D3.2
Computational solutions for decision making and compliance

Grant Agreement nº: 690974
Project Acronym: MIREL
Project Title: Mining and REasoning with Legal texts
Website: http://www.mirelproject.eu/
Contractual delivery date: 31/12/2017
Actual delivery date: 31/12/2017
Contributing WP WP3
Dissemination level: Public
Deliverable leader: HUD
Contributors: HUD

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690974
Computational solutions for decision making and compliance
Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Partner</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>17/12/2017</td>
<td>Sotirios Batsakis</td>
<td>HUD</td>
<td>Draft</td>
</tr>
<tr>
<td>1.0</td>
<td>31/12/2017</td>
<td>Sotirios Batsakis</td>
<td>HUD</td>
<td>Final Version</td>
</tr>
</tbody>
</table>

Contributors

<table>
<thead>
<tr>
<th>Partner</th>
<th>Name</th>
<th>Role</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUD</td>
<td>Sotirios Batsakis</td>
<td>Editor</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>Wolfgang Faber</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>DLVSystem</td>
<td>Nicola Leone</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>DLVSystem</td>
<td>Mario Alviano</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>Data 61</td>
<td>Guido Governatori</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>Data 61</td>
<td>Brian Lam</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>UNS</td>
<td>Andrea Cohen</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>UNS</td>
<td>Sebastian Gottifredi</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>UNS</td>
<td>Vanina Martinez</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>UNS</td>
<td>Guillermo Smari</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>UNS</td>
<td>Gerardo Simari</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>George Baryannis</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>Ilias Tachmazidis</td>
<td>Author</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>Grigoris Antoniou</td>
<td>Reviewer/WP leader</td>
<td>Reviewer</td>
</tr>
</tbody>
</table>

Disclaimer: The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. MIREL consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.
Table of Contents

List of Figures .......................................................................................................................... 5
List of Acronyms ...................................................................................................................... 5
Executive Summary .................................................................................................................. 6
1  Introduction .......................................................................................................................... 7
2  Background and related work .............................................................................................. 7
3  Reasoning based on argumentation ..................................................................................... 9
4  Reasoning based on defeasible rules ................................................................................ 16
5  Reasoning based on Answer Set Programming ................................................................. 21
6  Conclusions and Research Challenges .............................................................................. 23
References ............................................................................................................................... 25
List of Figures

Figure 3-1: AF of Example 3-2

Figure 3-2: AF illustrating the agent’s state from Example 3-3

List of Acronyms

AF: argumentation framework
ASP: Answer Set Programming
DL: Description Logics
OWL: Web Ontology Language
SWRL: Semantic Web Rule Language
Executive Summary

The main objective of deliverable 3.2 is to provide a description of the logic based reasoning mechanism over normative knowledge. Such a mechanism will allow for efficient reasoning and decision making over legal knowledge by automatically applying legal norms.

Introduction, and background and related work on legal norm representation and reasoning are the first sections of this deliverable. Several approaches for normative reasoning are presented in the following. Specifically the argumentation based approach is presented followed by a reasoning approach based on defeasible reasoning. An alternative approach to normative reasoning based on answer set programming is presented as well. The above mentioned approaches are presented based on their applicability on the legal domain. Finally conclusions and research challenges identified are presented as a first step towards the development of efficient solutions on legal based decision making and compliance.
1 Introduction

Decision making when compliance with rules must be enforced is a very important application area. Deliverable 3.1 of this project is focused on the representation of legal norms, regulations and related concepts using ontologies, while the focus of this deliverable is the corresponding reasoning mechanism. Regulations are a widespread and important part of governments and businesses. They encode how products are manufactured, and how the processes are to be performed. Such regulations, in general, are difficult to understand and apply. Undoubtedly, the law, for example, as the reflection of human society, presents the broadest range of expression and interpretation, since the interpretation of even the most common words becomes problematic. Even individual regulations may be self-contradictory as a result of their gradual development process, as well as the lack of a formal (formal) drafting process. The problem becomes more and more difficult when independent regulations are applied in the same circumstances. For example, when two regulations overlap, it is not clear whether a regulation will override or will both apply. Even though regulations are typically drawn up, as is often the case in the legal field, problems such as consistency, interpretation and use remain. In an increasingly complicated environment, as well as regulatory review, an automated reasoning process becomes more and more necessary. This deliverable is the first step towards the development of such a system.

Introduction, background, and related work on legal norm representation and reasoning are the first sections of this deliverable. Several approaches for normative reasoning are presented in the following. Specifically the argumentation based approach is presented followed by a reasoning approach based on defeasible reasoning. An alternative approach to normative reasoning based on answer set programming is presented as well. The above mentioned approaches are presented based on their applicability on the legal domain. Finally conclusions and research challenges identified are presented as a first step towards the development of efficient solutions on legal based decision making and compliance.

2 Background and related work

Research on the confluence of AI and Law has been active for more than four decades. We refer the interested reader to detailed accounts of such research in [1], [2], [3]. In this section, the focus is only on the various approaches for normative knowledge representation and reasoning.

Early attempts at realizing normative reasoning involved representing legislation in the form of Horn logic programs, such as Sergot et al.’s seminal work on the British Nationality Act [4]. However, monotonicity and the treatment of negation in pure Prolog proved problematic. Extensions that support negation as failure and negated conditions solve some issues but raise others, such as the cases of double negation and counterfactual conditionals (e.g. “if it didn’t rain”). Also, introducing new exceptions to existing legislation would mean rewriting the whole logic program to take them into account. Hence, Prolog and variants prove useful only in
Computational solutions for decision making and compliance

representing self-contained and stable legislation [5], but even so, they can only model the questions that need to be answered in a legal debate, not how they are to be answered [2].

Following the advent of the Semantic Web and the introduction of the OWL family of languages, several research efforts focused on examining whether description logics are a suitable candidate for representing and reasoning about legislation. A prime example is HARNESS [6] (also known as OWL Judge [7] which shows that well established sound and decidable DL reasoners such as Pellet can be exploited for normative reasoning, if, however, a significant compromise in terms of expressiveness is made. The most important issue is that relationships can only be expressed between concepts and not between individuals, e.g. using the example in [6], if we have statements expressing the facts that a donor owns a copyright donation and that a donor retains some rights, there is no way to express (in pure OWL) that the donor in both cases is the same individual. This can be expressed via rules (e.g. written in SWRL); however, to retain decidability these rules must be restricted to a so-called DL-safe subset [8].

A common issue that arises when using classical or description logics in normative reasoning is the fact that they are monotonic: logical consequences cannot be retracted, once entailed. This is in contrast to the nature of law, where legal consequences have to adapt in light of new evidence and conflicts between different regulations must be accounted for and resolved. Therefore, it is natural to employ non-monotonic logic for the purposes of normative reasoning. The Defeasible Logic framework [9] has been used for normative reasoning purposes due to its simplicity and flexibility and the fact that several efficient implementations exist [10], [11]. In the DL framework, rules can either behave in the classical sense (strict), they can be defeated by contrary evidence (defeasible), or they can be used only to prevent conclusions (defeaters). Additionally, shows how the framework can be extended with concepts such as intentions and obligations, which are of paramount importance in normative reasoning.

The aforementioned approaches are more suited to legal systems that are primarily based on civil law, due to their rule-based nature and the fact they focus on conflicts arising from conflicting norms and not from interpretation [12]. On the other hand, common law places precedents in the center of normative reasoning, which makes case-based approaches more applicable. The most prominent examples of case-based normative reasoning are HYPO [13], CATO [14] and GREBE [15]. HYPO represents cases in the form of dimensions which determine the degree of commonality between two precedent cases: a precedent is more “on-point”, if it shares more dimensions with the case at hand than another. CATO replaces dimensions with boolean factors organised in a hierarchy. GREBE is actually a rule/case hybrid, since reasoning relies on any combination of rules modelling legislation and cases represented using semantic networks (a precursor to ontologies in the Semantic Web). As noted in [2], using dimensions or factors to determine legal consequences is relatively tractable, but the initial step of extracting these dimensions or factors from case facts is deeply problematic.

Regardless of the legal system applied, normative reasoning at its core is a process of argumentation, with opposing sides attempting to justify their own interpretation. As succinctly stated in [3], legal reasoning goes beyond the literal meaning of rules and involves appeals to
Computational solutions for decision making and compliance

precedent, principle, policy and purpose, as well as the construction of an attack on arguments. This became especially apparent when Dung [16]’s influential work on argumentation frameworks started being applied in AI and law research. A notable example is Carneades [17], a model and a system for constructing and evaluating arguments that is in active development and has been used in a legal context. Using Carneades, one can apply pre-specified argument schemes that rely on established proof standards such as “clear and convincing evidence” or “beyond reasonable doubt”.

ASPIC+ [18] takes a more generic approach, providing a means of producing argumentation frameworks tailored to different needs in terms of the structure of arguments, the nature of attacks and the use of preferences. However, neither Carneades or any framework produced by ASPIC+

to can be used as-is for normative reasoning: they need to be instantiated using a logic language. For instance, versions of Carneades have used Constraint Handling Rules to represent argumentation schemes, while any ASPIC+ framework can be instantiated using a language that can model strict and defeasible rules, such as those in the aforementioned Defeasible Logic framework.

It is worth mentioning that the recent proliferation of machine learning research has led to several data-centric approaches, differentiated in [19] based on whether they are oriented towards documents, cases or corpora. The latter two are more related to normative reasoning, using predictive analytics based on either past cases or entire collections of legal texts. A recent notable example is [20], where binary classifiers are applied on documents of cases tried at the European Court of Human Rights in order to predict judgement on future cases based on similarity.

3 Reasoning based on argumentation

In pursuit of her goals, an agent must be able to reason about the actions she may perform, but must also consider the regulative norms imposed on her. Such norms define obligations and prohibitions (from here on referred to as normative provisions) on her behaviour. Then, since the agent should comply with her normative provisions, she will deem the compliance of her obligations and prohibitions as an additional goal. Given that normative provisions may be conflicting, an agent might be forced to violate some of them in order to comply with others. Furthermore, it could be the case that an agent is not able to perform the course of action leading to a normative provision’s compliance, due to some external constraints, resulting in a state where a violation is produced. In this context, the agent needs a strategy for selecting the actions leading her to achieve her goals, while maximising her normative provisions’ compliance. To illustrate this, let us consider the following scenario, which will serve as a running example throughout the rest of this section.
**Example 3-1:** Let us consider an agent that is currently at work and has the obligation to appear in court to testify in a trial. Right now, she is alone at the office and has to be in court before the office’s closing time. Her main problem is that she is forbidden from leaving the office unattended for the rest of the day. As an additional complication, her car is broken (today she got a ride to the office). In order to comply with the obligation to appear in court, the agent has to sort out transportation while trying not to violate the prohibition to leave the office unattended.

Argumentation is an attractive and effective paradigm for conceptualising common-sense reasoning [21][22] [23]. The argumentation process has been employed in various applications and domains such as decision making and negotiation [24][25], multi-agent systems [26][27], and normative reasoning [28][29], among others.

One of the strengths of argumentation is its ability to handle conflicts due to inconsistent information. Inconsistency naturally arises in agent-based systems since, among other reasons, an agent may have conflicting goals and beliefs [30][31][32]. Then, an agent’s beliefs and goals may be represented as arguments, and the conflicts between them can be captured through attacks between the corresponding arguments.

In this example use case, an agent’s knowledge will be represented by means of an abstract argumentation framework (AF) [16]. In particular, the agent’s normative provisions will be represented as arguments. Then, given a particular state of the world (represented as an AF), the agent can check what obligations and prohibitions she is complying with or violating by looking at the accepted arguments of the corresponding AF. In addition, the agent has a set of actions she may perform, which will allow her to modify the state of the world (hence, adding/removing arguments and their associated attacks in the AF). To maximise compliance of an agent’s normative provisions in this setting, we will follow a weighted MaxSAT approach [33].

For this, we will propose an encoding of the agent’s AF, the available actions, and the normative provisions as Boolean formulas. Also, we will provide an encoding for selecting the accepted arguments of an AF that accounts for the effect of the agent’s actions.

In the literature, there exist some approaches that make use of argumentation for reasoning about an agent’s normative provisions’ compliance. For instance, in [34], the authors propose an argumentation theory based heuristics allowing an agent to decide, given a set of conflicting norms, which ones should be complied with. As another example, [29] proposes to evaluate the level of compliance associated with different plans an agent may adopt, and provide justifications for the best plan using an argumentation-based dialogue game. In contrast, in this example we focus on selecting the course of action that will lead to the agent’s maximal level of compliance, considering that the agent’s knowledge representation and reasoning tasks are performed following an argumentative approach.

Furthermore, our approach will select the course of action that, in addition to maximise the normative provisions’ compliance, minimises the number of actions to be undertaken by the agent. As mentioned before, in this use case, agents will act towards complying with their
normative provisions. Then, as their knowledge is represented through AFs, their actions will produce changes in those AFs. Recently, the community of argumentation has developed interest in studying the computation of extensions (sets of collectively acceptable arguments under some criteria) in dynamic argumentation frameworks; that is, argumentation frameworks that may dynamically change. On the one hand, there exist works aimed at efficiently computing extensions after changes in an AF have been produced. Such is the case of [35] where the authors, based on the work in [36], propose an incremental approach for re-computing an extension of an AF. Briefly, given an AF, an extension of that AF under a given semantics, and a set of modifications of the AF, they determine an extension of the updated AF under the chosen semantics.

For that purpose, they identify a reduced version of the AF, the “affected” part, from which they start the incremental re-computation process. Even though our approach in this example use case is similar to [35] in that we both deal with dynamics in AFs, there are differences between them. On the one hand, our approach does not have a set of updates (i.e., set of actions to be performed over the AF) provided beforehand; rather, our proposal has the aim of identifying which set of actions the agent should perform, among the available ones. Also, the selection of actions will be such that compliance of the agent’s normative provisions is maximised.

Furthermore, the selected course of action (following [35] terminology) will be the minimal one (in terms of number of actions) leading to the compliance maximisation. As a result, the approach in the example use case could be considered complementary to the one proposed in [35].

Another line of work addressing dynamics in argumentation frameworks concerns the laying of theoretical foundations for adding/removing arguments and/or attacks from AFs. As an example, the work of [37] provides the grounds for updating an AF with the addition/removal of attacks, as well as changing the acceptance status of arguments. For this, they represent an AF (arguments and attacks) using propositional logical formulas, together with constraints encoding a semantics, and describe how to update the AF in terms of operations over these formulas by means of propositional dynamic logic.

Then, in [38] they extend their previous proposal in order to account for the addition/ removal of arguments as well. Similarly to the approach applied in this section, their approach aims at minimising the changes needed to enforce the acceptance of one or more arguments. The main difference between these approaches and ours is that, given a set of possible changes (available actions), an agent in our approach will have to select actions to be applied in order to reach a desired state (where total compliance of her normative provisions is achieved). Otherwise, given a scenario where such a desired state is not reachable, the agent would explicitly account for that situation and select the best alternative solution (that is, the course of action leading to a state where compliance is maximised).

In this example, knowledge will be represented through abstract argumentation frameworks (AFs) [16]. Briefly, an AF is defined in terms of a set of arguments and a set of attacks among them.
**Definition 3-1:** An Abstract Argumentation Framework (AF) is a pair \(<A, R>\) where \(A\) is a finite and non-empty set of arguments and \(R \subseteq A \times A\) is an attack relation. Arguments in an AF are abstract entities (their origin is not specified), and will be denoted using bold lower-case letters. The attack relation between two arguments \(a\) and \(b\) denotes the fact that these arguments cannot be simultaneously accepted, since they are conflicting. An argument \(a\) attacks an argument \(b\) iff \((a, b) \in R\), denoted \(a \rightarrow b\). The set of attackers of an argument \(a\) will be noted as \(a^- \equiv \{b \mid b \rightarrow a\}\). To illustrate this, let us consider the following example.

**Example 3-2:** Let us consider the situation described in Example 3-1. In addition, suppose the agent can leave the office, ask her partner for a ride to the courthouse, ask her partner to cover for her in the office, and/or take a train to the courthouse. We can represent the whole knowledge with the following arguments:

- **a:** "Go to court to testify in a trial"
- **b:** "Cannot go to court since the car is broken"
- **c:** "Take the train to the courthouse"
- **d:** "Working at the office"
- **e:** "Leave the office unattended"
- **g:** "Ask partner for a ride to the courthouse"
- **h:** "Ask partner to cover at the office"

In particular, note that arguments \(g\) and \(h\) correspond to petitions the agent may perform to her partner. That is, she can ask her partner to give her a ride to the courthouse, as well as to cover for her at the office. However, it should be noted that these two petitions cannot be simultaneously fulfilled by her partner, as expressed by the attacks between their associated arguments. Similarly, the attacks between arguments \(c\) and \(g\) represent the fact that even though the agent may ask her partner to give her a ride to the courthouse, she could not simultaneously take that ride as well as the train. Given the above listed arguments and the conflicts between them, we can characterise an AF \(\Omega \equiv <\{a, b, c, d, e, g, h\}, \{(b \rightarrow a, c \rightarrow b, d \rightarrow a, d \rightarrow e, h \rightarrow e, h \rightarrow g, g \rightarrow h, g \rightarrow b, g \rightarrow c, c \rightarrow g\}\}>\). Figure 3-1 illustrates \(\Omega\) as a directed graph, following the usual convention in the argumentation literature, where arguments of the AF are represented as nodes in the graph, and edges denote the attack relation between them.

![Figure 3-1: AF of Example 3-2](image-url)
In abstract argumentation, the formal definition of methods ruling the argument evaluation process corresponds to the characterisation of argumentation semantics. A semantics definition specifies how to obtain a set of extensions, where an extension is a set of arguments that can survive together or are considered to be collectively acceptable [40].

Intuitively, an argument $a$ is acceptable w.r.t. a set $S$ if for every argument $b$ that attacks $a$, there is an argument $c$ in $S$ that attacks $b$ (in which case $c$ is said to defend $a$). An admissible set $S$ can then be interpreted as a coherent defendable position. For instance, given the AF $\Omega$ of Example 3-2, the sets $\{d\}$ and $\{h, c\}$ are admissible whereas the set $\{e, a\}$ is not. Then, starting from the notion of admissibility, [16] defines the acceptability semantics of the framework.

Looking into the admissible set $\{h, c\}$ of $\Omega$, it can be noted that adding argument $d$ to the set will result in another admissible set. We can keep adding arguments defended by the resulting set, until every argument that can be defended by the set is in that set. Such a set will hold arguments that can stand on their own, and denotes a complete extension. In addition, if look for maximal complete extensions, we obtain preferred extensions. A plethora of semantics for abstract argumentation frameworks has been proposed in the literature [39][40]; nevertheless, the focus of this work is not in contrasting the characteristics of every conceivable semantics, but on using argumentation to model and reason with an agent’s normative provisions. As a result, in this work we will only consider the complete and preferred semantics, which are comprehensive enough for us to study how the addition and removal of arguments affects the normative provisions’ compliance. In the current example approach, an agent will be characterised by a specification and a state. The agent’s specification will hold the static information: an AF containing every conceivable argument and the attacks between them, the actions that the agent may perform, and the normative provisions the agent has to comply with complete extensions, we obtain preferred extensions.

The first component the agent’s specification corresponds to the potential knowledge; that is, all the information the agent is aware of. We refer to this knowledge as potential since it may be the case that the agent is aware of a piece of information (represented by a given argument) that is not currently in place (and thus is not available in the current state of the world). Then, the agent’s obligations, prohibitions, and actions (as the agent is able to add or remove knowledge), are represented in terms to such potential knowledge. However, in a particular moment of her lifespan, the agent’s knowledge will correspond to a subset of the potential knowledge. This partial knowledge is associated to the notion of state. In other words, a state holds the information that is available for the agent at a particular moment of her lifespan (namely, available arguments and the attacks between them). Then, given an agent’s specification and a state, an argument from the agent’s potential knowledge is characterised as available if it belongs to the set of available arguments ($A$) of that state; otherwise, it is characterised as unavailable.

Example 3-3: Consider the agent corresponding to the situation described in Examples 3-1 and 3-2. From there we can note that the agent is obliged to be in court to testify and is prohibited from leaving the office unattended. Therefore, the agent’s set of obligations will be $\{a\}$, whereas her set of prohibitions will be $\{e\}$. As illustrated in Example 3-2, the agent can leave the office (which
results in removing argument d), ask her partner for a ride to the courthouse (add argument g), ask her partner to cover for her at the office (add argument h) or take the train to the courthouse (add argument c). As a result, the agent’s specification (in a form \((\text{AF, added arguments, removed arguments, obligations, prohibitions})\)) will be \(S = (\Omega, \{c, g, h\}, \{d\}, \{a\}, \{e\})\). Then, if we consider that the agent is currently in a situation like the one described in Example 3-1, her associated state in \(S\) will be \(\Sigma = (\{a, b, d, e\}; \{b \rightarrow a, d \rightarrow a, d \rightarrow e\})\). The agent’s state from Example 3-3 is depicted in Figure 3-2. In particular, it can be noted that the available knowledge in \(\Sigma\) is a subset of the universal knowledge (with the available arguments and attacks being highlighted in the figure).

Clearly, an agent’s beliefs might change from state to state, depending on the corresponding available knowledge. Furthermore, the beliefs an agent has in a particular state will be determined by the accepted arguments that result from an argumentative analysis carried out over the available knowledge. As a result, the notion of state is central. This is because normative provisions are represented as arguments which will be tried to be made “in” or “out”, respectively. Then, the agent’s compliance (with respect to her obligations and prohibitions, as defined by her agent’s specification) will depend on the state she is in. Intuitively, an agent will comply with an obligation \(O\) if she is in a state where there exists a preferred extension deeming \(O\) as accepted. In contrast, compliance of a prohibition \(P\) will be achieved in a state where argument \(P\) is considered to be rejected.

![Figure 3-2: AF illustrating the agent’s state from Example 3-3](image)

Example 3-4. Let us consider the agent’s specification \(S\) and the state \(\Sigma\) presented in Example 3-3, which correspond to the situation described in Examples 3-1 and 3-2. In state \(\Sigma\), agent \(Ag\) has the preferred extension \(\{d, b\}\), under which she complies with the prohibition expressed by \(e\) and violates the obligation imposed by \(a\).

An agent’s specification establishes, among other things, the actions (sets Add and Rem) the agent is able to perform throughout her lifespan. These actions will allow the agent to modify the state she is in, thus changing the set of available arguments. In our approach, we are abstracting from the internal structure of actions; thus, they are simply specified in terms of the arguments they affect. Hence, the presence of an argument in Add (Rem) represents that the agent is able to perform an action that will render such argument as available (unavailable). Then, given an agent that is in a particular state, the application of some actions will result in her being in a new state.

Ideally, an agent being in a particular state will choose to apply a set of actions that will allow her to reach a desired state. Then, starting from a particular state, she has to strategically decide a course of action with the aim of achieving the desired state.
Example 3-5: Let us consider the agent’s specification $S$ and the state $\Sigma$ presented in Example 3-3, which correspond to the situation described in Examples 3-1 and 3-2. Also, let us suppose agent $Ag$ chooses a strategy in $\Sigma$ which consists in leaving the office, and taking a ride from her partner to the courthouse (after asking for it). Then, the corresponding strategy would be $\text{Strat}(S, \Sigma) = \{g\}$, $\{d\}$. As a result, after adding $\{g\}$ and removing $\{d\}$ in state $\Sigma$, the agent will reach a new state $\Sigma_2 = \langle\{a, b, e, g\}; \{b \rightarrow a, g \rightarrow b\}\rangle$. Note that, in state $\Sigma_2$, the agent will have only one preferred extension $\{a, e, g\}$, under which she complies with the obligation $a$ but violates the prohibition $e$.

As it can be observed in the previous example, the agent’s adopted strategy is such that it leads to a state where normative provisions are not fully complied with. In contrast, if the agent had chosen a strategy where she left the office, asked her partner to cover for her and took the train to the courthouse, compliance would have been maximised.

Next, given an agent whose normative provisions are determined by her specification, and a particular state she is in, we will address the issue of obtaining a strategy such that the application of its actions leads to a state where compliance of the agent’s obligations and prohibitions is maximised.

For the generation of agent’s strategies, we propose a two-step approach that reduces the search for strategies to a sequence of weighted MaxSAT problems. The algorithm is based on the idea of encoding the specification of the agent $S$ and the current state into a sequence of MaxSAT problems, in order to identify the maximum set of prohibitions and obligations that can be complied with at the same time, and to minimise the actions to execute. A weighted MaxSAT formula is a Boolean formula composed by hard constraints, that must be satisfied in every possible solution, and soft constraints. Each soft constraint has an associated weight. The MaxSAT problem consists of finding an assignment that maximises the sum of the weights of the satisfied clauses.

We are now in the position to illustrate the proposed procedure, listed in Algorithm 1, to generate strategies for the agent. Algorithm 1 requires as input the set of obligations of the agent $O$, the prohibitions $F$, the argumentation framework $\Omega$ — which encodes the potential knowledge available to the agent—, the list of available actions $A$, and an initial labelling $L_i$ — which distinguishes what arguments are initially in, out, nondec or unavailable given an initial state of the agent and a preferred extension of the AF encoding that state. The provided output is a labelling $L_m$ which maximises compliance with regards to the agents’ prohibitions and obligations, and at the same time minimises the number of actions to be undertaken.

Initially, in lines 1–3, it is checked whether it is possible for the agent to comply with all the obligations and prohibitions. This can be checked by generating an appropriate MaxSAT formula, where obligations and prohibitions to be complied with are encoded using hard constraints. If the formula is satisfiable, then the MaxSAT encoding proposed in the previous section will generate the appropriate strategy $\text{Strat}(S, \Sigma)$ minimising the number of actions needed to comply with every obligation and prohibition. In cases where total compliance is not achievable, the proposed procedure identifies the maximum set (in terms of size) of obligations and prohibitions that the
agent can comply with. This is done in lines 5–7, and corresponds to generating a MaxSAT formula where clauses 11–16 are omitted, and clauses 7 and 8 are encoded as soft clauses. In this way, action minimisation is ignored, and the focus is given to the maximisation of compliance.

Once the largest compliance set has been identified, this is set as the goal of the agent, and a new MaxSAT formula is generated to create the strategy $\text{Strat}(S, \Sigma)$. In this formula, only the prohibitions and obligations identified in lines 5–7 are encoded. It should be noted that it may occur that the agent cannot comply with any of the obligations and prohibitions. This case is handled in lines 8–9, and an unsatisfiable value is returned by the procedure. Remarkably, the formula $\text{MaxCompliance}$ can be improved in order to prefer some specific obligations or prohibitions, instead of maximising the set of those that the agent can comply with. This can be done by modifying the weight that the corresponding soft clauses have in the MaxSAT formula. The higher the weight, the higher the importance of the element it encodes.

### 4 Reasoning based on defeasible rules

Defeasible rules [9] can be used for normative reasoning as argumentation of the previous section and this approach will be presented in the following. A defeasible theory $D$ is a triple $(F, R, >)$ where $F$ is a finite set of facts (literals), $R$ a finite set of rules, and $>$ a superiority relation (acyclic relation upon $R$). A rule $r$ consists (a) of its antecedent (or body) $A(r)$ which is a finite set of literals, (b) an arrow, and, (c) its consequent (or head) $C(r)$ which is a literal. There are three types of rules: strict rules, defeasible rules and defeaters represented by a respective arrow $\to$, $\Rightarrow$ and $\neg$. Strict
rules are rules in the classical sense: whenever the premises are indisputable (e.g., facts) then so is the conclusion.

Defeasible rules are rules that can be defeated by contrary evidence. Defeaters are rules that cannot be used to draw any conclusions; their only use is to prevent some conclusions. Different types of rules supported by defeasible logic make it suitable for legal reasoning as will be presented in the following. Since legal rules are often conflicting rules, defeasible rules can be applied for such an application domain.

Legal doctrine and judicial practice distinguish among a number of canons for interpreting legal statutes, i.e., different rules that are employed in legal systems as patterns for constructing arguments aimed at justifying certain interpretations, while attacking other interpretations. [41], summarising the outcomes of a vast study on statutory interpretation, involving scholars from many different legal systems, distinguishes eleven types of arguments. A different list of interpretive arguments was developed by [42] and identifies fourteen types of arguments.

This sections presents a logical machinery for modelling reasoning about interpretive canons and thus for justifying the choice of a certain canon and the resulting legal outcome over competing interpretations (see [43, 44]). [45] argued that an interpretive canon for statutory law can be expressed as follows: if provision n occurs in document D, n has a setting of S, and n would fit this setting of S by having interpretation a, then, n ought to be interpreted as a. For instance, the ordinary language canon has the following structure: if provision n, stating that “Killing a man is punishable by no less than 21 years in prison”, occurs in document D = Penal code, n has a setting of ordinary language, and n would fit this setting of ordinary language by having interpretation a = “Killing an adult male person is punishable by no less than 21 years in prison”, then n ought to be interpreted as a.

In this section we basically accept this research line and work according to the following intuitions.

Intuition 4-1 (Reasoning and canons). We analyse the logical structure of interpretive arguments (in the sense of [41]) using a rule-based logical system. In particular, interpretation canons are represented by defeasible rules, where

- antecedent conditions of interpretation rules can be of any type (assertions, obligations, etc.), including the fact that another canon is refuted or that another legal provision ought to be interpreted in a certain way;
- the conclusion of interpretation rules is an interpretive act leading to an interpretation of a certain provision n and thus to a sentence which expresses the result of such an interpretation and paraphrases n [46]. If n and n’ are legal provisions, the following is an example of interpretation rule regarding n’:

\[
\text{IF} \\
\text{n ought to be interpreted literally as a; AND} \\
\text{n is related with n’ AND}
\]
Computational solutions for decision making and compliance

a entails a’;

THEN

n’ is interpreted by coherence as a’:

We will use these rules to devise a reasoning machinery that mirrors legal reasoning about interpretive canons. The resulting rule-based system is in line with the basic ideas inspiring the argumentation system by [44].

Notice that the above intuition distinguishes the interpretive act from the result of the interpretation:

Intuition 4-2 (A- and O-interpretation). We assume the distinction between interpretation as activity and as outcome [47, p. 117] (cf. [42, p. 39]):

- interpretation as activity (A-interpretation) (literal or from ordinary language, by coherence, etc.) views any argumentative canon as a means through which a certain meaning is ascribed to a legal provision, and

- interpretation as outcome (O-interpretation) is precisely the meaning obtained through a certain interpretive act and ascribed to the provision.

The distinction between interpretation as activity and as outcome is well known in continental legal theory, and it was introduced precisely to capture cases where, e.g., one has legal reasons to prefer a certain interpretive canon over others even though all considered canons support the same interpretive outcome. In other words, an interpretive act I of n as a (A-interpretation of n) is a way to bring about that a (O-interpretation of n) is the case. For example, in Intuition 4-1, the A-interpretation of n’ is the act interpreting n’ by coherence, while the resulting O-interpretation from that act is a’, i.e., a sentence expressing the meaning attributed to n’ through the interpretation by coherence.

Since different competing canons can be employed, different conflicting rules can be accordingly applied for interpreting statutes. Interpretation rules are thus defeasible. As argued in [45], some priority criteria should be applied to interpretation rules [48]. Such criteria impose preference relations over conflicting interpretive acts and outcomes. In other words, to address interpretive conflicts, we need to assume that one of the conflicting arguments is stronger than its competitors. Some legal traditions provide indeed general criteria for addressing conflicts of arguments on the basis of their priorities: for instance, several continental legal systems explicitly state that literal interpretation ought to be preferred, or that an argument concerning constitutional values ought to prevail over a historical argument (e.g., an argument based on the intent of the historical legislator).

However, ranking among interpretive acts and canons can be applied also when such acts are not in conflict. Suppose, for example, that provision n can be interpreted as a by adopting an
argument by analogy and one from substantive reasons⁴ if n is a provision of criminal law (but analogy is admissible whenever it favours the defendant), then the argument from substantive reasons ought to be preferred, even though both lead to read n as a.

Intuition 4-3 (Preferences over interpretations):

A standard priority relation [49] over interpretation rules can be introduced to handle and solve conflicts between different interpretation rules. Consider the following example:

Rule1
IF
n ought to be interpreted literally as a AND
n is related with n’ AND
a entails a’
THEN
n’ is interpreted by coherence as a’
Rule2
IF
n” ought to be interpreted literally as ¬a AND
n is related with n” AND
¬a entails ¬a’
THEN
n’ is interpreted by coherence as ¬a’

Here, we can handle the conflict by stating that Rule1 >Rule2 (or vice versa).

Ranking among interpretive acts can be applied also when such acts are not in conflict. We will thus introduce an operator that can be used to make explicit in single rules this idea. For instance,

IF
n ought to be interpreted literally as a; AND
n is related with n’

---

¹ An argument from substantive reasons states that, if there is some goal that can be considered to be fundamentally important to the legal system, and if the goal can be promoted by one rather than another interpretation of the statutory provision, then the provision should be interpreted in accord with the goal.
THEN

\( n' \) is interpreted by coherence as \( a' \otimes \)

\( \otimes n' \) is interpreted by analogy as \( a' \)

means that the most preferred interpretation resulting in \( a \) is the one by coherence, but, if this is refuted, the second option is the interpretation by analogy. This does not require to only derive one interpretation resulting in \( a \) (other rules could first support interpretation by analogy of \( n \)).

Following some doctrinal and judicial practice, [45] argued that interpretive canons are defeasible rules licensing deontic interpretive claims, namely, the claim that a certain expression in a statute ought, ought not, may or may not be interpreted in a certain way. For example, art. 12 in the general provisions of the Italian civil code states that the literal interpretation of statues ought to be preferred and this option is nothing but an interpretive prescription. Here, we follow this intuition with some adjustments.

Intuition 4-4 (Obligatory interpretations). An interpretation can be admissible or obligatory. In the case of \( A \)-interpretations, for instance, an interpretive act \( I \) of \( n \) (\( A \)-interpretation of \( n \)) is admissible, if it is provable using a defeasible interpretation rule; it is obligatory, if this interpretation of \( n \) is the only one admissible. Similarly for \( O \)-interpretations. Indeed, consider the general provisions of the Italian civil code, which state at art. 12 that literal interpretation \( I_{lit} \) ought to be preferred: this would support that such interpretation is obligatory, unless another interpretation prevails. We have two options here:

- other conflicting interpretations can be derived, thus requiring to check if literal interpretation overrides the other options; if it does not, then the interpretation at stake is not even admissible;
- other non-conflicting interpretations can be provable; if they are, the interpretation at stake is only admissible, otherwise, it is obligatory.

On the basis of the above intuitions, we will offer two options for modelling reasoning about interpretations: a defeasible logic for reasoning about the interpretation of abstract, non-analysed provisions and of structured provisions.

Intuition 4-5 (Abstract or structured provisions).

A provision \( n \) is abstract if it is taken in its sentential entirety for interpretive purposes, i.e., as a non-analysed sentence without considering its internal (logical) structure. In this chapter, Option 1 amounts to interpreting \( n \) by ascribing to \( n \), intended as an abstract provision, a sentential meaning that can be expressed by another sentence paraphrasing this provision as a whole.

Rather, a provision \( n' \) is logically structured if it corresponds to a linguistic sentence having the structure of a rule \( a_1, ..., a_n \Rightarrow b \) this means that \( n' \) is semi-interpreted provision, since expressing the logical structure of \( n' \) requires an interpretive effort on the original textual version of \( n' \). In this chapter paper, Option 2 amounts to interpreting \( n' \) by considering the components \( a_1, ..., a_n, b \) of \( n' \).
Computational solutions for decision making and compliance

and ascribing to them a meaning as already explained above. The above intuitions are implemented adjusting the framework in [50], which is a Modal Defeasible Logic [51] extended with the operator $\otimes$. The logic is a significant extension of standard Defeasible Logic [49], which however preserves linear computational complexity (like standard Defeasible Logic).

A Modal Defeasible Logic for Option 1 (see Intuition 5 above), i.e., a machinery for reasoning about the interpretation of abstract provisions and a similar Modal Defeasible Logic for reasoning about the interpretation of structured provisions are presented in [52].

[52] presented a logical machinery for reasoning about interpretive canons, which is based on the following intuitions: (a) canons are represented by defeasible rules; (b) different reasoning patterns can be identified depending on whether we work on interpretations as activities or as outcomes [47]; (c) competing interpretive options can be handled by stating a priority over conflicting rules, but different ranking preferences can also be introduced among compatible interpretive acts; (d) canons are defeasible rules licensing deontic interpretive claims; (e) the logic can deal with the interpretation of abstract, non-analysed provisions and of structured provisions.

A valuable aspect of the proposed machinery in [52] based on defeasible rules is that it can accommodate different doctrinal views regarding legal interpretation. In particular, it is argued that two different but non-conflicting interpretations of the same provision can be admissible.

5 Reasoning based on Answer Set Programming

In this section a reasoning example based on Answer Set programming (ASP) will be presented. Although ASP has not be applied in the legal domain, as argumentation and defeasible rules, it’s expressivity is adequate for such an application domain. This will be demonstrated using an example use case. The example is a US bar exam question [53]. Specifically the description of the case and the corresponding questions are:

“An entrepreneur from State A decided to sell hot sauce to the public, labelling it “Best Hot Sauce.” A company incorporated in State B and headquartered in State C sued the entrepreneur in federal court in State C. The complaint sought $50,000 in damages and alleged that the entrepreneur’s use of the name “Best Hot Sauce” infringed the company’s federal trademark. The entrepreneur filed an answer denying the allegations, and the parties began discovery. Six months later, the entrepreneur moved to dismiss for lack of subject-matter jurisdiction.

Should the court grant the entrepreneur’s motion? No, because the company’s claim arises under federal law. The claim asserts federal trademark infringement, and therefore it arises under federal law. Subject-matter jurisdiction is proper under 28 U.S.C. § 1331 as a general federal-question action. That statute requires no minimum amount in controversy, so the amount the company seeks is irrelevant.
Should the court grant the entrepreneur’s motion? No, because the entrepreneur waived the right to challenge subject-matter jurisdiction by not raising the issue initially by motion or in the answer. Under Federal Rule 12(h)(3), subject-matter jurisdiction cannot be waived and the court can determine at any time that it lacks subject-matter jurisdiction. Therefore, the fact that the entrepreneur delayed six months before raising the lack of subject-matter jurisdiction is immaterial and the court will not deny his motion on that basis.

Should the court grant the entrepreneur’s motion? Yes, because although the claim arises under federal law, the amount in controversy is not satisfied. There is no amount-in-controversy requirement for actions that arise under federal law. Although diversity jurisdiction requires an amount in controversy of $75,000 or more, when diverse parties are litigating a federal claim, the action is treated for jurisdictional purposes as a federal-question action, not a diversity action. The claim here asserts federal trademark infringement and therefore it arises under federal law. The fact that the action does not meet all the requirements for diversity jurisdiction is irrelevant.

Should the court grant the entrepreneur’s motion? Yes, because although there is diversity, the amount in controversy is not satisfied.”

In D3.1 a similar example for this use case was presented based on Answer Set Programming (ASP) with inheritance networks [54]. Norms were represented as rules and their priority as inheritance networks. Since support for preferences is now part of main Answer Set Programming tools [55], the abovementioned solution based on inheritance networks is not the only one available when using ASP for legal reasoning. An example ASP program compatible with current ASP standards for the above mentioned use case is the following one:

```prolog
copyrightInfringement(A,B,X):-sells(A,X),holdCopyright(B,X),A<>B.
isAt(A,S):- from(A,S).
isAt(A,S):- incorporated(A,S).
isAt(A,S):- headquartered(A,S).
diversityCase(A,B):-copyrightInfringement(A,B,X),accuses(B,A),diversityAmount(B,A,M).
federalCase(A,B):-copyrightInfringement(A,B,X),accuses(B,A),isAt(A,S1),isAt(B,S2),S1<>S2.

% prefer o true at level 1
{o}.
:- not o. [1@1,o]
acceptsClaim(C,A) :- accuses(B,A),court(B,A,C),challenges(A,C),not challengesEarly(B,C), o. % rule in o

% prefer o1 true at level 1
{o1}.
:- not o1. [1@1,o1]
acceptsClaim(C,A) :- accuses(B,A),court(B,A,C),challenges(A,C),not diversityCase(A,B), o1. % rule in o1

% prefer o2 true at level 2 (hence, prefer o2 to o and o1
{o2}.
```
By executing the above program the fact that copyright infringement occurs is inferred, along with the fact that it is a federal case and that the court does not accept the defendant’s claim. The above example was executed in less than 1 sec using CLINGO ASP reasoner\(^2\). Overall the logic programming approach offers all required expressiveness, and is an approach that can be used for normative reasoning.

6 Conclusions and Research Challenges

In this deliverable the representation of legal norms in conjunction with a reasoning mechanism is examined. The available options and work related the proposed reasoning mechanisms are presented. Specific examples are also presenting, demonstrating the available options.

There are several options for the normative reasoning mechanism including argumentation, defeasible rules and Answer Set Programming. The reasoning mechanism will be applied on rules and concepts representations presented in Deliverable 3.1 of MIREL project. Reasoning can be achieved using argumentation and an important research topic is if this approach is applicable only to small scale or it can be scalable to large scale reasoning. The expressiveness of the argumentation based approach is adequate for legal reasoning as demonstrated by a use case and such an approach does not has the limitations of approaches based on OWL Axioms or i.e., Horn formulas that can be naturally represented using rules (e.g. using SWRL). In the general case that norms are conflicting and/or contain negation or disjunction then expressive non-monotonic (for dealing with conflicts) logic programming formalisms must be used. This is the most generic approach for realistic use cases.

In addition to argumentation, both defeasible rules and Answer set programming are reasoning mechanism supporting the required expressiveness and they can be also used in conjunction with the representation of legal norms. Parallelizing these approaches and applying them in large scale applications is an open research challenge, and the main task of WP3 or MIREL project.

\(^2\) https://sourceforge.net/projects/potassco/files/clingo/
This task is orthogonal to the task of creating the reasoning rules from the natural language description, which is also a challenging task requiring legal and technical expertise. In a realistic large scale use case, the number of rules and facts can be big, thus causing scalability problems. Dealing with such problems and achieving efficient large scale semantic reasoning in the main research challenge to deal with.
References


Computational solutions for decision making and compliance


[34] Oren, N.; Luck, M.; Miles, S.; and Norman, T. An argumentation-inspired heuristic for resolving normative conflict. In 5th Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems (COIN@AAMAS), Estoril, Portugal. 2008.


