NAI - The Normative Reasoner

Tomer Libal
The American University of Paris
Paris, France
tlibal@aup.edu

Alexander Steen
University of Luxembourg
Esch-sur-Alzette, Luxembourg
alexander.steen@uni.lu

CCS CONCEPTS

• **Human-centered computing → Graphical user interfaces**; • **Theory of computation →** *Automated reasoning*; Modal and temporal logics; • **Applied computing →** *Law*.

KEYWORDS

Deontic logic, Legal reasoning, Automated reasoning, Graphical user interfaces

ACM Reference Format:
Tomer Libal and Alexander Steen. ???. NAI - The Normative Reasoner. In *???*. ACM, New York, NY, USA, 2 pages. https://doi.org/???

Introduction. Computer systems are playing a substantial role in assisting people in a wide range of tasks, including search in large data and decision-making; and their employment is progressively becoming vital in an increasing number of fields. One of these fields is *legal reasoning*: New court cases and legislations are accumulated every day. In addition, international organizations like the European Union are constantly aiming at combining and integrating separate legal systems [4]. In contrast to this situation, the automation of legal reasoning is still underdeveloped albeit being an growing field of research. In recent years automatic procedures, e.g. for courtroom management[1] and legal language processing/management [3], expert systems based on cases or rules [10], and normative compliance tools[2] have been introduced. Additionally, logical systems for automatic reasoning over sets of norms have been developed, such as for the HIPAA and GLBA privacy laws [5], for business [6] and the GDPR [9].

In this extended abstract we describe the new normative reasoning framework NAI that addresses the normalization of legal texts and automated reasoning over them. NAI is a web application and is readily available at https://nai.uni.lu. NAI is also open-source, its source code is freely available at GitHub[3] under Apache license.

Goals and Architecture of NAI. The NAI framework aims at integrating novel theorem proving technology into a usable graphical user interface for computer-assisted normalization of legal texts and normative reasoning. In particular, if offers

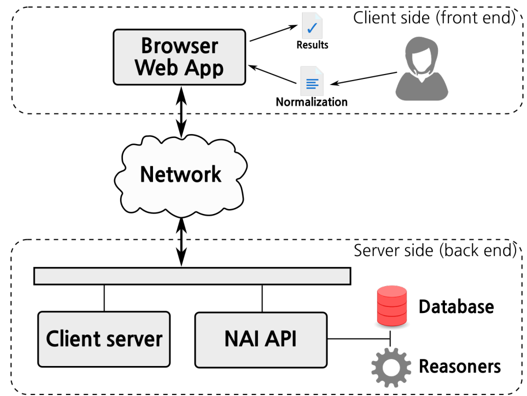

Figure 1: SaaS architecture of the NAI reasoning framework.

(1) means of assessing the quality of entered normalizations, e.g. by automatically conducting consistency checks and proving logical independence,
(2) ready-to-use theorem prover technology for evaluating user-specified queries wrt. a given normalization, and
(3) the possibility to share and collaborate, and to experiment with different normalizations and underlying logics.

NAI is realized using a web-based Software-as-a-Service architecture, cf. Fig. 1, comprising of a graphical user interface, implemented as a Javascript browser application, and a NodeJS application on the back-end side, connecting to theorem provers, data storage and relevant middleware. This way no further software is required from the user perspective for using NAI and its reasoning procedures, as all computationally heavy tasks are executed on the remote servers.

Logic and legal foundations. A very simple logical system dealing with normative concepts is **SDL** (*Standard Deontic Logic*). This system can be easily encoded in theorem provers but is affected by seemingly unsurmountable difficulties in representing very common normative scenarios, such as those in which a contrary-to-duty norm applies [8]. On the other hand, there are extensions of **SDL** that are still fairly simple but do not share most of its weaknesses. One such extension is the system developed in [7]. This system balances effectively between normative expressivity and automation. Moreover, its language is simple enough and can therefore serve as a formal layer for the normalization of legal text. In the current version of the tool, this system serves as the underlying logical language.

Using a formal language to describe legal texts has its roots in the need to improve the readability and comprehensibility of legal texts. In his paper [1], Allen suggests using a formal language as a remedy to inadvertent vagueness and ambiguity in legal texts.

[1] See http://softpert.com/legal/court-management/winjuris.
[2] See https://cst.cnpd.lu/portal for GDPR compliance checking.
[3] See https://github.com/normativeai.

ACM ISBN ???.
https://doi.org/???

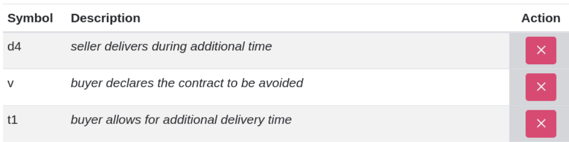

Figure 2: The definitions of sentences from a normalization of articles 47(1) and 49(1b) of the convention

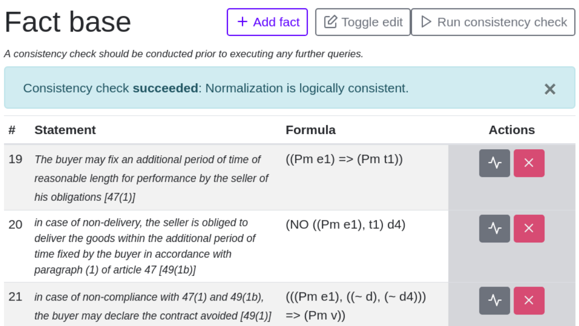

Figure 3: Symbolic encoding of a normalization of articles 47(1) and 49(1b) of the convention

By using propositional connectives, Allen shows how logic can be used for improving the "art" of legal drafting.

In the case study which follows, we have decided to use a formal legal language based on the one developed in [2]. Beside being relatively expressive, this language can be straightforwardly encoded into our chosen logic.

Using the application. As an example we have decided to use articles from the *The United Nations Convention on Contracts for the International Sale of Goods*. In addition to the ones described in [7], we have added also articles 47(1) and 49(1b).

Once the users have logged-in to the system, they have the possibility to create and update legal theories and queries. A legal theory corresponds to the normalization of a legislature in the normalization language which was mentioned in the previous section. Right now we support only a symbolic version of this language but future extensions of the tool will support a textual and more user-friendly version.

Figure 2 shows some of the definitions which were used in the normalization process. Figure 3 presents a part of the symbolic and formal encoding of the norms as presented in the articles. In the process of building the legislation, the user is encouraged to repeatedly press the **run consistency check** button. This action will initiate the back-end prover to check the current legislation for logical consistency.

Once some norms have been encoded, the user can immediately start to reason automatically over them. By creating a new query using the **Queries** menu item, the facts of a certain case as well as possible normative consequences can be inserted. Figure 4 displays one such case where the buyer asks if she is allowed to declare the contract avoided, given that the seller also failed to deliver during the extended period of time. By clicking the **Execute query** button, the user can ask the back-end prover if the consequence follows from the legislation and facts (an answer of 'Theorem') or not (an answer of 'Non-theorem').

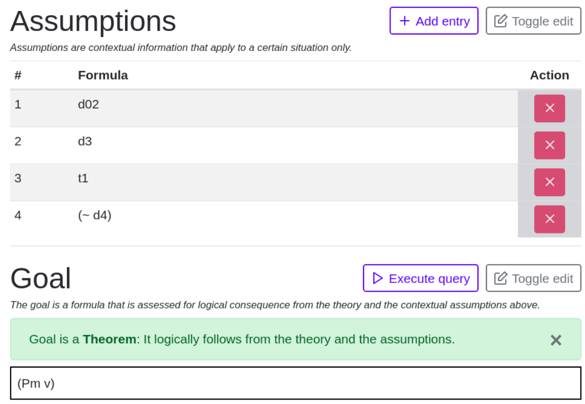

Figure 4: Can the buyer declare the contract to be avoided in case the seller failed to deliver within the extended period of time?

REFERENCES

[1] Layman E Allen. 1956. Symbolic logic: A razor-edged tool for drafting and interpreting legal documents. *Yale LJ* 66 (1956), 833.
[2] Layman E Allen and C Rudy Engholm. 1977. Normalized legal drafting and the query method. *J. Legal Educ.* 29 (1977), 380.
[3] Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, Piercarlo Rossi, and Leendert van der Torre. 2016. Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. *Artificial Intelligence and Law* 24, 3 (2016), 245–283.
[4] Anne-Marie Burley and Walter Mattli. 1993. Europe before the court: a political theory of legal integration. *International organization* 47, 1 (1993), 41–76.
[5] Henry DeYoung et al. 2010. Experiences in the logical specification of the HIPAA and GLBA privacy laws. In *Proc. of the 9th annual ACM workshop on Privacy in the electronic society*. ACM, 73–82.
[6] Guido Governatori and Sidney Shek. 2013. Regorous: a business process compliance checker. In *Proc. of the 14th Int. Conf. on Artificial Intelligence and Law*. ACM, 245–246.
[7] Tomer Libal and Matteo Pascucci. 2018. Automated Reasoning in Normative Detachment Structures with Ideal Conditions. *CoRR* abs/1810.09993 (2018). arXiv:1810.09993
[8] Pablo E Navarro and Jorge L Rodríguez. 2014. *Deontic logic and legal systems*. Cambridge University Press.
[9] Monica Palmirani and Guido Governatori. 2018. Modelling Legal Knowledge for GDPR Compliance Checking. In *Legal Knowledge and Information Systems: JURIX 2018: The Thirty-first Annual Conference*, Vol. 313. IOS Press, 101.
[10] John Zeleznikow and Andrew Stranieri. 1995. The split-up system: integrating neural networks and rule-based reasoning in the legal domain. In *Proc. of the 5th Int. Conf. on Artificial Intelligence and Law*. ACM, 185–194.