1 - Grant of exclusive rights

The text of A1 of the OECD guidelines states that a law should be flagged if it "Grants exclusive rights for a supplier to provide goods or services". Exclusive rights are suboptimal from a competition perspective because they are the "ultimate barrier to entry" and can lead to monopoly pricing. The exclusive supplier, once entrenched, may also be difficult to regulate for the purpose (in terms of limiting market power and protecting public welfare).[253] The rule embedded in this text can be expressed as a normative conditional statement:

> IF a law grants exclusive rights for the provision of goods or services to a supplier, THEN the law **must** be flagged.

The statement yields several nouns that can be treated as classes, as well as verbs and adjectives that can be mapped into properties in the DL/OWL ontology, as laid out in Table 1.

[251]Hoekstra et al., "The LKIF Core Ontology of Basic Legal Concepts", at 56.
[252]Ibid., at 56-57.
[253]OECD, *Competition Assessment Toolkit - Volume 1 (Principles)*, at 10.

Nouns	Adjective	Verb	Object
Law		grant - grants, is_granted_by	Rights
Rights	exclusive		
Seller (Supplier)		sell (provide) - sells, is_sold_by participates_in	Goods OR Service Market
Goods			
Services			
Market			
Buyer		participates_in	Market

Table 4: Classes and properties in Rule A1

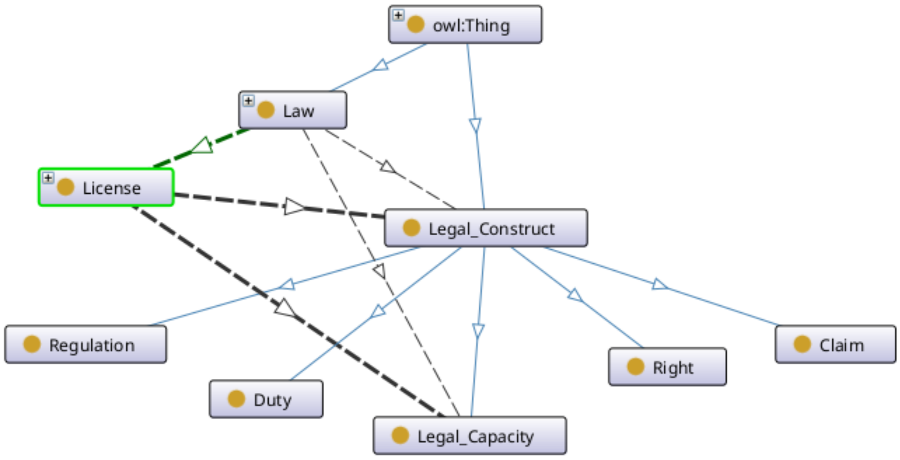
Model of Law and Rights The ontology adopts the LKIF distinction between: 1. The law, as embodied as an artifact, usually a `Legal_Document` (that is a subclass of `Document`, or $Legal\_Document \sqsubseteq Document$). It serves as the medium through which a `Legal_Expression` is made known.[254] Subclasses of this concept are organized based on the authority holding the norm, e.g. a `Legal_Document` can be `Statute`, `Treaty`, or `Contract` 2. The `Legal_Expression` itself, an abstract object which can contain propositions that define, or make deontic qualifications concerning a situation.[255] A law, as `Legal_Document`

[254] "A legal document is a document bearing norms or normative statements. By virtue of this definition the norm-as-propositional-attitude is reified as norm-as proposition. In other words, the norm being expressed through the legal source is an expression of the propositional attitude." ESTRELLA Consortium. *Legal Knowledge Interchange Format (LKIF) - Core Ontology.* Version 1.1. 2008. URL: `http://www.estrellaproject.org/lkif-core`, at #Legal_Document; This corresponds with the concept of "Legal Authority" in FOLIO, defined as "Documents or publications that guide legal rights and obligations (e.g., caselaw, statutes, regulations, rules) or that can be cited as providing guidance on the law (e.g., secondary legal authorities)." The Institute for the Advancement of Legal and Ethical AI. *Federated Online Legal Information Ontology (FOLIO).* FOLIO - Federated Online Legal Information Ontology (FOLIO). 2024. URL: `https://folio.openlegalstandard.org/taxonomy/browse` (visited on 04/28/2025), See "Legal Authority".

[255] "A norm is a kind of Qualification. A qualification which normatively qualifies some thing (i.e. some normatively qualified): i.e. a qualification which allows or disallows some thing." ESTRELLA Consortium, *Legal Knowledge Interchange Format (LKIF) - Core Ontology*, at #Norm; This corresponds with the FIBO concept of "law", defined as a "rule recognized by some community as regulating the behavior of its members and that it may enforce through the imposition of penalties" EDM Council. *The Financial Industry Business Ontology.* FIBO. Jan. 2020. URL: `https://spec.edmcouncil.org/fibo/` (visited on 01/18/2024), See https://spec.edmcouncil.org/fibo/ontology/FND/Law/LegalCore/Law.

can contain one or several `Legal_Expressions`. Each legal expression is the basis for each encoded formula in the knowledgebase. A form of a norm is a `Right`, which qualifies a situation as something that a rights holder (usually some `Agent`) is entitled to.[256]

Figure 6: Class Hierarchy of Legal Concepts in A1



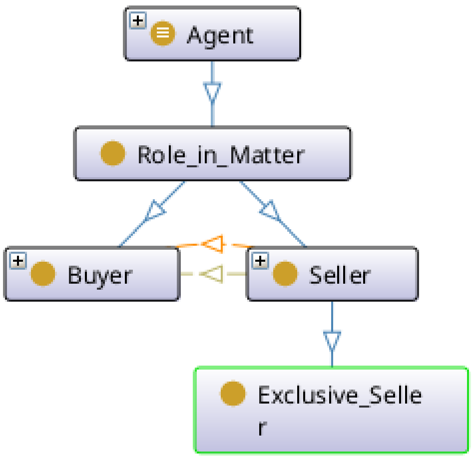
Persons Subject to A1 The rule contemplates several `Persons` who are the subjects of the rule - such as the `Seller`, `Buyer`, These participants are based on the LKIF `Agent` class, which is designed to be the basis of any actor with regard to an action. Based on the FOLIO ontology, a `Seller` is the person or organization that can, through a norm expressed in the law, have the right to supply the goods or services.[257] On the other side of that transaction

[256] "A right is the legal or moral entitlement to do or refrain from doing something or to obtain or refrain from obtaining an action, thing or recognition in civil society." ESTRELLA Consortium, *Legal Knowledge Interchange Format (LKIF) - Core Ontology*, See #Right; "entitlement (not) to perform certain actions, or (not) to be in certain states; or entitlement that others (not) perform certain actions or (not) be in certain states" EDM Council, *The Financial Industry Business Ontology*, See also.

[257] "A seller is a person, company or entity who sells a thing or property in exchange for other property (often money). One who disposes of a thing in consideration of money." The Institute for the Advancement of Legal and Ethical AI, *Federated Online Legal Information Ontology (FOLIO)*, See "Seller".

is the `Buyer`[258]. Both `Seller` and `Buyer` are $\sqsubseteq Persons \sqsubseteq Agent$, with sellers also included in $\sqsubseteq Organization \sqsubseteq Persons$. Although sellers and buyers share the same ancestors (they are both persons and agents), the domain does not contemplate that individuals can both be buyers and sellers, hence these two subclasses are **disjoint** ($Buyer \sqcap Seller \sqsubseteq \bot$) - that is, the intersection of these two classes will always be empty.[259]

Figure 7: Class Hierarchy of Participants in A1



Interaction of Concepts As discussed above, DL semantics can be elaborated not just through IS-A relationships between classes and subclasses, but through roles and their restrictions. Rule A1 contemplates that the sellers are in a position to sell `Goods` or `Services` to buyers. To represent these relationships, we can model the action of selling through a class called `Sale`. For every individual sales transaction, there is a seller, a good or service sold, and a buyer. The sale of goods can thus be represented through binary relationships with these other concepts:

[258] "A buyer or purchaser generally means a person, company or entity contracting with a seller for services or merchandise to be provided or delivered for a named individual." ibid., See "Buyer".
[259] $\bot$ is a special concept in DL, representing the class with no inviduals as members

$$Sale \sqsubseteq \exists hasSeller.Seller$$

$$Sale \sqsubseteq \exists hasGoodSold.Good$$

$$Sale \sqsubseteq \exists hasBuyer.Buyer$$

To further clarify the semantics of these relationships, we can also define the inverses of these roles, as reckoned from the objects of these transactions. Thus:

$$makesSale \equiv hasSeller^{-}$$

$$isGoodSoldIn \equiv hasGoodSold^{-}$$

$$participateInSale \equiv hasBuyer^{-}$$

Provide counterpart relationships linking each seller, good, and buyer, to a specific sales transaction.

Representing Exclusivity Each `Seller` participates in a `Market` where such seller can offer goods and services to multiple buyers. The rule in A1 prohibits (or at least negatively evaluates) a legal environment that allows any given seller to be the $Exclusive\_Seller$. We define an exclusive seller in terms of its participation in a $Monopoly\_Market$. That is, a market that only has one seller:

$$Exclusive\_Seller \equiv Seller \sqcap (\exists participates\_in.Monopoly\_Market)$$

$$Monopoly\_Market \equiv Market \sqcap (= 1 has\_participant.Seller)$$

This is a very coarse, if not naive view of what it means to be an exclusive supplier for the purpose of competition impact analysis. There may be few purely monopolistic markets, and the OECD Guidlines are actually more concerned with manifestations of **market power**. Reckoned from the supplier side, this could mean the "the ability to profitably increase price, decrease quality, or decrease innovation relative to the levels that would prevail in a competitive market".[260] Although market power may elude precise legal definitions, there are assertions in the economic literature that can serve as the basis of an initial ontological encoding. There is the traditional line of thinking that evaluates market power bases on the extent of horizontal competition. That is, "competition from established or potential rivals within a relevant market".[261] This is often focused whether or not any particular firm has the ability to set prices.This emphasis on the price setting power is incomplete, however and fails to give a satisfactory account of business models in a digital economy - not when the price is zero for the final consumer is zero for the final consumer, which in effect is subsidized by attention and data markets.[262] Lianos(2021), outlines possible alternative approaches based on either:1. The potential for coercion between market participants[263] and 2. A process-based definition, based on identifying specific situations of assymetry.[264]

Normative Encoding of Rule A1 The classes and properties in the ontology can then be used as the variables and predications in the norm, expressed as Reified I/O Logic formulae. In the case of Rule A1:

[260] OECD, *Competition Assessment Toolkit - Volume 1 (Principles)*, at 10, footnote 2.

[261] Ioannis Lianos and Bruno Carballa. "Economic Power and New Business Models in Competition Law and Economics: Ontology and New Metrics". In: *Center for Law Economics & Society Research Paper Series* (Mar. 2021). ISSN: 978-1-910801-37-6. URL: https://www.ucl.ac.uk/cles/research-papers (visited on 05/08/2025), at 6.

[262] Ibid., at 8.

[263] Coercion can still be vaguely defined, but Lianos proposes some possible operational bases: I.e., There is coercion when the choice forced upon the coercee is such that she has no reasonable choice but to accept it. The absence of choice will require consideration of: 1. The relative bargaining position of the parties. 2. The history of their interactions. ibid., at 10.

[264] Ibid., at 11.

$$
\begin{aligned}
&\forall_x \forall_t (\\
&\exists_{e_a} \exists_{e_{xs}} \exists_{s_1} [(RexistAtTime\ e_a\ t) \wedge \\
&(allows'\ e_a\ e_{xs}\ x) \wedge (exsupplier'\ e_{xs}\ s1) \wedge \\
&(supplier\ s_1)], \\
&\exists_{e_f} [(RexistAtTime\ e_f\ t) \wedge \\
&(flagged'\ e_f\ x)]) \in O
\end{aligned}
$$

This translates to an obligation to flag a law if such a law allowws any supplier to become an exclusive supplier (however it may be defined in the ontology).

Modelling Exceptions The OECD guideline provide an exception to the negative evaluation of exclusive suppliers. That is, of the exclusive supplier is a natural monopoly. Assuming the concept can be encoded into the ontology, we can then model it as an exception to the normative encoding:

1. A law that allows a supplier to become an exclusive supplier ought to be flagged
2. BUT: If the exclusive supplier is a natural monopoly, then the law should not be flagged

In Reified I/O logic, exceptions are represented through Hobbs and Gordon's constructs for representing defeasibility: There are general predicates, and special "blocking" predicates that apply in the event of the exception. That is, the general inferences apply if more specific exceptions do not occur. The system for exceptions employ negation-as-failure - $naf(X)$, supported in LegalRuleML. That is, the predication $naf(X)$ is true when it cannot be derived that X is true (it is either false or unknown).[265] The exception, as the more

[265] Negation as failure - A rule of inference where if a starting proposition P cannot be proven to be true based on the available knowledge and after an exhaustive search, then its negation (not true) is considered to be true. Contrast this with strong negation (or classical negation), where $notP$ can only be concluded if there is explicit evidence or a direct derivation proving that P is

specialized situation, is presumed to operate by default. Only when the exception does not occur will the general rule become applicable. For Rule A1, we can represent the exception as $naf(exceptionNaturalMonopoly\ s_1)$, which is true if the supplier s_1 is not a natural monopoly. Accounting for the exception, the encoding for A1 will now be:

$$\begin{aligned}
&\forall_x \forall_t (\\
&\exists_{e_a} \exists_{e_{xs}} \exists_{s_1} [(RexistAtTime\ e_a\ t) \wedge \\
&(allows'\ e_a\ e_{xs}\ x) \wedge (exsupplier'\ e_{xs}\ s1) \wedge \\
&naf(exceptionNaturalMonopoly\ s_1) \wedge (supplier\ s_1)], \\
&\exists_{e_f} [(RexistAtTime\ e_f\ t) \wedge \\
&(flagged'\ e_f\ x)]) \in O
\end{aligned}$$

Modelling Interpretations Legal texts may contain ambiguous terms that can be subject to multiple, often conflicting interpretations. For example, we have previously discussed that in the context of competition law, "exclusivity" can be correlated with market power, which in turn can be viewed from multiple aspects:

- **Pricing Power View** - A firm has market power if it can set prices;
- **Coercion View** - A firm has market power if it can deny competitors reasonable choices;
- **Process View** - A firm has market power if it has bargaining power, especially related to the allocation of surplus.

We can capture the proposition that exclusivity and pricing power are correlated through a constitutive rule:

false. Robaldo, Bartolini, and Lenzini, "The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML", at 5693.

$$
\begin{aligned}
&\forall_{e_{xs}} \forall_t (\\
&\exists_f [(RexistAtTime\ e_{xs}\ t) \wedge \\
&(exsupplier'\ e_{xs}\ f)], \\
&(PricingPower\ e_{xs})) \in C
\end{aligned}
$$

As in the case of exceptions, the representation of proposed interpretations is done through a special predicate $(assumption\ e_{xs})$, which is true if it can be assumed that the previous formula is valid:

$$
\begin{aligned}
&\forall_{e_{xs}} \forall_t (\\
&\exists_f [(RexistAtTime\ e_{xs}\ t) \wedge \\
&(exsupplier'\ e_{xs}\ f) \wedge (assumption\ e_{xs})], \\
&(PricingPower\ e_{xs})) \in C
\end{aligned}
$$

Differing interpretations as to the meaning of exclusivity can then be represented based on whether or not $(assumption\ e_{xs})$ holds or is negated. hus, an interpretation that exclusivity should not be understood in terms of pricing power can simply negate the assumption:

$$
\begin{aligned}
&\forall_{e_{xs}} \forall_t (\\
&\exists_f [(RexistAtTime\ e_{xs}\ t) \wedge \\
&(exsupplier'\ e_{xs}\ f)], \\
&\neg(assumption\ e_{xs})) \in C
\end{aligned}
$$

Combining the mechanism for exceptions, we can model another possible interpretation where:

1. Exclusivity means pricing power in the general case;
2. Firms in the digital sector are an exception, where exclusivity means the ability to coerce possible competitors.

The general case is encoded as a constitutive rule where $(assumption\ e_{xs})$ and $naf(exceptionDigitalSector$ both hold true:

$$\begin{aligned}&\forall_{e_{xs}}\forall_t(\\&\exists_f[(RexistAtTime\ e_{xs}\ t)\wedge\\&(exsupplier'\ e_{xs}\ f)\wedge naf(exceptionDigitalSector\ e_{xs})],\\&(assumption\ e_{xs}))\in C\end{aligned}$$

On the other hand, the exceptional case has additional predicates to reflect the situation where a firm is part of the digital sector:

$$\begin{aligned}&\forall_{e_{xs}}(\\&\exists_f\exists_t[(RexistAtTime\ e_{xs}\ t)\wedge\\&(exsupplier'\ e_{xs}\ f)\wedge(partOf\ e_{xs}\ e_d)\wedge(digitalSector\ e_d)],\\&(exceptionDigitalSector\ e_{xs}))\in C\end{aligned}$$

Conclusion This chapter provided details on the formalization of a specific rule in the OECD Guidelines, using the methodologies outlined om the previous two chapters. Although both the ontological and normative layers of the encoding are implemented in software, a logical account was necessary to describe the structure and content of the encoding in a human-readable fashion, as well as demonstrate the feasibility of using formalizations for the domain of competition law.

The knowledgebase, as of time of writing, encompasses both the ontological and normative encoding of rules A1 to A5 of the OECD Guidelines. From this point, further research can be performed to expand the coverage to the rest of the guidelines. Alternatively, future work can focus on enhancing the granularity and resolution of the encoding, through the definition of additional concepts, exceptions and interpretations.

7 Bibliography

Alarie, Benjamin. “The Path of the Law: Toward Legal Singularity”. In: *University of Toronto* 66.4 (2016), pp. 443–445. ISSN: 1556-5068. DOI: `\https://doi.org/10.3138/UTLJ.4008`.

Aldisert, Ruggero, Stephen Clowney, and Jeremy Peterson. “Logic for Law Students: How to Think Like a Lawyer”. In: *University of Pittsburgh Law Review* 69 (2007), pp. 1–22.

Andersson, Hans. “Computational Law: Law That Works Like Software”. CodeX – The Stanford Center for Legal Informatics, Feb. 10, 2014. URL: `https://www.academia.edu/9286857/Computational_Law_Anderrson_and_Lee`.

Armgardt, Matthias. “Leibniz as a Legal Scholar”. In: *Fundamina* (2014), pp. 27–38.

Armgardt, Matthias, Patrice Canivez, and Sandrine Chassagnard-Pinet, eds. *Past and Present Interactions in Legal Reasoning and Logic*. Vol. 7. Logic, Argumentation & Reasoning. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-16020-7 978-3-319-16021-4. DOI: `10.1007/978-3-319-16021-4`. URL: `https://link.springer.com/10.1007/978-3-319-16021-4` (visited on 03/16/2024).

Artosi, Alberto and Giovanni Sartor. “Leibniz as Jurist”. In: *The Oxford Handbook of Leibniz*. Ed. by Maria Rosa Antognazza. Oxford University Press, Dec. 27, 2018, pp. 640–663. ISBN: 978-0-19-974472-5. DOI: `10.1093/oxfordhb/9780199744725.013.38`. URL: `https://academic.oup.com/edited-volume/34667/chapter/295400716` (visited on 10/10/2023).

Ashley, Kevin et al. “Legal Reasoning and Artificial Intelligence: How Computers ”Think” Like Lawyers”. In: *University of Chicago Law School Roundtable* 8.1 (2001), pp. 1–28.

Athan, Tara et al. “OASIS LegalRuleML”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*. ICAIL ’13: International

Conference on Artificial Intelligence and Law. Rome Italy: ACM, June 10, 2013, pp. 3–12. ISBN: 978-1-4503-2080-1. DOI: 10.1145/2514601.2514603. URL: https://dl.acm.org/doi/10.1145/2514601.2514603 (visited on 12/16/2024).

Bayon, Juan Carlos. “Why Is Legal Reasoning Defeasible?” In: *Diritto & Questioni Pubbliche* 2 (2002), pp. 1–18.

Bhuiyan, Hanif et al. “A Methodology for Encoding Regulatory Rules”. In: *Proceedings of the 4th International Workshop on MIning and REasoning with Legal Texts Co-Located with the 32nd International Conference on Legal Knowledge and Information Systems (JURIX 2019)*. International Workshop on MIning and REasoning with Legal Texts 2019. Vol. 2632. Madrid, Spain: Rheinisch-Westfaelische Technische Hochschule Aachen, Dec. 11, 2019.

Bhuiyan, Hanif et al. “Traffic Rules Encoding Using Defeasible Deontic Logic”. In: *Frontiers in Artificial Intelligence and Applications*. Ed. by Serena Villata, Jakub Harašta, and Petr Křemen. IOS Press, Dec. 1, 2020. ISBN: 978-1-64368-150-4 978-1-64368-151-1. DOI: 10.3233/FAIA200844. URL: http://ebooks.iospress.nl/doi/10.3233/FAIA200844 (visited on 11/23/2024).

Burgin, Mark. “Evolution of logic as an information processing mechanism in advanced biological systems”. In: *Bio Systems* 221 (2022), p. 104758. ISSN: 0303-2647. DOI: 10.1016/j.biosystems.2022.104758. URL: https://www.sciencedirect.com/science/article/pii/S0303264722001393.

Competition & Markets Authority. *Competition Assessment Guidelines Part 2: Guidelines*. UK Government, 2023. URL: https://www.gov.uk/government/publications/competition-impact-assessment-guidelines-for-policymakers (visited on 01/10/2024).

— *Competition Impact Assessment Part 1: Overview*. UK Government, Sept. 15, 2015. URL: https://www.gov.uk/government/publications/competition-impact-assessment-guidelines-for-policymakers (visited on 01/10/2024).

De Bellis, Michael. *A Practical Guide to Building OWL Ontologies*. Oct. 8, 2021. URL: https://www.michaeldebellis.com/post/new-protege-pizza-tutorial (visited on 01/31/2024).

Deakin, Simon F. and Christopher Markou. "From Rule of Law to Legal Singularity". In: *Is Law Computable? Critical Perspectives on Law and Artificial Intelligence*. Oxford: Hart Publishing, 2020, pp. 1–29.

Duso, Tomaso, Jo Seldeslachts, and Florian Szücs. "The Impact of Competition Policy Enforcement on the Functioning of EU Energy Markets". In: *The Energy Journal* 40.5 (Sept. 2019), pp. 97–120. ISSN: 0195-6574, 1944-9089. DOI: 10.5547/01956574.40.5.tdus. URL: http://journals.sagepub.com/doi/10.5547/01956574.40.5.tdus (visited on 01/08/2024).

EDM Council. *The Financial Industry Business Ontology*. FIBO. Jan. 2020. URL: https://spec.edmcouncil.org/fibo/ (visited on 01/18/2024).

— *The Financial Industry Business Ontology*. FIBO. URL: https://spec.edmcouncil.org/fibo/ (visited on 01/18/2024).

ESTRELLA Consortium. *Legal Knowledge Interchange Format (LKIF) - Core Ontology*. Version 1.1. 2008. URL: http://www.estrellaproject.org/lkif-core.

European Bioinformatics Institute (EMBL-EBI). *Ontology Search*. Ontology Lookup Service (OLS). URL: https://www.ebi.ac.uk/ols4/ontologies (visited on 01/18/2024).

European Commission. *Statistical Classification of Economic Activities in the European Union*. Rev. 2. 2008. URL: https://ec.europa.eu/eurostat/documents/3859598/5902521/KS-RA-07-015-EN.PDF (visited on 05/13/2024).

Feynman, Richard P. *Feynman Lectures on Computation*. Ed. by Anthony J. G. Hey and Robin W. Allen. Boca Raton: CRC Press, 2018. 303 pp. ISBN: 978-0-7382-0296-9.

Freeman, Kathleen and Arthur M. Farley. "A Model of Argumentation and Its Application to Legal Reasoning". In: *Artificial Intelligence and Law* 4.3-4 (1996), pp. 163–197.

Genesereth, Michael. *Computational Law: The Cop in the Back Seat*. CodeX: The Center for Legal Informatics Stanford University. 2015. URL: https://law.stanford.edu/publications/computational-law-the-cop-in-the-backseat/ (visited on 09/18/2021).

Genesereth, Michael and Nathaniel Love. "Computational Law". In: *Proceedings of the 10th International Conference on Artificial Intelligence and Law - ICAIL '05*. The 10th International Conference. Bologna, Italy: ACM Press, 2005. ISBN: 978-1-59593-081-1. DOI: 10.1145/1165485.1165517. URL: http://portal.acm.org/citation.cfm?doid=1165485.1165517 (visited on 09/18/2021).

Genesereth, Michael and Michael Thielscher. *General Game Playing*. Red. by Ronald J. Brachman, William W. Cohen, and Peter Stone. Synthesis Lectures on Artificial Intelligence and Machine Learning 24. Morgan & Claypool Publishers, 2014.

Goldstein, Rebecca. *Incompleteness: The Proof and Paradox of Kurt Godel*. New York, London: Atlas Books, 2005.

Goldsworthy, Jeffrey. "The Limits of Judicial Fidelity to Law: The Coxford Lecture". In: *Canadian Journal of Law and Jurisprudence* 24.2 (July 2011), pp. 305–325. DOI: 10.1017/S084182090000518X.

Gordon, Andrew S. and Jerry R. Hobbs. *A Formal Theory of Commonsense Psychology: How People Think People Think*. 1st ed. Cambridge University Press, Aug. 31, 2017. ISBN: 978-1-107-15100-0 978-1-316-58470-5. DOI: 10.1017/9781316584705. URL: https://www.cambridge.org/core/product/identifier/9781316584705/type/book (visited on 12/05/2024).

Grossman, Garry S and Lewis D Solomon. "Computers and Legal Reasoning". In: *ABA Journal* 69 (1983), pp. 66–70.

Guest, Anthony G. “Logic in the Law”. In: *Oxford Essays in Jurisprudence*. Oxford: Oxford University Press, 1961, pp. 176–197.

Haack, Susan. “On Logic in the Law: “Something, but Not All””. In: *Ratio Juris* 20.1 (2007), pp. 1–31. ISSN: 0952-1917, 1467-9337. DOI: 10.1111/j.1467-9337.2007.00330.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9337.2007.00330.x (visited on 03/26/2024).

Hage, Jaap. “A Theory of Reasoning and a Logic to Match”. In: *Artificial Intelligence and Law* 4.3-4 (1996), pp. 199–273.

Halper, Thomas. “Logic in Judicial Reasoning”. In: *Indiana Law Journal* 44.1 (1968), pp. 33–48.

Hampshire, Stuart, ed. *Public and Private Morality*. Cambridge: Cambridge University Press, 1991. 143 pp. ISBN: 978-0-521-22084-2 978-0-521-29352-5.

Hasan, Iftekhar and Matej Marinč. “Should Competition Policy in Banking Be Amended during Crises? Lessons from the EU”. In: *European Journal of Law and Economics* 42.2 (Oct. 2016), pp. 295–324. ISSN: 0929-1261, 1572-9990. DOI: 10.1007/s10657-013-9391-2. URL: http://link.springer.com/10.1007/s10657-013-9391-2 (visited on 01/08/2024).

Hawkins, Brian. “The Life of the Law: What Holmes Meant”. In: *Whittier Law Review* 33 (Winter Issue 2012), pp. 323–370.

Hinson, Christopher L. “Legal Informatics: Opportunities for Information Science”. In: *Journal of Education for Library and Information Science* 46.2 (2005), p. 134. ISSN: 07485786. DOI: 10.2307/40323866. JSTOR: 10.2307/40323866. URL: https://www.jstor.org/stable/10.2307/40323866?origin=crossref (visited on 11/02/2023).

Hoekstra, Rinke et al. “The LKIF Core Ontology of Basic Legal Concepts”. In: *Proceedings of LOAIT 07*. II Workshop on Legal Ontologies and Artificial Intelligence Techniques. 2007, p. 43–63.

Hülser, Karlheinz. “Proculus on the Meaning of OR and the Types of Disjunction”. In: *Past and Present Interactions in Legal Reasoning and Logic*. Springer International Publishing, 2015, pp. 7–30.

ICN Advocacy Working Group. “Framework of Competition Assessment Regimes”. In: ICN 14th Annual Conference. Sydney, Apr. 2015. URL: https://www.internationalcompetit org/wp-content/uploads/2018/09/AWG_FrameworkCompetitionAssessmentRegimes.pdf (visited on 10/10/2023).

— *Recommended Practices on Competition Assessments*. International Competition Network, 2014. URL: https://www.internationalcompetitionnetwork.org/wp-content/uploads/2018/07/AWG_RP_English.pdf (visited on 01/10/2024).

Jean-Baptiste, Lamy. *Ontologies with Python: Programming OWL 2.0 Ontologies with Python and Owlready2*. Berkeley, CA: Apress, 2021. ISBN: 978-1-4842-6551-2 978-1-4842-6552-9. DOI: 10.1007/978-1-4842-6552-9. URL: http://link.springer.com/10.1007/978-1-4842-6552-9 (visited on 04/03/2024).

Kowalski, Robert and Marek Sergot. “The Use of Logical Models in Legal Problem Solving”. In: *Ratio Juris* 3.2 (July 1990), pp. 201–218.

Krötzsch, Markus, Frantisek Simancik, and Ian Horrocks. “A Description Logic Primer”. In: *IEEE Intelligent Systems* 29.1 (2014), pp. 12–19. DOI: 10.48550/arXiv.1201.4089. arXiv: 1201.4089 [cs]. URL: http://arxiv.org/abs/1201.4089 (visited on 06/20/2025).

Leith, Philip. “Logic, Formal Models and Legal Reasoning”. In: *Jurimetrics* 24.4 (1984), pp. 334–356.

Lianos, Ioannis and Bruno Carballa. “Economic Power and New Business Models in Competition Law and Economics: Ontology and New Metrics”. In: *Center for Law Economics & Society Research Paper Series* (Mar. 2021). ISSN: 978-1-910801-37-6. URL: https://www.ucl.ac.uk/cles/research-papers (visited on 05/08/2025).

Lovevinger, Lee. “An Introduction to Legal Logic”. In: *Indiana Law Journal* 27.4 (Sum. 1952), pp. 471–522.

Ma, Megan. “The Law’s New Language”. In: *Harvard International Law Journal Frontiers* 61 (2020), pp. 1–9.

Markou, Christopher and Simon Deakin. “Ex Machina Lex: Exploring the Limits of Legal Computability”. In: *Is Law Computable? Critical Perspectives on Law and Artificial Intelligence*. 2020, pp. 31–65.

McNamara, Paul and Frederik Van De Putte. “Deontic Logic”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2022. Metaphysics Research Lab, Stanford University, 2022. URL: https://plato.stanford.edu/archives/spr2022/entries/logic-deontic/ (visited on 06/08/2022).

Medalla, Erlinda M. “Understanding the New Philippine Competition Act”. In: *Philippine Institute for Development Studies (PIDS) Discussion Paper Series* (No. 2017-14 2017), pp. 1–24. URL: http://hdl.handle.net/10419/173591 (visited on 01/08/2024).

Michael, Jerome and Mortimer J. Adler. “The Trial of an Issue of Fact: I”. In: *Columbia Law Review* 34.7 (Nov. 1934), pp. 1224–1306. ISSN: 00101958. DOI: 10.2307/1116103. JSTOR: 1116103. URL: https://www.jstor.org/stable/1116103?origin=crossref (visited on 12/11/2024).

Modgil, Sanjay and Henry Prakken. “The ASPIC+framework for Structured Argumentation: A Tutorial”. In: *Argument & Computation* 5.1 (Jan. 2, 2014), pp. 31–62. ISSN: 1946-2166, 1946-2174. DOI: 10.1080/19462166.2013.869766. URL: http://content.iospress.com/doi/10.1080/19462166.2013.869766 (visited on 11/27/2023).

Musen, Mark A. “The protégé project: a look back and a look forward”. In: *AI Matters* 1.4 (2015), pp. 4–12. DOI: 10.1145/2757001.2757003. URL: https://doi.org/10.1145/2757001.2757003.

Nakaizumi, Takuya. *Impact Assessment for Developing Countries: A Guide for Government Officials and Public Servants*. Contributions to Economics. Singapore: Springer Nature Singapore, 2022. ISBN: 978-981-19549-3-1 978-981-19549-4-8. DOI: 10.1007/978-981-19-5494-8. URL: https://link.springer.com/10.1007/978-981-19-5494-8 (visited on 07/14/2024).

Nardi, Daniele and Ronald J. Brachman. "An Introduction to Description Logics". In: *The Description Logic Handbook*. Cambridge University Press, 2007, pp. 1–43.

Navarro, Pablo E. and Jorge L. Rodríguez. *Deontic Logic and Legal Systems*. New York: Cambridge University Press, Sept. 29, 2014. ISBN: 978-0-521-76739-2. DOI: 10.1017/CBO9781139032711.

Noy, Natalya F and Deborah L McGuinness. "Ontology Development 101: A Guide to Creating Your First Ontology". In: *Stanford Medical Informatics Technical Report* (SMI-2001-0880 Mar. 2001). URL: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

Nute, Donald, ed. *Defeasible Deontic Logic*. Dordrecht: Springer Netherlands, 1997. ISBN: 978-90-481-4874-5 978-94-015-8851-5. DOI: 10.1007/978-94-015-8851-5. URL: http://link.springer.com/10.1007/978-94-015-8851-5 (visited on 11/23/2024).

Oasis Open. *LegalRuleML Core Specification Version 1.0*. Aug. 30, 2021. URL: http://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/legalruleml-core-spec-v1.0.html (visited on 10/06/2023).

OECD. *Competition Assessment Reviews: Logistics Sector in the Philippines*. 2020. URL: https://www.oecd.org/daf/competition/oecd-competition-assessment-reviews-philippines-2020.pdf (visited on 10/10/2023).

— *Competition Assessment Toolkit - Volume 1 (Principles)*. 2019. URL: https://www.oecd.org/daf/competition/46193173.pdf (visited on 10/10/2023).

OECD. *Competition Assessment Toolkit - Volume 2 (Guidance)*. 2019. URL: https://www.oecd.org/daf/competition/45544507.pdf (visited on 10/10/2023).

— *Competition Assessment Toolkit - Volume 3 (Operations Manual)*. 2019. URL: https://web-archive.oecd.org/2020-01-22/370055-COMP_Toolkit_Vol.3_ENG_2019.pdf (visited on 10/10/2023).

— *Competitive Neutrality Reviews: Small-Package Delivery Services in the Philippines*. 2020. URL: https://www.oecd.org/daf/competition/oecd-competitive-neutrality-reviews-philippines-2020.pdf (visited on 10/10/2023).

Pound, Roscoe. “Spurious Interpretation”. In: *Columbia Law Review* 7.6 (June 1907), p. 379. ISSN: 00101958. DOI: 10.2307/1109940. JSTOR: 1109940. URL: https://www.jstor.org/stable/1109940?origin=crossref (visited on 04/28/2024).

Prakken, Henry and Giovanni Sartor, eds. *Logical Models of Legal Argumentation*. Netherlands: Kluwer Academic Publishers, 1997. ISBN: 0-7923-4413-8. DOI: 10.1007/978-94-011-5668-4.

Robaldo, Livio, Cesare Bartolini, and Gabriele Lenzini. “The DAPRECO Knowledge Base: Representing the GDPR in LegalRuleML”. In: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*. Marseille, May 11–16, 2020, pp. 5688–5697.

Robaldo, Livio and Xin Sun. “Reified Input/Output Logic: Combining Input/Output Logic and Reification to Represent Norms Coming from Existing Legislation”. In: *Journal of Logic and Computation* 27.8 (Dec. 1, 2017), pp. 2471–2503. ISSN: 0955-792X, 1465-363X. DOI: 10.1093/logcom/exx009. URL: http://academic.oup.com/logcom/article/27/8/2471/3098296 (visited on 12/16/2024).

Robaldo, Livio et al. “Formalizing GDPR Provisions in Reified I/O Logic: The DAPRECO Knowledge Base”. In: *Journal of Logic, Language and Information* 29.4 (Dec. 2020), pp. 401–449. ISSN: 0925-8531, 1572-9583. DOI: 10.1007/s10849-019-

09309-z. URL: http://link.springer.com/10.1007/s10849-019-09309-z (visited on 11/25/2024).

Robins, Nicole and Hannes Geldof. "Ex Post Assessment of the Impact of State Aid on Competition". In: *European State Aid Law Quarterly* 17.4 (2018), pp. 494–508. ISSN: 16195272, 21908184. DOI: 10.21552/estal/2018/4/6. URL: http://estal.lexxion.eu/article/ESTAL/2018/4/6 (visited on 01/03/2024).

Samuel, Geoffrey. "Is Legal Reasoning like Medical Reasoning?" In: *Legal Studies* 35.2 (2015), pp. 323–347. ISSN: 0261-3875, 1748-121X. DOI: 10.1111/lest.12063. URL: https://www.cambridge.org/core/product/identifier/S026138750000581X/type/journal_article (visited on 12/18/2023).

Sartor, Giovanni. "A Formal Model of Legal Argumentation". In: *Ratio Juris* 7.2 (July 1994), pp. 177–211. ISSN: 0952-1917, 1467-9337. DOI: 10.1111/j.1467-9337.1994.tb00175.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9337.1994.tb00175.x (visited on 06/20/2023).

— "Defeasibility in Legal Reasoning". In: *Rechtstheorie* 24.3 (1993), pp. 281–316.

— "Legal Concepts as Inferential Nodes and Ontological Categories". In: *Artificial Intelligence and Law* 17.3 (Aug. 21, 2009), pp. 217–251. ISSN: 0924-8463, 1572-8382. DOI: 10.1007/s10506-009-9079-7. URL: http://link.springer.com/10.1007/s10506-009-9079-7 (visited on 06/20/2023).

Searle, John R. *The Construction of Social Reality*. New York: The Free Press, 1995.

Serafica, Ramonette B. et al. "Issues Paper on the Philippine Digital Commerce Market". In: *PCC Issues Paper*. 2020th ser. 3 (2020). URL: https://www.phcc.gov.ph/wp-content/uploads/2020/07/PCC-Issues-Paper-2020-03-Issues-Paper-on-the-Philippine-Digital-Commerce-Market.pdf.

Summers, Robert S. "Logic in the Law". In: *Cornell Law Faculty Publications* (Paper 1133 1963), pp. 254–258. URL: http://scholarship.law.cornell.edu/facpub/1133.

The Institute for the Advancement of Legal and Ethical AI. *Federated Online Legal Information Ontology (FOLIO)*. FOLIO - Federated Online Legal Information Ontology (FOLIO). 2024. URL: https://folio.openlegalstandard.org/taxonomy/browse (visited on 04/28/2025).

The World Bank. *Markets and Competition Policy*. World Bank. URL: https://www.worldbank.org/en/topic/competition-policy (visited on 01/16/2024).

Tribe, Laurence H. "The Curvature of Constitutional Space: What Lawyers Can Learn from Modern Physics". In: *Harvard Law Review* (Nov. 1989), pp. 1–68.

Turing, Alan M. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society* 2.42 (1937), pp. 230–265. DOI: 10.1112/plms/s2-42.1.230.

United Nations. *International Standard Industrial Classification of All Economic Activities (ISIC)*. Revision 4. United Nations, 2008. ISBN: 978-92-1-161518-0. URL: https://unstats.un.org/unsd/publication/seriesm/seriesm_4rev4e.pdf.

Van Eemeren, Frans H. et al. *Handbook of Argumentation Theory*. Dordrecht: Springer Netherlands, 2014. ISBN: 978-90-481-9472-8 978-90-481-9473-5. DOI: 10.1007/978-90-481-9473-5. URL: https://link.springer.com/10.1007/978-90-481-9473-5 (visited on 06/20/2023).

Von Wright, G.H. "Deontic Logic". In: *Mind* 60.237 (Jan. 1951), pp. 1–15.

W3C. *RuleML - W3C RIF-WG Wiki*. 2005. URL: https://www.w3.org/2005/rules/wg/wiki/RuleML (visited on 01/12/2025).

Wikimedia Foundation. *Wikidata*. URL: https://www.wikidata.org/wiki/Wikidata:Main_Page (visited on 01/18/2024).

Wolfram, Stephen. *A New Kind of Science*. Champaign, Illinois: Wolfram Media, 2002. 1197 pp. ISBN: 978-1-57955-008-0.

— *A Project to Find the Fundamental Theory of Physics*. Champaign, Illinois: Stephen Wolfram, LLC, 2020. 770 pp. ISBN: 978-1-57955-035-6.

Wolfram, Stephen. "AI Law and Computational Irreducibility". FutureLaw 2023, Stanford Law School. Apr. 25, 2023. URL: https://www.youtube.com/watch?v=8oG1FidVE2o (visited on 01/14/2024).

— *Computational Law, Symbolic Discourse and the AI Constitution*. Stephen Wolfram Writings. Oct. 12, 2016. URL: https://writings.stephenwolfram.com/2016/10/computational-law-symbolic-discourse-and-the-ai-constitution/ (visited on 01/14/2024).

— "Computational Law, Symbolic Discourse and the AI Constitution". In: *Data-Driven Law: Data Analytics and the New Legal Services*. Ed. by Ed Walters. Red. by Jay Liebowitz. Data Analytics Applications. Boca Raton, FL: CRC Press Taylor & Francis Group, 2019, pp. 144–174. ISBN: 13: 978-1-4987-6665-4. URL: https://writings.stephenwolfram.com/2016/10/computational-law-symbolic-discourse-and-the-ai-constitution/ (visited on 01/14/2024).

— *How to Think Computationally about AI, the Universe and Everything*. Stephen Wolfram Writings. Oct. 27, 2023. URL: https://writings.stephenwolfram.com/2023/10/how-to-think-computationally-about-ai-the-universe-and-everything/ (visited on 12/14/2023).

World Bank Group. *Enhancing Competition Conditions and Competitiveness of Philippine Domestic Shipping*. World Bank, Sept. 2014. DOI: 10.1596/24800. URL: http://elibrary.worldbank.org/doi/book/10.1596/24800 (visited on 08/18/2021).