Università degli Studi di Verona

DIPARTIMENTO DI INFORMATICA
Corso di Dottorato di Ricerca in Informatica (XXXIV ciclo)

Tesi di Dottorato di Ricerca
S.S.D. INF/01

A Computational Framework for Formalizing Rules and Managing Changes in Normative Systems

Dottorando:
Luca Pasetto

Tutor:
Prof. Matteo Cristani

Coordinatore:
Prof. Massimo Merro

Anno Accademico 2020–2021

A Computational Framework for Formalizing Rules and
Managing Changes in Normative Systems
Luca Pasetto
Tesi di Dottorato
Verona, 2022

Abstract

Legal texts are typically written in a natural language. However, a legal text that is written in a formal language has the advantage of being subject to automation, at least partially. Such a translation is not easy, and the matter is even more complex because the law changes with time, so if we formalized a legal text that was originally written in natural language, there is a need to keep track of the change. This thesis proposes original developments on these subjects. In order to formalize a legal document, we provide a pipeline for the translation of a legal text from natural to formal language and we apply it to the case of natural resources contracts. In general, adjectives play an important role in a text and they allow to characterize it: for this reason we developed a logical system aimed at reasoning with gradable adjectives. Regarding norm change, we provide an ontology to represent change in a normative system, some basic mechanisms by which an agent may acquire new norms, and a study on the problem of revising a defeasible theory by only changing its facts. Another contribution of this thesis is a general framework for revision that includes the previous points as specific cases.

Contents

Chapter 1

Introduction

Most organizations impose a set of rules that have to be followed in order to be part of the organization group or to engage with it. This is necessary so that the organization in question can achieve its specific goals while avoiding unwanted consequences. The vast majority of the rules of an organization are still usually written in natural language. This holds true even if programs that execute actions according to the terms of a contract or an agreement, called smart contracts, are becoming more common nowadays with the recent diffusion of blockchain technology. These programs present specificities that are not the focus of this thesis.

For what regards rules that are written in natural language, the person who writes the rules can make the effort to use expressions that are as precise as possible, but natural language has some intrinsic ambiguities, and oftentimes needs to be interpreted. On the other hand, a formal language has the disadvantage of not being easily understandable by humans: the personnel might need to be trained in order to read and write the formal language. Indeed, even law professionals that are already proficient with legal technicalities might need to be instructed on how to use the specific details of a formal language. However, this comes with the advantage of being able to write rules that are precise, not ambiguous, and, most importantly, automatable.

We separate organizations into two kinds: public institutions and private businesses. In both cases, rules are created and enforced by the organization. Indeed, institutions themselves have been defined as sets of rules [27]. Both institutions and private businesses design business processes [1] to carry out the tasks to achieve their goals, and these processes need to be compliant to the relevant normative background. While institutions and businesses can last for a large amount of time, they do not remain static, but instead they constantly change during this time. Moreover, even the law changes with time. Actually, development itself has been defined as a process of changing institutions and laws. When the legal landscape goes through some change, a process that was previously compliant might become non-compliant or vice-versa. Therefore, the problem of managing change in a normative framework and its consequences to compliance is of paramount importance.

In this thesis we tackle some aspects of two different problems: the problem of translating a legal document from natural to formal language, and the problem of managing change in a formalized normative system.

Business processes can be of different kinds, let us distinguish here between *administrative processes* and *commercial processes*. Examples of administrative processes are the issue of an invoice, the registration of an invoice, the payment of a worker's wage, or the removal of a product from the inventory of a warehouse. Examples of commercial processes are sending an advertising letter or delivering a product to a client.

In all of these cases, compliance to the relevant normative background is essential, and should be actually checkable by means of a procedure. This means that we can control it, as it is potentially automatable.

The normative background can be of two kinds: *hard law* or *soft law*. Soft law is not issued by the legislator, and consists in codes of conducts or procedures that are issued by other entities. Therefore, the consequences that apply when trespassing soft law are of a different kind then the ones that apply when trespassing hard law. Examples of soft law are

- the ISO 9001 certification standard for quality management,
- the ISO 14001 certification standard for environmental management,
- the ISO/IEC 27001 certification standard for information security management, or
- the EU Council Regulation for the labeling of organic products (Council Regulation(EC) No. 834/2007).

Hard law itself is issued at different levels, and can be represented in a hierarchy of laws such as the following:

- international treaties or obligations,
- country's constitution,
- national or regional legislation (including executive orders and presidential decrees),
- local regulations.

In the hierarchy, the laws are listed in decreasing order of authority. In formal terms, such a hierarchy can be represented with a partially ordered set and the relation *lex superior*. Examples of other relations that give an order on laws are *lex posterior* and *lex specialis*, and we will describe them in a successive chapter.

There are at least two families of law: civil law and common law, and therefore it is difficult to give a general hierarchy. In countries where the legal system is based on common law, judicial precedents such as court rulings and case law carry important weight. A general intuition that we can give is that a country's constitution, which states the grounding legal and democratic principles that the people of the state have to respect, is superior to any other domestic law. International law is recognized and promoted between nations, and it also includes some rules, called *peremptory norms* (also called jus cogens) [83], that are so fundamental that they can be imposed to all states. Indeed, international law can be considered superior even to a country's constitution, and usually efforts are put in so that the international law and the constitution do not conflict.

In general, handling this normative background is a complex operation which needs specialized workers in order to be completed. If we are able to formalize such a legal system and a business process in a logical language, then it is possible to mechanically check for the compliance of the business process to the legal system. However, formalizing a legal document is a cumbersome task. If the formalization is carried out by a person, the process can be too expensive, time-consuming, complex and error-prone. Therefore, there is a need to automate such a translation process. In this thesis, we make a first step in this automation, by proposing an interactive method that automates at least the operations that are routinary. For instance, if a contract was written by using a standard template, the structure of the template can be used to successively extract the information contained in the contract.

The problem of getting a formalized legal document instead of one in a natural language can be approached in two ways:

- encode directly the document in a formal language, such as LegalRuleML [5], possibly with the help of an assisted editor, or
- translate the document form natural to formal language, (e.g., see the works in [132, 131]).

The first approach is feasible for new documents, but it does not solve the problem of pieces of law that already exist and are written in natural language. In this thesis, we follow the second approach. Since different kinds of laws present different structure and features, it will be necessary to have different specialized methods.

Once the legal document and the normative background are formalized, it is possible to check for compliance. For what regards the mechanisms of checking for compliance, we can cite the works in [71, 99, 69] for a general approach, and in [73, 110] for the research approach of *compliance by design*. However, with time a new problem may arise: norms are not static, but change. Therefore, there is a necessity for tools and methods that can manage change in normative systems. An important aspect here is to have a notion of *relevance*: to check the compliance of a specific business process, only some parts of the normative background apply. This is especially important when a norm changes: there is a need for an efficient method that checks whether a norm that was relevant becomes irrelevant or vice-versa as a consequence of change.

The work for the PhD resulted in the publication of the following articles:

- Luca Pasetto, Matteo Cristani, Francesco Olivieri, and Guido Governatori. Automated Translation of Contract Texts into Defeasible Deontic Logic (LNGAI2021)
- Matteo Cristani, Claudio Tomazzoli, Francesco Olivieri, and Luca Pasetto. An ontology of changes in normative systems from an agentive viewpoint (PAAMS2020)
- Matteo Cristani, Francesco Olivieri, and Luca Pasetto. Revising Ethical Principles and Norms in Hybrid Societies: Basic Principles and Issues (KES-AMSTA2021)
- Nicola Assolini, Adelaide Baronchelli, Matteo Cristani, Luca Pasetto, Francesco Olivieri, Roberto Ricciuti, and Claudio Tomazzoli. Text Analytics Can Predict Contract Fairness, Transparency and Applicability (WEBIST2021)
- Matteo Cristani, Luca Pasetto, Claudio Tomazzoli. Protecting the environment: a multi-agent approach to environmental monitoring (KES2020)
- Matteo Cristani, Luca Pasetto, Claudio Tomazzoli. A knowledge-intensive methodology for explainable sales prediction (KES2020)

This thesis is organized as follows. In Chapter 2 we describe the literature that is relevant to the work in this thesis and the necessary technical background of defeasible deontic logic. Chapter 3 provides a general outline of a pipeline for legal text analysis. In Chapter 4 we apply a specialized version of this pipeline to the case of *contracts that govern exploitation and use of natural resources*, giving an example of a complete translation from natural to formal language. Chapter 5 presents $\mathscr{A}$-Log, the *Adjective Logic*, by describing its syntax, semantics, proof system, proof theory and algorithms.

Chapter 6 provides an ontology of the possible ways in which a normative system can change. In Chapter 7 we study the processes by which a logical agent can acquire and learn new norms. In Chapter 8 we proceed to investigate the problem of revising a theory by only changing its facts. In Chapter 9, the different kinds of revisions treated in Chapters 6, 7 and 8 are generalized to a meta-theoretical framework. Chapter 10 is devoted to a discussion of the results of this thesis, while in Chapter 11 we give conclusions and we sketch future work.

Chapter 2

Background and Related Works

The contributions of this thesis can be summarized by the definition of the items in the following points:

- a pipeline for the translation of a legal text from natural to formal language and its application to the case of natural resources contracts,
- a logical system aimed at reasoning with gradable adjectives,
- an ontology to represent change in a normative system,
- some basic mechanisms by which an agent may acquire new norms,
- a study on the problem of revising a defeasible theory by only changing its facts and the application of this to the problem of measuring pledges between organizations,
- a general theory that includes as specific cases all the three previous points.

In this Chapter we give the necessary technical background for the rest of the thesis, we provide a motivating example that will be used again at the end of the thesis, in Chapter 10, and we provide a review of relevant related works for the points treated in the thesis.

2.1 Background

2.1.1 The Theorem Proving Problem

If we are able to formalize both a business process and its relevant normative background in a logical language, then we can check for compliance of the process to the norms. This compliance check is carried out by automated procedures that decide whether a formula is a theorem in a given logical language. In this section we give a background on the *automated theorem proving problem*.

The objective of automated theorem proving is to decide mechanically whether a *conjecture* φ is a logical consequence of a set H of *assumptions*, in symbols $H \models \varphi$. If the logic that we consider is complete, this is equivalent to searching for a *proof* of φ from H, in symbols $H \vdash \varphi$. If such a proof is found, φ is a *theorem* of H. The theorem proving decision problem for first-order logic (*Entscheidungsproblem*) was a major impetus at the origins of computer science, as Alan M. Turing invented Turing machines to show that there can be no general decision procedure for this problem (see Chapter 7 of [52], Chapter 3 of [112], or [125]).

Intuitively, the language of first-order logic is infinite as soon as the signature includes at least one function symbol, and therefore it yields an infinite search space. As it is typical in logic, there is a trade-off between expressivity and decidability. The first-order logic language is so expressive that it allows a *diagonal* approach. Specifically, a set $H_{\mathscr{T}}$ of first-order assumptions can describe the functioning of a Turing machine $\mathscr{T}$, and a conjecture $\varphi_{\mathscr{T}}$ can state that $\mathscr{T}$ halts on a given input. Then, the machine $\mathscr{T}$ halts if and only if $H_{\mathscr{T}} \models \varphi_{\mathscr{T}}$, and the *undecidable* halting problem is reduced to the theorem proving problem. In 1936, Church and Turing proved independently that, in general, we cannot decide whether φ is *not* a logical consequence of H. However, in 1930 Herbrand provided an algorithm that halts if φ is a logical consequence of H, and therefore the theorem proving problem is *semi-decidable* (see Chapter 4 of [38]).

Strategies for automated theorem proving in first-order logic are *semi-decision procedures* that, given a set of assumptions H and a conjecture φ, try to determine if $H \models \varphi$. In practice, they search for a proof of φ from H, written $H \vdash \varphi$. These procedures approach the problem by working *refutationally*, as they try to prove a contradiction from $H \cup \{\neg\varphi\}$, in symbols $H \cup \{\neg\varphi\} \vdash \bot$, or, in specific cases, to disprove φ by providing a *model* of $H \cup \{\neg\varphi\}$. The additional assumption $\neg\varphi$ provides extra valuable information. Two simplifications help the automation of the process: one is syntactic and the other is semantic. Syntactically, only formulae in clausal form are considered, where a *clause* is a disjunction of literals. Therefore, the theorem proving problem $H \cup \{\neg\varphi\}$ is transformed into a set S of clauses, where the clauses obtained from $\neg\varphi$ are called *goal clauses*. A proof of $H \cup \{\neg\varphi\} \vdash \bot$ is found when S is shown unsatisfiable.

This set S of clauses is unsatisfiable if and only if it is false under all interpretations over all domains. Since it is not possible to consider all interpretations over all domains, the semantic restriction fixes a special domain $\mathscr{U}$ such that we can consider only the interpretations over $\mathscr{U}$ to decide that S is unsatisfiable. This domain $\mathscr{U}$ is the *Herbrand universe* of S, defined as the set that contains all the *ground* (i.e., without variables) terms obtained from the constant and function symbols that appear in S. If no constant appears in S, a new constant symbol is used. If S contains function symbols, $\mathscr{U}$ is infinite. The set of ground atoms built by applying the predicate symbols of S to the terms of $\mathscr{U}$ is called the *Herbrand base* of S. A *Herbrand interpretation* $I_{\mathscr{U}}$ is an interpretation of S over $\mathscr{U}$ that assigns the constants in S to themselves and any n-ary function symbol f to a mapping from $\mathscr{U}^n$ to $\mathscr{U}$. We can view Herbrand interpretations as a syntactic characterization of a semantic notion. We can consider only Herbrand interpretations because we work on formulae in clausal form. Closely related to this result is *Herbrand's theorem*, which states that a set S of clauses is unsatisfiable if and only if there is a finite unsatisfiable set S' of ground instances of clauses of S. All theorem proving procedures implement and are inspired by Herbrand's theorem.

2.1.2 Defeasible Deontic Logic

Defeasible logic is a rule-based skeptical approach to non monotonic reasoning. It is based on a logic programming-like language and is a simple, efficient but flexible formalism capable of dealing with many intuitions of non-monotonic reasoning in a natural and meaningful way [3]. Defeasible deontic logic (DDL) is defeasible logic with deontic operators. We describe below defeasible logic with propositional literals; the features of DDL are implemented in the markup language LegalRuleML [6], that also allows to represent temporal relationships.

Consider a set PROP of propositional atoms. The set Lit = PROP $\cup\{\neg p | p \in \text{PROP}\}$ denotes the set of literals. The *complement* of a literal q is denoted by $\sim q$; if q is a positive literal p, then $\sim q$ is $\neg p$, and if q is a negative literal $\neg p$ then $\sim q$ is p.

A defeasible theory D is a tuple $(F, R, >)$. $F \subseteq$ Lit are the facts, which are always-true pieces of information. R contains three types of rules: strict rules, defeasible rules and defeaters. A rule is an expression of the form $r : A(r) \hookrightarrow C(r)$, where r is the name of the rule, the *arrow* $\hookrightarrow \in \{\rightarrow, \Rightarrow, \leadsto\}$ is to denote, resp., strict rules, defeasible rules and defeaters, $A(r)$ is the antecedent of the rule, and $C(r)$ is its consequent. A strict rule is a rule in the classical sense: whenever the antecedent holds, so does the conclusion. A defeasible rule is allowed to assert its conclusion unless there is contrary evidence to it. A defeater is a rule that cannot be used to draw any conclusion, but can provide contrary evidence to complementary conclusions. Lastly, $> \subseteq R \times R$ is a binary, antisymmetric relation, with the exact purpose of solving conflicts among rules with opposite conclusions by stating superiorities. We use the following abbreviations on R: R_s is to denote the set of strict rules in R, R_{sd} the set of strict and defeasible rules in R, and $R[q]$ the set of rules in R s.t. $C(r) = q$.

A *derivation* (or *proof*) is a finite sequence $P = P(1), \ldots, P(n)$ of *tagged literals* of the type $+\Delta q$ (q is definitely provable), $-\Delta q$ (q is definitely refuted), $+\partial q$ (q is defeasibly provable) and $-\partial q$ (q is defeasibly refuted). The proof conditions below define the logical meaning of such tagged literals. Given a proof P we use $P(n)$ to denote the n-th element of the sequence, and $P(1..n)$ denotes the first n elements of P. The symbols $+\Delta$, $-\Delta$, $+\partial$, $-\partial$ are called *proof tags*. Given a proof tag $\pm\# \in \{+\Delta, -\Delta, +\partial, -\partial\}$, the notation $D \vdash \pm\# q$ means that there is a proof P in D such that $P(n) = \pm\# q$ for an index n.

In what follows we only present the proof conditions for the positive tags: the negative ones are obtained via the principle of *strong negation*. This is closely related to the function that simplifies a formula by moving all negations to an innermost position in the resulting formula, and replaces the positive tags with the respective negative tags, and the other way around.

The proof conditions for $+\Delta$ describe just forward chaining of strict rules.

$+\Delta$: If $P(n+1) = +\Delta q$ then either
(1) $q \in F$, or
(2) $\exists r \in R_s[q]$ s.t. $\forall a \in A(r).\ +\Delta a \in P(1..n)$.

Literal q is definitely provable if either (1) it is a fact, or (2) there is a strict rule for q, whose antecedents have all been definitely proved. Literal q is definitely refuted if (1) it is not a fact and (2) every strict rule for q has at least one definitely refuted antecedent.

The conditions to establish a defeasible proof $+\partial$ have a structure similar to arguments in natural language, where an argument might provide support for its conclusion but not be deductively valid in general, because it is defeated by a stronger counter-argument.

$+\partial$: If $P(n+1) = +\partial q$ then either
(1) $+\Delta q \in P(1..n)$, or
(2) (2.1) $-\Delta{\sim}q \in P(1..n)$ and
(2.2) $\exists r \in R_{sd}[q]$ s.t. $\forall a \in A(r) : +\partial a \in P(1..n)$, and

(2.3) $\forall s \in R[\sim q]$. either
(2.3.1) $\exists b \in A(s) : -\partial b \in P(1..n)$, or
(2.3.2) $\exists t \in R[q]$ s.t. $\forall c \in A(t) : +\partial c \in P(1..n)$ and $t > s$

A literal q is defeasibly proved if, naturally, it has already strictly proved. Otherwise, we need to use the defeasible part of the theory. Thus, first, the opposite literal cannot be strictly proved (2.1). Then, there must exist an applicable rule supporting such a conclusion, where a rule is applicable when all its antecedents have been proved within the current derivation step. We need to check that all counter-arguments, i.e., rules supporting the opposite, are either discarded (condition (2.3.1), at least one of their premises has been defeasibly rejected), or defeated by a stronger, applicable rule for the conclusion we want to prove (2.3.2).
As common in modal logic, the modals are dualized: in the specific case of deontic logic, given a literal l, $\mathscr{O} l \Leftrightarrow \mathscr{F} \sim l$ and $\mathscr{O} \sim l \Leftrightarrow \mathscr{F} l$.

2.2 Motivation: a Regulation Example

In this section we show an example of a regulation, and we describe some of the problems that we aim at tackling in this thesis. The example will be considered again in Chapter 10, where we apply the contents of the thesis to solve the problems presented here.

Consider the following excerpt of an on-board regulation of a public transport company. The regulation has to be accepted and is applied to everyone that accesses a bus of the company.

Access to the bus

1. It is mandatory to board from the front door
2. It is forbidden to board when the vehicle is moving
3. It is mandatory to have a travel document or to purchase a ticket on board of the bus
4. A travel document can be a ticket or a subscription
5. Tickets purchased on board have an extra charge of 1.00 €
6. It is mandatory to validate the ticket
7. A ticket which is not validated is equivalent to travelling without a travel document

Travel behaviour

8. It is forbidden to smoke
9. It is forbidden to hold disrespectful or indecorous behaviour
10. It is forbidden to hold behaviour that can harm the safety of the service

Penalties

11. Travel documents must be exhibited at the request of drivers or inspectors
12. Passengers without a travel document shall pay a fine of 100.00 €
13. Passengers without a travel document that pay while on the bus shall pay a fine of 80.00 €
14. Passengers that smoke shall pay a fine of 150.00 €
15. Passengers that hold disrespectful or indecorous behaviour shall leave the bus
16. Passengers that hold behaviour that can harm the safety of the service shall pay a fine of 250.00 €

Animal transport

17. Pets travel for a fee
18. Pets shall wear a muzzle and be kept on a leash
19. Pets accompanying visually impaired persons are an exception, they travel free of charge and can be without a muzzle

Luggage transport

20. Standard luggage does not exceed weight of 10 kg
21. Heavy luggage exceeds weight of 10 kg and does not exceed weight of 20 kg
22. Very heavy luggage exceeds weight of 20 kg
23. It is permitted to carry standard luggage free of charge
24. It is permitted to carry heavy luggage with payment of fare #1
25. It is permitted to carry very heavy luggage with payment of fare #2
26. One extra bag can be transported with payment of a special fare
27. For passengers with an extra bag, it is mandatory to put one bag in the luggage compartment: the bag in the luggage compartment has to be heavier than the other bag

2.2.1 Problems

Given this legal text in natural language (English), the problems that we want to approach are:

- translating from natural to formal language;
- reasoning in the formal language, with the goal of checking for compliance;
- reasoning with the adjectives that appear in the text;
- managing change in the norm.

In Chapters 3 and 4 we give and apply a pipeline for the translation to a formal language, so that we can reason on the text. For instance, consider the following situation: a passenger boards form the front door, has a ticket but does not validate it, and smokes on the bus. The inspector gets on the bus and the passenger is ready to immediately pay the fine. Is the passenger compliant to the regulation? Does he get any penalties?

The main formal language that we adopt is that of defeasible deontic logic. Adjectives are important in defining the contents of any text, and specifically of a legal text, but they present some specificities that are not captured by defeasible deontic logic. For this reason, in Chapter 5 we propose a logical system with the special purpose of reasoning with adjectives. For instance, the example described needs this mechanism to reason with adjectives such as *heavy* or *unsafe*.

Also the problem of the management of change in the norm is not trivial. For instance, things that may change are the definition of what a travel document is, or the perception of what a specific adjective means. Another example is that the perception of the adjective *heavy* may change for the inspector agent because she checks the weights of the two bags with a faulty scale. In Chapters 6, 7, 8, and 9 we treat different aspects of norm change.

2.3 Related Works

2.3.1 The Legal Text Analysis Pipeline

Computational methods applied to the law can be distinguished in two general approaches (see [59]): the *law-as-code* approach and the *law-as-data* approach. The law-as-code approach aims at interpreting and representing legal rules in a formal language, such as defeasible deontic logic [108]. An example is that of [20], where the authors provide a method for encoding traffic regulatory rules, or that of [51], where changes in normative systems are modelled in a formal manner.

On the other hand, the law-as-data approach aims at extracting information from high-dimensional legal datasets. This approach can be applied to many problems in the legal context, and applications can range from interpretation of legal texts to quantitative analysis of external factors that influence the law. In this regard, there exist different pattern-based approaches to assign labels to (parts of) the text of legal documents.

The work that we describe in this thesis sits in the middle between the two approaches: the goal of this research is to start from a natural-language legal text and to interpret it and synthesize it to rules in defeasible deontic logic.

There have been many investigations concerning the development of specific pipelines for regulatory texts, and some studies on specific parts of such a pipeline [97]. We cite the recent GarNLP pipeline [25], that aims at automatically processing garnishment documents in two phases: the document is first categorized onto a predefined taxonomy, and then various relevant information is extracted from the text. Solon [91] is a legal document management platform that can be used to model, manage, and mine legal sources by extracting semantic representations of them. A related problem is that of assigning labels to the text of legal documents. We cite here a number of approaches to this kind of semantic processing of legal texts. Work in [92][98] combines learning and reasoning to automatically detect and explain unfair clauses in Terms of Services of online consumer contracts, while [40] presents a methodology and an annotated corpus for processing GDPR privacy policies and checking compliance. In [37] several methods to extract contract elements are explored and evaluated.

Much work has been done on legal decisions or judgments. The system Salomon [128] processes Belgian criminal cases by performing an initial categorisation and structuring of the texts and an extraction of the most relevant text units. In [124] authors perform automatic categorization of case law documents into 40 high-level categories. The research in [81] is applied to judgments of the UK House of Lords: the rhetorical status of sentences is predicted and some of them are selected to provide a summarisation of the document. In [60] legal case reports are automatically summarized by combining different techniques in a rule-based system with a knowledge base.

Topic models [22] are an approach to describe legal documents as distributions over subject matters (or topics), that are identified in unstructured corpora of texts as distributions of words over the vocabulary. The frequency distribution of these words (and of the citations) can then be compared to classify documents, for instance in [36] topic modelling is applied to judgments of the High Court of Australia. On the other hand, in [53] a classifier is built and applied to Dutch law texts as a basis to perform automatic modelling of the law. In [85] authors give a conceptual framework for modelling document semantics and an ontological extension for the legal domain.

Authors in [8] developed a system that classifies paragraphs according to regulatory content and extracts relevant information, and apply it to Italian law texts. On the other hand, in [68] linguistic information such as lemmatisation and part-of-speech tags is used to improve the classification of Portuguese legal texts. Authors in [127] propose a method for classifying the subcomponents of the writing styles that bound German legal language, and a specialized annotated corpus is again necessary.

Other applications of quantitative text analysis have been used to examine how the content of legal texts changes over time, for instance [35] analyses the evolution of Supreme Court opinions and [89] the relationship between law and society; or to analyse the qualitative notion of *legal complexity*, as in [115] or [88].

2.3.2 Reasoning with Gradable Adjectives

The main initial claim of the research whose first step is documented in this study is to devise a propositional method to deal with descriptive adjectives, that constitute a significant part of the semantics of natural language. In this thesis we have been focusing upon the *intersective gradable* adjectives.

It has emerged that the main differences between adjectives and nouns are four:

– Adjectives' semantics is not always subsective, as this is not always the case that when an adjective is attributed to a noun, then the object referred to in the noun phrase belongs to the set that interprets that noun. For instance, *alleged murderer* is not necessarily a murderer, and *fictitious doctor* is certainly not a doctor. The exception to this is the set of subsective adjectives;
– Adjectives do not always allow rewriting to nominal sentences. For instance, a *clever surgeon* should not be asserted to be clever in general, because there may be domains where she is not competent. Rewriting in nominal sentences is allowed for intersective adjectives;
– Adjectives are often crisp, but others have *vague* meaning. For instance, *exhausted* is not vague, whilst *large* is so. Gradable adjectives are typically vague, at least in a number of usages, but there are cases in which a gradable adjective is used in a context where it becomes crisp, as the adjective *old* in a legal context, where someone is crisply old when their age is higher than 70, for instance;
– Adjectives are *polarised* in a large number of cases because they possess their *antonym* that sometimes corresponds to their negation. This does not happen for the majority of nouns. Can you devise the antonym of *silk*? Moreover, many antonyms of nouns do not have the same purpose of antonyms of adjectives, for they are used to describe an object that has a different character, possibly a different purpose, but not necessarily a different orientation in terms of measures. For instance the antonyms broad/narrow design breadth from top to bottom, whilst the antonyms civilian/military apply to a completely different situation.

Vagueness is crucial in the definition of semantics, therefore references to studies on the nature of vagueness have been fundamental in developing this research. In particular, various studies have devised a general view of vagueness in artificial intelligence ([56]; [100]; [111]). Some studies have provided a direct analysis of the aspects of vagueness in adjectives, as in [82] and [96].

A few investigations by [84] and by [105] have addressed early the topic of reasoning with adjectives. However, these two foundational works have two main limits that we are trying to address here: first of all, none of these investigations have focused upon a *general* viewpoint on gradable adjectives, and specifically on the *intersective* nature, with all the distinguished aspects that we dealt with in this work, and, secondly, these investigations did not address computational specific aspects.

Further investigations have addressed some of the topics we discussed, but again, not in general terms. Specifically, there is an interesting investigation on some empirical properties that adjectives exhibit by [32]. In that paper some of the inferences that we identified as relevant are studied, as well as others.

Apart from the aforementioned researches, there is one specific group of scholars who are responsible for having provided a strategic view on adjectives' aspects of language and deserve to be recognized as the initiators of this approach: Brandon Bennett and his collaborators. The studies on vagueness of Bennett and collaborators have focused upon spatial vagueness on one side ([18, 19, 12, 13, 15]), and on vagueness as a source of semantics for adjectives on the other ([14, 16, 17]).

One of the foundational aspects of Bennett's view of adjectives' semantics is the notion of *individual variability*, that we have captured with a notion of *agency* based on labelled logics. Overall, the idea is taken from studies on meaning negotiation conducted in the past by Cristani and Burato ([30, 31, 47]). Supposedly, when two individuals discuss upon the meaning of a sentence, for they cannot derive a common conclusion, we assume that the process ends up with some sort of disagreement. However, this type of disagreement is not *unsolvable* or solvable only by means of *context* or by *meaning negotiation*: the disagreement may be solved just by *implicit understanding*. For instance, consider the disagreement between two judgments of positive and negative nature of two distinct individuals where one says that "John is tall" and the other disagrees on this with "No, he is not". In this case, we derive the conclusion directly from the sentence, due to the nature of the adjective *tall*: the two interlocutors have different thresholds on the adjective *tall*. The above would be named, often, an *understood disagreement*.

The problem of establishing the connection between the meaning of a theory and the meaning of the terms with representation of disagreements in a non-classical fashion has been the subject of a long stream of investigations, including those in the theory of contexts as in the studies by [123], by [65] and by [10], and specifically in the theory of contexts in defeasible logic by [21].

Interference of adjectives with nouns are also affecting reasoning with classes, as in the Description Logic framework by [26, 64]. Adjectives have also been investigated in research regarding machine learning, co-occurrences and in application fields of various nature, with a very strong presence of adjectives in studies regarding Recommender Systems, Social Network Analysis and Sentiment Analysis.

The limits of a purely logic approach to a problem like the one of adjectives have been dealt with in the current literature on *semantic social network analysis* in the reference investigations by [86] and [67], and also in the studies regarding semantic social network analysis as a means for sentiment analysis [43, 48]. The application demand is paired with the problem of forming vocabularies with correct annotations for adjectives, as studied by [63] in general, and by Cristani et al. ([49, 50]) in more practical terms.

2.3.3 Acquisition and Learning of Principles and Norms

Some effort has been posed by scholars in the field to identify methods able to revise defeasible logic, and in general non monotonic frameworks. [24, 23] propose a Defeasible Logic framework to model extensive and restrictive legal interpretation. This is achieved by using revision mechanisms on constitutive rules, where the mechanism is defined to change the strength of existing constitutive rules.

Also on revision of Defeasible Logic we can find [79], where the key idea is to model revision operators corresponding to typical changes in the legal domain, specifically, abrogation and annulment. Further on this, some investigation effort has been posed on temporary norm change [45], and more generally on the ways in which those changes can be performed [51].

Non-monotonic revision through argumentation was also investigated in [103] by using Defeasible Logic Programming (DELP).

Other works related to the development of revision methods are [113, 4, 102]. In [4], it is possible to have rules of the form $r : a \Rightarrow (s > t)$ where s and t are identifiers for rules. Accordingly, to assert that rule s is stronger than rule t we have to be able to prove $+\partial a$ and that there are no applicable rules for $\neg(s > t)$. In addition, the inference rules require that instances of the superiority relation are provable (e.g., $+\partial(s > t)$) instead of being simply given (as facts) in $>$, that is $(s,t) \in >$. This is the base for a method of revision for preferences (see [72, 77, 122, 109]).

Chapter 3

The Legal Text Analysis Pipeline

In the majority of real-world cases, processing texts is a twofold activity: we have to take in consideration both the *content* of the text and the *context* of the text. When we look at a text, in fact, we are considering it as a *source of information*, but in order to obtain this information, we need to know *how* to look for it: we need to know *what* is contained in the document, and *where* we should look for its context. The ability to perform such a search typically depends upon the domain of knowledge that the text pertains to.

Texts, in other terms, need contexts to be understood. Essentially, text lexicon, format, content and structure are *domain* dependent but also *task* dependent. In fact, their meaning needs to be related to the ontological layers [101] that are actually pertinent. For instance, a medical document containing a prescription belongs to the ontological domain of *medicine* and to the task layer of *health management*, for its task is to give operational orders to the health system, not to devise a diagnosis or a therapy (that may be contained in different documents).

The case in which a specific single text contains information coming from diverse layers is rather difficult, as also the case in which more than one document contains pieces of the information that can be referred to a single layer.

Specialized languages differ from other specialized languages in their lexicon and because of the the typical document types and structures that characterize them. Plenty of examples can be found in the medical or in the legal language.

Consider, for instance, the text included in Box 1. A large number of people reading this document may have serious difficulties in understanding the exact legal terms included in it, but they might grasp a neat understanding of the following facts, that emerge from the lexicon, structure, and other contextual tokens:

1. It is a legal document;
2. It is connected to other existing documents;
3. It contains reference to two parties, one issuing and one demanding;
4. There are legal terms issued, including, in particular, limits to the content of the issue;
5. There are Institutions cited, and roles are referred to them;
6. There are monetary amounts corresponding to rent, royalty, insurance, and bond.

A priori knowledge of the domain could be helpful in determining the bounds of the above mentioned concepts to the elements of the text. However, many of these activities

RESPONSE TO APPLICATION FOR RENEWAL OF A MINERAL EXTRACTION LEASE FOR MINERALS OTHER THAN OIL, GAS, OR GEOTHERMAL RESOURCES

APPLICANT: U.S. Borax Inc.
AREA, LAND TYPE, AND LOCATION: Approximately 15,534 acres of State sovereign land located on the dry lakebed of Owens Lake in western Inyo County. This sovereign land is located east of U.S. Highway 395, and approximately 10 miles south of the city of Lone Pine (see Exhibits A and B, attached).
AUTHORIZED USE: Extraction of industrial minerals, a sodium carbonate mineral (trona), and any other mineral deposits except oil, gas, other hydrocarbons, and geothermal resources.
LEASE TERM: This is the second renewal with a lease term of 10 years, effective August 1, 2018, through July 31, 2028, with a preferential right to renew for two additional terms, not to exceed 10 years each, upon such reasonable terms and conditions as may be prescribed by the Commission.
CONSIDERATION: The State's royalty is 10 percent for unprocessed trona and 7 percent for processed trona. Annual rent of $2.50 per acre or $38,858 for a total of 15,543 acres and a minimum annual royalty of $120,000.
INSURANCE: In an amount not less than $2,000,000.
BOND: In an amount of $60,000 to guarantee the faithful performance of Lessee's lease requirements.

Box 1: A legal text: excerpt from a contract (http://resourcecontracts.org).

are related to *information retrieval* and *knowledge extraction*. For instance, consider point 1 in the above text. Clearly, the document has a legal nature because it contains elements that can be summarized in the following list:

- Legal *lexical tokens*, including proper legal terms, such as *sovereign* or *State*, or *requirement*;
- Legal binding terms, such as *right* and *prescribed*;
- Expressions of legal conflicts, in particular *except*, *preferential*;
- Specific tokens for a specific document type, for instance a concession, that is the specific case of the document here, such as *royalty*, *bond*, *authorized*, and *term*;
- Numeric values close to descriptions of elements in the discussion, for instance *10 per cent* close to *royalty*;
- Specific tokens for a specific domain of concession, in this case, extraction of minerals, that are the result of the instersection of a technical language with a legal language, for instance *trona* or *extraction*.

It is easy to note that all the above mentioned patterns can be detected without effectively understanding the document content. However, without a correct contextualization it would be hard to identify the meaning of the sentences in the text.

Now, consider the following research aim: starting from a legal document, transform it into a token or a set of tokens in a formal language, so that we can process it in an automated way in order to verify its coherence and the compliance to other implemented legal tokens, such as the normative background constituted by the laws in force in that specific legal context.

This research aim has been pursued in many actual efforts, with a particular emphasis on the structure of the target language [20, 93]. We commit ourselves to formalization into *defeasible logic* as proposed in [131] and base the methodology for the translation on the researches documented in [114] for the language issues, and [107] for technological issues.

It is therefore worth to devise a pipeline of activities with the final purpose of the formalization of the text in a legal-driven logic language represented in a XML-serialised markup framework, known as LegalRuleML, that we describe in the next section. The above is a long-term research aim, that makes sense only when we consider the foundational notions of document processing. The amount of documents that have a legal interest is gigantic. In principle, once we shall have defined a correct pipeline for processing documents, it will be possible to consider a document repository containing documents of various types, detect the legally significant ones, and then process them.

In this Chapter we discuss some issues about the correct definition of pipelines and show that it is rather impractical to imagine the above described pipeline for the general case. It makes much more sense to evaluate a situation in which we have already stepped down two levels in the ontological layers of legal document analysis: the domain of knowledge, and the task of the documents (for an analysis of the principles to be used in this case see [41, 61]). In other terms, we shall devise a specific methodology for processing corpora of documents that we already know to be of legal nature, and that refer to a single application domain.

3.1 Basic Definitions: LegalRuleML

LegalRuleML extends RuleML by implementing specific features of the legal domain in a rich markup language. Some of its most important features and functionalities are:

1. Deontic concepts, such as obligations, permissions, prohibitions, and even more articulated effects. Rules can detect violations, and which effects are triggered by such violations (these are called *reparation chains*).
2. It implements defeasibility, and therefore it can handle exceptions. In the law, there are prescriptive behaviours that apply under some circumstances but not under others. Accordingly, inherited from RuleML, LegalRuleML can handle and resolve conflicts.
3. Semantic management of negation.
4. It models different types of rules, in particular *constitutive* and *prescriptive* rules. Constitutive rules model concepts or institutional actions (w.r.t. the example illustrated in Box 1, a constitutive rule may define what is intended by ‘industrial minerals’, and another for ‘geothermal resource’); prescriptive rules tell which actions and their outcomes are permitted, mandatory, or forbidden.
5. Multiple semantic annotations: a legal rule may have multiple, different annotations to represent different situations/scenarios/legal interpretations.

6. It allows to link rules to provisions. LegalRuleML includes a mechanism, based on IRI, that allows many to many (N:M) relationships among the rules and the textual provisions: multiple rules are embedded in the same provision, several provisions contribute to the same rule.
7. Temporal management: in LegalRuleML, all entities (rules, references to texts, etc.) can vary through time, and the temporal relationships are represented and handled in an unambiguous way.

When translating a legal corpus, the user produces three different documents. First, a vocabulary that contains *atomic* definitions. An entry in the vocabulary has: (i) a unique label/identifier called *key*, (ii) the textual name X of the atom itself (within command $< atom > X < /atom >$), and (iii) its textual description, typically taken form the referenced normative corpus. With reference to the example in Box 1, ‘city of Lone Pine’ will have $key = cityLonePine$, atom $< atom > cityofLonePine < /atom >$, and whichever textual description is reported in the legal corpus (possibly linked to pictures of maps or other types of tokens we do not treat in this study). Note that the vocabulary does not contain only atomic concepts, but also non-atomic ones that are described below.

Second, the (main) document containing all the constitutive and prescriptive rules: such a document will also contain rules defining non-atomic (constitutive) concepts (such rules are often referred to as *count-as* rules). For instance, both oil and gas are hydrocarbons, and thus a constitutive rule defining what is an hydrocarbon will have a content of the form: “`If` variable X is oil `or` X is gas `Then` X ‘count-as’ hydrocarbon”.

Third, an association document that links rules (entities in the second document) to their provisions (in the original corpus). This document is fundamental as within the same norm/policy, different prescriptive behaviours are often described to tackle different applicability conditions and exceptions. We thus need to link all rules formalising such a prescription.

3.2 Goals of Legal Text Analysis

In order to be effective in devising a pipeline for text analysis we generally have to focus on the result of the analytical process on the one hand, and on the queries we aim at extracting on the other hand. Consider the case presented in Box 1, and assume that we look at the text in order to translate in into a formal language. Clearly, it would be essential to understand what one can find in the text, in terms of elements to be represented in the target system, but this would not be enough to establish a correct process of knowledge extraction. For instance, if we look at a clause in a legal text that is a contract, we see the clause as what in LegalRuleML is named a *prescriptive* rule, but elements of the rule can be defined in *constitutive* rules, and we may also have *references* to rules that belong to the normative background, or to other related documents, both affecting the constitutive rules and the prescriptive ones. When translating rules we naturally have side effects due to the usage of terms that are not defined in the text itself. These can be *general* terms, or *specific* terms. In the second case, we need to refer those terms to a provided ontology of legal terms, that is stratified. More specifically, terms occurring in a text can be of four levels:

- General terms, belonging to the general dictionary of the language in which the legal document is written;
- General legal terms, commonly employed in legal documents;
- Domain specific terms, that are typical of the technical matter;
- Task specific terms, that are typical of the document type.

In fact, legal texts belong to many different categories, and, because of their specific nature, they have received due attention in the current literature.

First of all, we should note that every single legal system has diachronically created a set of admissible document concepts, that form the specific corpora of documents of a country or a group of countries. Therefore, it is not possible to devise a *general* typology of documents, for these types concern the specific legal system. As an example, the website https://www.lawdepot.com can be used to build a legal document while starting from typologies in several different countries (Australia, Canada, Germany, India, Ireland, New Zealand, Nigeria, Singapore, The United Kinkdom, The United States of America). We can, however, make some general distinctions, based upon foundational works on the nature of legal concepts [104, 129, 11, 28, 94, 95, 61, 41].

Legal documents can be distinguished, in a very general sense, by their *origin*, *nature*, and *form*. Documents can be originally public or private, have a numeric or non-numeric nature and can be in written form or not. Moreover, we can derive a very general concept of a legal document as a *normative* document, issued by an *authority*, or a *non-normative* document whose effect can be variable, depending on the legal system, but also based on the nature of the activity driven by the document.

In particular, non-normative documents may regard operations such as:

- *Deliberations* made by public bodies on issues they have some reason to decide about (for instance, a sentence in a trial, but also an administration decision on a petition)
- *Demands* posed by a private subject to a public body, including *lawsuits*, *petitions*, *applications*;
- *Demonstrations* proposed by a private subject about an act or a fact represented by the document itself, including *evidences* for trials but also business documents such as *transportation documents*, *invoices*, *receipts* and many other *deeds*;
- *Agreements* among private subjects including *contracts*, *memoranda of understanding*, *Non-disclosure agreements* and many other ones;
- *Decisions* made by private subjects regarding their wealth, life or other aspects that are under their control, including *legatos*, *testaments*, *do not resuscitate declarations* and *power of attorney*.

The typologies devised above, that are the result of a synthesis work upon the classifications in the above mentioned systematic analyses of legal terms, allow to descend the layers of the hierarchy of legal ontologies as shown in Figure 3.1.

3.3 Legal Texts at a Glance: an Ontological Perspective

In practice, what we said so far boils down to the fact that the first step in the analysis of the structure of a legal text is to obtain some basic information about the document. This

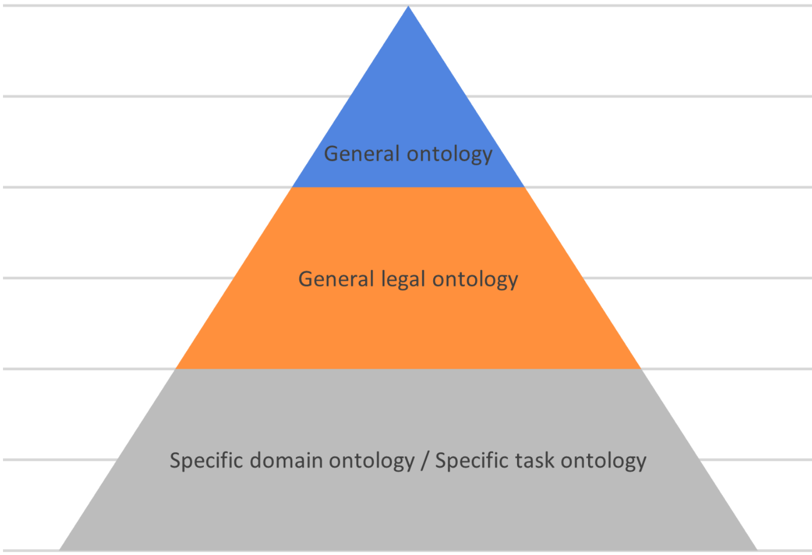

Figure 3.1: The four layers of a legal ontology for terms to be processed in document analysis.

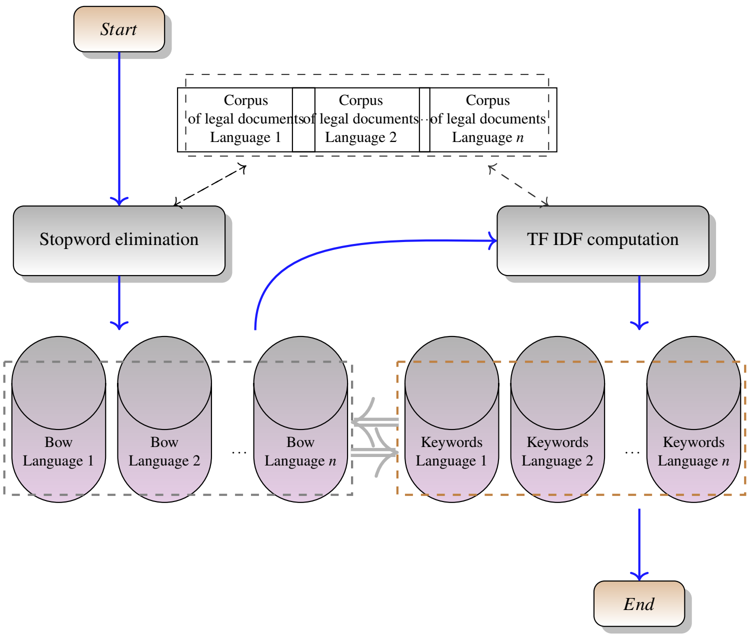

Figure 3.2: The general pipeline of text analysis: creation of the classification system

can be achieved either by using external information or by means of text processing with machine learning and information retrieval methods.

We can, in other terms, divide the process of analyzing a legal document in two phases, followed by the automated conversion to a formal language (such as LegalRuleML):

– **Document classification**. By machine learning techniques we process the document within a set of reference corpora to identify the type of document. To each type of document we attach a *template* that consists in the description of the document structure by a sequence of segments, each forming one part of the document itself,

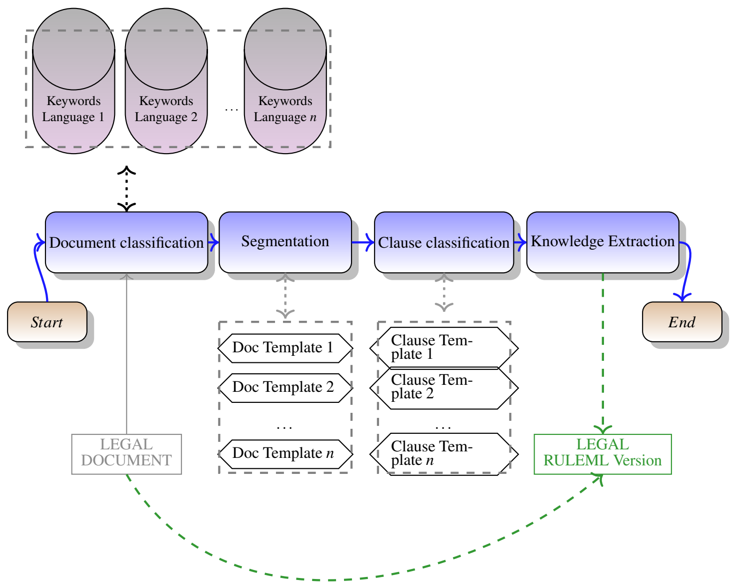

Figure 3.3: The pipeline of legal text analysis: application of the classification system

and containing, in turn, a sequence of *elements*.

– **Segmentation**. By information retrieval techniques, we extract the elements from each segment. Each element is attached with an element class among the following:
  - **Constitutive elements**: definition of terms;
  - **External references**: links to other documents, possibly with a specific reference to a single element in the referred document;
  - **Normative background references**: links to the normative background;
  - **Prescriptive elements**: rules specifying constraints to the behaviour of the subjects on which the legal document takes effect:
    - **Obligations**;
    - **Prohibitions**;
    - **Permissions**.

Once the first phase of analysis is performed, the single elements of the document can be translated into elements of LegalRuleML by applying to them specific symbolic translation methods, in particular the symbolic analysis based on the Parts-of-speech tagging, Named Entity Recognition and Syntactic Tree Parsing [33], that can be also partly supported by machine learning methods [39]. The goal is to obtain a first kind of logical representation that is to be converted into rules based on specific schemata, but there is also the option to construct the rules directly from the text.

The symbolic pre-processing and the direct translation approach have been tested, so far, only partly. For instance [107] approached the problem by symbolic pre-processing, and [131] dealt with a direct translation approach but no comparison of the performance has been attempted yet. There is, however, a good argument in favour of the direct translation approach: specificity of regulatory texts. Legal texts contain prescriptions that have the intrinsic nature of *if-then* rules, also called *conditionals* in linguistic terms. Therefore, it would be more consistent to define a method based on second-level templates, forming schemata to translate from text to LegalRuleML single constitutive or prescriptive elements of the segmented text.

Segmentation of legal texts can be rather puzzling in practice, for it may involve tokens other than linguistic, for instance graphic ones, or representation of elements that are employed for devising the parts themselves in a way that is specific of the particular context. In the chosen example, we can see that some of the segments trigger translations to a specific sub-type of *contract obligations* that are usually named *value clauses*. However, the element considered is much more a deliberation than a contract, for it corresponds to the answer on a request.

Clearly, no general pipeline can be devised, for the specificity of the translation schemata for the above elements depends on variables such as language, normative background, type of document, or technical domain.

3.4 A Methodology for the Definition of the Pipeline

The three main issues of the methodology are:

1. It is necessary to use a *context* in order to know the kind of legal document we are treating. Following the ontological layers defined above, we need to be able to treat legal terms and to perform the correct binding. For instance, words such as *fruitful* or *extension* may have different meanings depending on the context.
2. It is necessary to have the availability of the *corpus* of legal documents we are referring to, that gives the context.
3. Also the *normative background* has to be available, and in certain contexts more than one normative background may apply.

In the next chapter, we will see an application of the methodology hereby defined to the specific case of contracts involving natural resources.

Chapter 4

A Method for Rules Extraction from Contract Texts

Sources of legal knowledge are usually written in Natural Language, and in this way they form the **legal documents**. On the other hand, formalisms that can be treated by computational engines, such as *defeasible deontic logic*, lie at the opposite side of the line that goes from informal to formal structures that can be used to deploy juridically relevant information. Current methods to concretely transfer to formal language the knowledge contained in legal documents are of two kinds, both imperfect:

- Translation by hand, that can be rather accurate but definitely not sustainable in practice, as it requires a lasting effort by highly qualified personnel.
- Methods based on computer-based automation, that have been attempted in numerous studies, but that still give rather inaccurate, and in several cases erroneous, results.

Some investigations have proven that it is possible to devise a correct pipeline for the above mentioned goal, and that the accuracy limits of the automated translation can be overcome. In particular, starting from the pioneering work of Wyner and Peters [132], an enhanced method was developed collaborating with Governatori [131]. Subsequently, Camilleri et al. [34] investigated the general issues related to detecting legal knowledge in legal documents, and further some studies have looked at the pipeline to perform that process [107, 55, 58].

However, although these methods have shown some positive progress over the years, there are still several issues to solve in order to provide an appropriate overall method for the translation process. In particular, this research addresses the technical aspects of the translation from natural sentences in legal documents to formal language that emerged in the studies cited above, but remained open.

When we process a legal document to formalize it, we start by executing the basic operations of the typical pipeline of natural language processing, in summary: (1) Tokenisation, (2) Part-of-speech tagging, (3) Syntactic tree generation, (4) Translation into formal language.

One well-known source of computational complexity of the above process is Step (3), that is due, in turn, to the possible syntactic/semantic ambiguity generated in Step (2). The number of syntactic trees to be explored can be exponential. Another known source of complexity for natural language processing is constituted by *anaphoras*, in particular *pronoun* anaphoras, *noun* anaphoras or *elliptic* ones. These are very costly in natural language processing, but, fortunately, we can overlook them when treating legal documents. It is in fact essentially anomalous in legal texts to incorporate references that are not explicit, and therefore in these documents anaphoras are rare, if not inexistent.

Clearly, when dealing with the translation process of legal documents, the fundamental steps to devise depend on the structure of the target formal language. In particular, we aim at devising methods to identify modal operators that represent *obligations*, *permissions* and *prohibitions* and operators that assert *exceptions* to other deontic rules. We also need to detect which tokens can be relevant to the above mentioned deontic operators. Among these tokens we should particularly consider, as usual in syntactic tree generation, nouns and verbs (excluding modal operators that are already considered in the previous step). Moreover, we want to identify *noun phrases*, that are commonly used in place of nouns.

Once the correct pipeline for the process described above is determined, we are able to build an experimental test with human subjects to test the validity of the method; this experiment is still on its way. In order to achieve this long-term goal, first we need to check the correctness of the pipeline on some examples. In this Chapter we provide a proof-of-concept of the method by implementing the techniques mentioned above to the specific case of contracts and other documents for *natural resources* that are contained in a stable and continuously fed document repository publicly available on the web[1]. We chose one specific case where we have been able to operate the entire pipeline in a semi-automated way, employing the GATE text analysis tools and human analysis. The resulting pipeline is analysed at a high level to identify drawbacks and advantages, and to design the successive step of the process: the experimental phase.

The rest of the Chapter is organized as follows. Section 4.1 describes the adopted approach and gives more details about the pipeline. Section 4.2 shows the application of the method to a specific case: an exploration permit issued to three companies by the government of the State of Western Australia.

4.1 Approach

In this section we describe the specific method of natural language processing that we are implementing, and specify for each part of the process what has already been completed and what is still under development. Furthermore, we also specify how we aim to solve the parts of the pipeline that are not completed yet.

The general schema of document classification and knowledge extraction techniques that summarizes the existing methods of the current literature is presented in Figure 4.1.

In the pipeline we introduce the concept of *template* in order to identify (1) patterns of interpretation for legal document categories and (2) segments within one category, that can also be called call *clauses*, as in Figure 4.1. In the specific case of *exploration permits* that we analyse in this Chapter, there are essentially only three segments in terms of document template: the introduction or *preamble* segment, where the parts of the permit and its nature are declared; the *interpretation* segment, that contains both *references* to the relevant normative background and *internal definitions* to the relevant terms; and the

[1] www.resourcecontracts.org

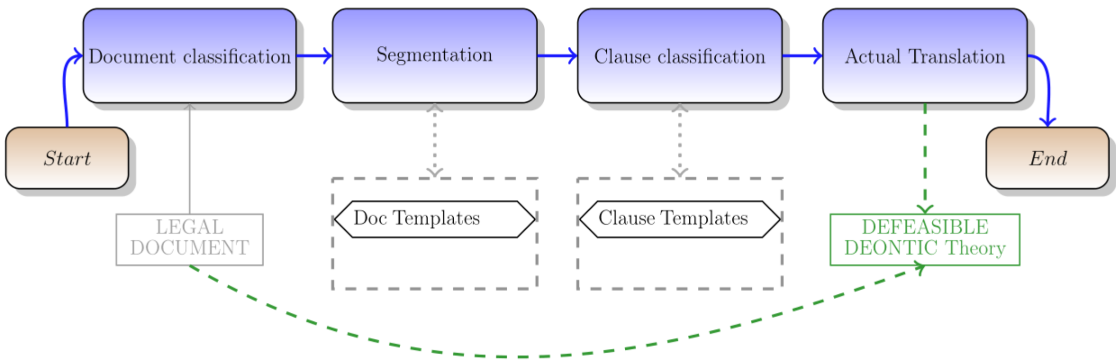

Figure 4.1: The pipeline to treat legal texts.

prescriptive segment, where the permittee is assigned with specific duties. If a term does not appear in the interpretation segment because it is not defined nor referenced there, then this term should be treated with a *common sense interpretation*. We dismiss this aspect that is very important, but concerns the application of a formalized piece of legal text much more than the subject of this investigation, that is, the translation in formalized language.

When processing a legal document, the performance of the knowledge extraction process is enhanced by knowing *a priori* the category of the document and consequently its structure in terms of segments. Therefore we have constructed a system of classification that acts on a corpus with a priori knowledge of the assumed *homogeneity*. Clearly, it makes sense to identify the properties of the corpora that are useful to the above pipeline, and not those that cannot be used in practice. Therefore, in order to organize correctly the second experimental phase we firstly need to devise a functional pipeline for the legal document from text to formal language, and then define a classification system for templates.

The considered approach involves a pipeline that consists of the following steps:

1. Classify the document in order to identify its *type*, *language* and *normative background*. Typically, the type can be inferred from the title of the document, the language can be detected by computing the frequencies of the words in the text, or as an *a priori* knowledge and the normative background could be identified by nouns that provide references to Authorities that can be found by Named Entity Recognition. We are also interested in the other legal texts that are usually referenced at the beginning or at the ending of the document, or, often, by specific reference formulae. At this step, the document can also be segmented in order to apply what follows only to specific portions of the text.
2. Apply *named-entity recognition (NER)* to extract entities that are mentioned in the text. These are the relevant terms that appear in the text and that are to be used in the translation. In particular we performed regular NER, in order to detect *dates*, *proper nouns*, *organisations* and *money*. We also implemented a recognizer for specific tokens, in particular *modalities* and specific expressions typical of the legal domain. This step is carried out by the usage of knowledge extraction and information retrieval methods and aims at extracting:

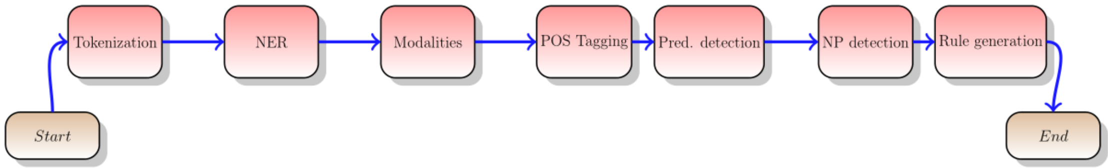

Figure 4.2: The specific translation pipeline from legal text to Defeasible Deontic Logic.

- Constitutive elements: definitions of terms that are used in the text.
- Prescriptive elements: rules that constrain the behaviour of the subjects involved in the legal document, they can be obligations, prohibitions, permissions, or exceptions to those.
- External references: links to other documents, with possibly a reference to specific terms or rules in these documents.
- Normative background references: links to the normative background that was in force when the document was drafted.

3. *Translate to DDL (defeasible deontic logic) rules* by using the recognized entities. If a translation is available for the other legal texts that are referenced in the current document, then also the corresponding rules are included in the framework.

The above specified pipeline is defined in Figure 4.2, and explodes the details of the node **Actual Translation** that appears in Figure 4.1.

In order to identify the relevant *verbs*, *nouns* or *noun phrases* in the text, a *preprocessing step* may be necessary. We will describe a way to do this automatically but at this stage of the research we have automated only the detection, while the list of terms to be used in that phase is obtained manually.

The technical target of the translation process is the formal language of *Defeasible Deontic Logic* [108], that has been described already in Section 2.1.2. This formalism can then be targeted to LegalRuleML [6], a markup language that enjoys the same expressiveness. Section 4.1.1 describes the architecture and specifies the functionalities of the well-known GATE system[2], that is employed in this study as a framework for the development of the solution in a prototype.

As in [51], we assume that norms are represented in defeasible deontic logic by the Definition that follows:

Definition 1 (Norm). A norm n is a finite set of rules in defeasible deontic logic, where each rule is either a definition $l_1,\ldots,l_n \rightarrow l$, that means a strict rule, a fact l, an unconditional rule with a modal, $\mathscr{M}\ l$, or a conditional rule $l_1,\ldots,l_n \Rightarrow\ \mathscr{M}\ l$, where $l, l_1,\ldots,l_n$, with $n \geq 0$, are propositional literals representing states, actions, or events (asserted to occur or negated as not occurring). $\mathscr{M}$ is a deontic operator indicating an obligation $\mathscr{O}$, a prohibition $\mathscr{F}$, or a negation of one of them.

4.1.1 Text Processing

GATE is an open-source infrastructure that can be used to develop natural language processing (NLP) software components. It can be used in an interactive manner and it allows

[2] www.gate.ac.uk

to extract information by writing rules that make use of syntactic analyses. We built a custom pipeline with the following language resources:

– *English Tokeniser*, that splits the text into tokens;
– *Gazetteer*, that annotates terms in the text;
– *English Sentence Splitter*, based on punctuation;
– *POS Tagger*, that assigns POS (Part of Speech) tags to tokens;
– *JAPE Transducer*, that attempts to find unknown Named Entities based on extraction templates written in the JAPE language.

A *gazetteer list* is a pre-made lookup list adopted to annotate terms in the text and perform Named Entity Recognition. More complex formulae built with these terms can be detected by using JAPE (Java Annotation Pattern Engine), that allows to specify regular expressions that make use of those simpler annotations. For instance, it can be used to annotate as an NE of type Person lookups of type "title" followed by a "firstname" and "lastname" or to annotate as an NE of type Organization lookups with an NNP (proper noun) POS tag followed by an annotation for Company suffixes ("Ltd" or "GmBH").

In order to perform Named Entity Recognition, a list of relevant words or locutions have to be fed to the gazetteer resource. Performing this step in an automated step could be potentially disruptive, as shall be clear in the application of the pipeline to one sample document in Section 4.2, and therefore we are planning a new experiment with this specific purpose.

In a corpus of documents the *corpus frequency (cf)* of a term t is the ratio between the number of occurrences of t and the sum of the lengths of documents in the corpus, whilst the *document frequency (df)* of t is the ratio between the number of documents in which t occurs and the number of documents in the corpus. For a single document, the *text frequency (tf)* is the corpus frequency limited to the single document. The index *tf-idf* is the product between tf and the logarithm of the inverse of df in a corpus. Therefore, when a term has a high index tf-idf for a specific document, it is relevant to that document, and this measure is used in Information Retrieval for detecting the possible keywords of a given document.

When a term has an average high tf-idf on a portion of a corpus we can consider that term *significant* in the corpus. A significant term, usually, needs to be processed when important parts of the meaning of a sentence are to be captured, but finding significant terms could also be useful to improve the computational performance of an NLP technique. In fact, when we identify *locutions* that result significant, we can stop the generation of a syntactic tree, for we have already generated the right tag on the analysis. Moreover, finding significant terms prevents misunderstanding of the sentence when lexical ambiguities arise.

Given a corpus of documents, a list of operations to automatically find such relevant terms as a preprocessing step to the methodology is as follows:

– Compute *corpus frequency*, *document frequency* and *text frequency* for terms (words and n-grams up to a given n).
– Compute the tf–idf (*term frequency–inverse document frequency*) value for each of these terms.
– Perform a cut-off of terms with "high" corpus frequency or document frequency.
– Consider the terms with a "high" tf–idf.

The terms identified in this manner can also be used to segment the text in different portions.

The above technique has been tested for a limited number of analogous documents of the mentioned corpus of Resource Contracts and exhibited a significant number of potential elements of the gazetteer. In particular, when launched with thirty documents among more than 2500, all of the same type (exploration permits) and in English, it extracted by itself 12000 terms that resulted correctly devised as being not stop-words. Once actually analysed for tf-idf high values on the percentile 99 we get roughly 500 terms, that is a promising starting point for a classifier that aims at devising elements for syntactic analysis. The processing analysis has been performed by OpenNLP with one, two and three-gram analyses.

In this Chapter we process the document partly by hand, in particular by mimicking the detection of the significant terms through common sense of the human tagging.

However, limited to the processing of one single document, that is obviously not an adequate experimental corpus, we can notice that the application of the automation process as devised above identifies all the significant terms we tagged by hand, and found, naturally, a large number of terms that we overlooked while performing the process without following a pre-defined algorithm. This means that, although the automated process may result erroneous in places, if this trend will be confirmed in the second phase of the experimental trial, the cutoff that we shall obtain in the syntactic tree generation will cause a measurable improvement in the computational performance.

4.2 Case Study: an Exploration Permit

In this section we apply the methodology described in the previous Chapter to a real-world resource contract.

Table 4.1: Named Entities Recognized in the Text

Name	Entities
Act	Petroleum Act, 1967; Aboriginal Heritage Act, 1972
Location	State of Western Australia; San Francisco, California, USA; Tulsa, Oaklahoma, USA
Organization	GEOPETRO COMPANY; SEVEN SEAS PETROLEUM AUSTRALIA INC.; AMITY OIL NL
Person	NORMAN MOORE; Minister for Mines; WILLIAM LEE TINAPPLE; Director Petroleum Operations Division; inspector; a person; the permittee
Date	JULY 2, 1997; June 1998; November 1998

We tried our approach on contracts from the *resourcecontracts.org* website, that contains thousands of petroleum and mining contracts, and chose to present the analysis of one specific contract[3] with the features that we are interested in, but that also maintains a small size, so that it is readable and understandable by humans. The contract concerns the grant of an exploration permit for petroleum in the State of Western Australia to three companies. The analysis is carried out on the .doc version of the document, containing 1264 words.

We apply the pipeline to the document as follows. First, we apply the preprocessing step, where we detect the relevant words or locutions in the text. Then, we execute the first step of the pipeline to classify the document in order to identify its type, language and normative background. In our case, the type is inferred by the title, the English language is contained in the metadata of the repository, and the normative background is also inferred from the title, where "State of Western Australia" appears. Moreover, there is a reference to the Petroleum Act (1967) in the first page and a reference to the Aboriginal Heritage Act (1972) in the last page; these references are found later by a JAPE rule.

Subsequently, we apply named-entity recognition (NER) to extract named entities that are mentioned in the text. Table 4.1 summarizes the named entities that can be easily found by a combination of gazetteer lists and pre-defined JAPE rules.

```
(({obligation})
({Token.category==VB})
({Token.category!=NN})*
({Token.category==NN})
):ObligationRelation1
```

```
(({permission})
({Token.category==VB})
({Token.category!=NN})*
({Token.category==NN})
):PermissionRelation1
```

```
(({prohibition})
({Token.category==VB})
({Token.category!=NN})*
({Token.category==NN})
):ProhibitionRelation1
```

```
(({exception})
({Token.category==VB})
):ExceptionRelation1
```

Figure 4.3: JAPE rules to detect obligations, permissions, prohibitions and exceptions

Gazetteer lists contain pre-defined terms that we are interested to detect in the text, while more complex JAPE rules are used to detect the combination of a prescription with a verb and an object. We give more details about the process while focusing on "Schedule 2" of the document, that is the section that contains prescriptive elements. In Figure 4.3 we show the LHS of the JAPE rules that are adopted to detect obligations, permissions, prohibitions and exceptions: these rules detect patterns containing terms that appear in gazetteer lists (i.e., `obligation`, `permission`, `prohibition`, and `exception`) and terms tagged as specific parts of speech (such as verbs `VB` or nouns `NN`), and tag them as relations (i.e., `ObligationRelation1`, `PermissionRelation1`, `ProhibitionRelation1`, and `ExceptionRelation1`).

Deontic modalities are identified by the tokens **shall** and **must** for obligations (translated to the operator $\mathscr{O}$); the expression **may** for permissions (translated to the operator $\neg\mathscr{F}$); **shall not** for prohibitions (translated to the operator $\mathscr{F}$); and **except** for exceptions (translated

[3] The contract that we analyse can be found at https://www.resourcecontracts.org/contract/ocds-591adf-5231394526/view.

to a precedence > between rules). On the other hand, named entities that are recognized as Locations, Organizations, Persons, and Dates will be translated to constant terms in DDL.

Entity Name	DDL
Act	include rules from other documents
Location	constant term in literal
Organization	constant term in literal
Person	constant term in literal
Date	constant term in literal

Table 4.2: Correspondence between Named Entities and DDL

Modality	Text	Operator in DDL
Obligation	shall; must	$\mathscr{O}$
Permission	may	$\neg\mathscr{F}$
Prohibition	shall not	$\mathscr{F}$
Exception	except	precedence > between rules

Table 4.3: Correspondence between Modalities and DDL

Figure 4.4: JAPE rules to detect other syntactical structures

```
(
({Token.kind==word, Token.orth==upperInitial})
({Token.string=="Act"})
({Token.kind==punctuation})
({Token.kind==number, Token.length==4})
):ReferenceAct


(
({Token.string=="grant"})
({Token.category==TO})
({Token.kind==word, Token.orth==allCaps})*:companyName1
({Token.string!=";"})*
({Token.string==";"})
({Token.kind==word, Token.orth==allCaps})*:companyName2
({Token.string!=";"})*
({Token.string==";"})
({Token.kind==word, Token.orth==allCaps})*:companyName3
):grantRelation
```

Some of the expressions that appear in the text deserve specific attention as they can be classified as specific *legal terms*, and therefore can be easily bound with specific gazetteers, which is more convenient than to apply general rules. The number of expressions that need to be captured in this manner is rather low in the practice of legal text processing, and these special cases are usually just verbs or verbal locutions. Once a specific subdomain has been engineered in terms of document corpus for this goal, the objects inserted in the gazetteers shall be limited to a controlled number, and therefore manageable in practice.

In the specific case of the exploration permit that we applied the techniques to, we have identified on purpose some cases that need this special treatment. The special terms we encountered are listed below:

1. in accordance with the approval
2. production testing; measuring
3. permit ...to test
4. authorised in writing for the purpose
5. take adequate measures
6. comply

The expression *in accordance with the approval* is treated as special because the structure of the corresponding noun phrase is inserted in a specific context, that of providing an exception to another rule, that occurs in many legal texts for the specific noun *approval*. The expressions *production testing* and *measuring* are in the gerund form, that is somewhat uncommon in many other cases but can appear in legal texts. A good approach to solve this is purely *syntactic* as the majority of verbs form regular gerunds. The expression *permit...to test* contains an expression (permit) that could be translated into a modal, but for the purpose of this investigation we limited ourselves to modals whose scope is on a propositional structure.

The expression *authorised in writing for the purpose* is a relative clause, and we take *authorised* as the corresponding past passive expression. Again, we detect this by having it in the gazetteer list. The idiosyncratic expression *take adequate measures* is common in technical language related to legal texts. The expression *comply* could be considered potentially redundant in terms of application. The full expression contains a basic obligation: the permittee has to comply with the requests of the Minister for Mines. This is in some sense intrinsic in the definition of an exploration permit for natural resources, but it is also asserted in the referenced Petroleum Act for what concerns exploration permits.

Clause	Exception
(1.2)	$r1.2\#2 > r1.2\#1$
(2)	$r2\#2 > r2\#1$
(4)	$r4\#2 > r4\#1$
(5)	$r5\#2 > r5\#1$

Table 4.4: Exceptions between rules

In Table 4.7 and 4.8 we enumerated the rules by an automated mechanism that generates the rule labels by the enumeration found in the text. When a single piece of text is translated into more than one single rule, the enumeration builds a # pattern at the end of the automatically generated string and follows it with a counter that starts from 1. The superiority relation $>$ is established between the rules as a direct effect of the exception operators found in the text and is represented in Table 4.4.

In Figure 4.4 we show the LHS of the JAPE rules that are adopted to detect other syntactical structures such as the acts that are referenced in the text and the names of the companies to which the exploration permit is granted. These last rules are an example of rules that have to be modelled after studying the specific structure of the analysed document: we use the information that acts are always referenced with their issue year and that the exploration permit is granted to three companies, the names of which appear in capital letters. Each of these companies is a *permittee* in the language of the legal text and the corresponding detected structure is to be translated to a definition in DDL.

Finally, the recognized entities are used to create DDL (defeasible deontic logic) rules. The JAPE rules allow to tag complex expressions, but this step is still performed mainly manually in the majority of applications. We have developed this part of the pipeline by adding elements by hand and processing them in an automated way. The process can be represented by the following schemas for the JAPE rules in Figure 4.3:

- In the case of *ObligationRelation1*, *PermissionRelation1* or *ProhibitionRelation1*, represented by a tuple
$$< M, V, NE_1, \ldots, NE_n >$$
where M is a modality as in Table 4.3, V is a term of type predicate as in Table 4.6 and NE_i $(1 \leq i \leq n)$ is a term of type constant as in Table 4.5, we translate to a rule $\Rightarrow \mathscr{M}\ p(ne_1, \ldots, ne_n)$, where $\mathscr{M}$ is the modal operator corresponding to M, p is the predicate corresponding to V and ne_i is the constant corresponding to NE_i;
- In the case of *ExceptionRelation1*, represented by a tuple
$$< E, Q >$$
where E is an exception as in Table 4.3 and Q is a term of type predicate as in Table 4.6, that follows a tuple
$$< M, V, NE_1, \ldots, NE_n >$$
, defined and translated as above, we translate to a rule $q(ne_1, \ldots, ne_n) \Rightarrow \sim \mathscr{M}\ p(ne_1, \ldots, ne_n)$, where q is the predicate corresponding to Q and this rule has precedence over the previous rule.

Tables 4.2,4.3, 4.5 and 4.6 illustrate the correspondence between parts of the text in the document and the logical formulae. Tables 4.7 and 4.8 show the final translation of “Schedule 2” of the document in DDL allowing to compare original text and DDL formulae. At this step also the logical rules of the two referenced documents Petroleum Act (1967) and Aboriginal Heritage Act (1972) are included in the logical theory.

Tagged Term	DDL
the permittee	*permittee*
the work... that year	*current_work*
the work... of a subsequent year	*subsequent_work*
work in addition to the work	*additional_work*
any works... operations	*work*
the Minister for Mines	*minister*
petroleum	*petroleum*
full particulars	*particulars*
a person	*person*
device	*device*
equipment	*equipment*
well	*well*
environment	*environment*

Table 4.5: Correspondence between Tagged Terms and DDL - constants

Tagged Term	DDL
carry out	*carry_out*
commence	*commence*
in accordance with the approval	*approval*
recover	*recover*
production testing	*testing*
pay	*pay*
furnish	*furnish*
permit ... to test	*permit_test*
authorised in writing for the purpose	*authorised*
measuring	*measuring*
construct any installation or install	*install*
abandon, suspend or complete	*abandon*
take adequate measures	*take_measures*
comply with all Directions	*comply*

Table 4.6: Correspondence between Tagged Terms and DDL - predicates

Table 4.7: Translation to DDL: constants, modalities, predicates, exceptions.

Original text and corresponding DDL code
1.1 … the permittee: (1.1a) shall carry out in or in relation to the permit area, to a standard acceptable to the Minister for Mines, the work specified in the minimum work requirements set out opposite that year in the second column of the table;
■ $r1.1a: \Rightarrow \mathscr{O}\ carry\_out(permittee, current\_work)$
(1.1b) may carry out in or in relation to the permit area, to a standard acceptable to the Minister for Mines, all or part of the work specified in the minimum work requirements of a subsequent year or years of that term set out opposite that year or those years in the second column of the table; and
■ $r1.1b: \Rightarrow \neg\mathscr{F}\ carry\_out(permittee, subsequent\_work)$
(1.1c) may carry out in or in relation to the permit area, to a standard acceptable to the Minister for Mines, work in addition to the work specified in the minimum work requirements set out opposite that year and in the subsequent year or years, if any, of that term in the second column of the table
■ $r1.1c: \Rightarrow \neg\mathscr{F}\ carry\_out(permittee, additional\_work)$
1.2 The permittee shall not commence any works or petroleum exploration operations in the permit area except with, and in accordance with the approval in writing of the Minister for Mines or of a person authorised by the Minister for Mines to give that approval.
■ $r1.2\#1: \Rightarrow \mathscr{F}\ commence(permittee, work)$ ■ $r1.2\#2: approval(minister, permittee, work)$ $\Rightarrow \neg\mathscr{F}\ commence(permittee, work)$
2 The permittee shall not recover any petroleum from the permit area except as a result of production testing of a well.
■ $r2\#1: \Rightarrow \mathscr{F}\ recover(permittee, petroleum)$ ■ $r2\#2: testing(permittee, petroleum)$ $\Rightarrow \neg\mathscr{F}\ recover(permittee, petroleum)$

Table 4.8: Translation to DDL. : constants, modalities, predicates, exceptions.

Original Text alternated to DDL DDL code
3 The permittee shall (3a) pay to the Minister for Mines, in respect of petroleum recovered by the permittee in the permit area, royalty at the rate that is for the time being the prescribed rate in respect of that petroleum
■ $r3a : \Rightarrow \mathscr{O}\ pay(permittee,\ minister,\ petroleum)$
(3b) in respect of each royalty period, furnish to the Minister for Mines, in such form as the Minister for Mines may from time to time require, full particulars of the quantity of petroleum recovered by the permittee and full particulars of matters relevant to ascertaining the value at the well-head of that petroleum; and
■ $r3b : \Rightarrow \mathscr{O}\ furnish(permittee,\ minister,\ petroleum,\ particulars)$
(3c) permit a person authorised in writing for the purpose, by the Minister for Mines, or an inspector, to test or examine any measuring device installed that has been, is being or is to be used by the permittee to measure the quantity of any petroleum recovered in the permit area.
■ $r3c :\ authorised(person),\ measuring(device,\ petroleum)$ $\Rightarrow \mathscr{O}\ permit\_test(permittee,\ person,\ device)$
4 The permittee shall not construct any installation or install any equipment in the permit area except with and in accordance with the approval in writing of the Minister for Mines or a person authorised in writing by the Minister for Mines to give that approval.
■ $r4\#1 : \Rightarrow \mathscr{F}\ install(permittee,\ equipment$ ■ $r4\#2 :\ approval(minister,\ permittee,\ equipment)$ $\Rightarrow \neg\mathscr{F}\ install(permittee,\ equipment)$
5 The permittee shall not abandon, suspend or complete any well except with and in accordance with the approval of the Minister for Mines or of a person authorised by the Minister for Mines to give that approval.
■ $r5\#1 : \Rightarrow \mathscr{F}\ abandon(permittee,\ well)$ ■ $r5\#2 :\ approval(minister,\ permittee,\ well)$ $\Rightarrow \neg\mathscr{F}\ abandon(permittee,\ well)$

Chapter 5

Reasoning on a Legal Text: the Case of Adjectives

If we want to convert a natural language text into logical formulae, we also have to deal with *adjectives*, which are words that are used to modify or describe nouns or pronouns. Oftentimes, adjectives carry a lot of meaning and are used to qualify the subjects of the discourse. In short, adjectives can be used as a way to understand a text. Indeed, it is possible to guess the *polarity* of a piece of text (i.e., whether the text has positive or negative sentiment) just by considering the adjectives that appear in it [126]. Therefore, when we formalize a text in natural language, we need to have mechanisms that capture the interactions between the adjectives that are applied to the different objects in the text. To this purpose, in this Chapter we give a logical system, that we name *Adjective Logic* $\mathscr{A}$-Log, that is specialized to represent adjectives and can be used to derive consequences with them. The system is given by describing its syntax, semantics, proof system, proof theory and algorithms.

Adjectives can be used with nouns to form noun phrases, to *modify* nouns, or as *judgments* expressed in nominal sentences. For instance, the sentence "*I ate a tasty ice cream*" contains a judgment, subjectively expressed by the speaker, regarding a specific quality of the ice cream, that is modified by the adjective. In this sense, we state that the speaker has intended to describe a proper subset of the ice creams, the tasty ice creams. Moreover, we derive the nominal sentence "*The ice cream is tasty*" as a consequence of the sentence used above. Sometimes it happens that adjectives are used *in place of nouns* (substantivized adjectives) as in "the French" or "the unknown". In this Chapter we shall *only* consider adjectives as the specific part of speech that is attached to a noun in a noun phrase or used as a nominal predicate in a nominal sentence.

There are certain consequences that we derive from sentences where adjectives are employed. Consider, in particular, the sentence "*I ate a tasty ice cream*". We ensure, in particular, that the ice cream *is not bland*, in the opinion of the speaker, but on the contrary, we cannot ensure that other speakers *agree* on the judgment, possibly thinking that the very same ice cream is bland. Someone may also disagree on both the opinions, considering that specific ice cream neither tasty nor bland.

Adjectives are ubiquitous and often decisive in providing meaning. Consider the text given in Box 2, that is taken from the first premise of the Universal Declaration of Human Rights.

Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world [ctd...]

Box 2: Universal Declaration of Human Rights: first premise.

One may be tempted to assume that legal texts do not contain adjectives or that adjectives might be cancelled without loosing a significant part of the meaning. This assumption is derived by the idea that adjectives are used to introduce *refinements* to the nouns that they are attached to, and therefore, in some texts in particular, they might be omitted. In Box 3 we provide the residual part of the text after removal of descriptive adjectives. We deleted one "and" conjunction in the first sentence to preserve grammatical correctness.

Whereas recognition of the dignity and of the rights of all members of the **family** is the foundation of freedom, justice and peace in the world [ctd...]

Box 3: Universal Declaration of Human Rights: first premise without adjectives.

Clearly, the sentence has very little significance if the adjectives *inherent*, *equal* and *inalienable* are removed, but the worst effect is determined by the elimination of the adjective *human* that suppresses any intended meaning at all. Grammatically, the phrase *the family* may exhibit both an implicit *anaphoric* meaning in this context, intended to refer to something that has been considered before (and to which the reader naturally refers to), and an *abstract* meaning. In this case, the only possible reference is *abstract* for the text is in its incipit. Thus, it would be intended to refer to the general meaning of the term family, that is not what the original text referred to.

We now go into the analysis of a review in a recommender system, presented in Box 4[1]. Reviews in recommender systems are supposed to contain judgments, and therefore several adjectives, that are a common way to express judgments in natural language. Note that some of the adjectives, when removed, produce either a distortion of meaning or a loss of significance of the sentences per se. In particular, the parentheses containing the judgment "location" would be insignificant, as well as the judgments about the owner and the staff. The same exercise we made for the legal text above is carried out here, and the result is in Box 5. Positive orientation of the judgment is derivable from the second version of the text as well, but details of the meaning are lost.

As we have seen with these two potentially dissimilar situations, adjectives may be relevant in language usages that are completely different. Legal language is one of the most *restricted* languages that is commonly used, where legal terms are considered to mean something that is set, typically invariable over time and among speakers. Reviews, on the other hand, are written with common lexicon, and represent one of the most individually variable parts of language, containing opinions and individual variants of the meaning

[1]Actual references are here omitted. In the example, we underlined the descriptive adjectives. We preserved the original text that contains also some grammar anomalies, including slang terms and emoticons, that we left for the sake of truthfulness.

> Located just near the fort in city (perfect location), the hotel is clean and provides you great hospitality. The hotel owner is very kind and will guide you throughout your stay. The staffs were very helpful. The hotel provides you almost every facilities from good rooms to desert safari, jeep safari, camel safari and tent stay. The rooftop cafeteria of hotel was great. When I say food was great (I mean it :)). Overall experience was awesome. I'll recommend the hotel to every couple and families:).

Box 4: A review of a hotel.

> The hotel is located just near the fort in city. The hotel owner will guide you throughout your stay. The hotel provides you almost every facilities from rooms to desert safari, jeep safari, camel safari and tent stay. The hotel has a rooftop cafeteria. I'll recommend the hotel to every couple and families:).

Box 5: A review of a hotel without descriptive adjectives.

of terms. Although there is such a strong difference, in both pieces of text the usage of adjectives exhibits a strong relevance.

When treating natural language, therefore, it is necessary to deal with the representation of adjectives and the processing of them by a logical engine, in order to derive the logical consequences of assertions, that are produced by the analysis of the natural language text. The first brick in this process is the treatment of noun phrases containing adjectives, that, as we shall see in Section 5.1, have consequences in form of *simple nominal sentences* when the adjectives are intersective. We deal with standard computational problems in a formal logic system with its own inference rules, able to deal with this type of sentences. We introduce a syntax for the *Adjective Logic* along with a semantics for truth values and functional meanings of formulae, a *proof system* that we show to be sound and complete, and a set of algorithms used to manage derivation of formulae and consistency checking for a finite set of formulae. We also study the computational complexity of the problems, as related to the proposed solutions.

A few investigations that in the past looked at the development of mechanisation of reasoning with natural language have included adjectives in the process, in particular the studies on *Natural Logic*, that is defined as a logic framework to reason on the logical relations among sentences in natural languages. In particular [84] dealt with the basic notion of Natural Logic, and provided a semantics that accommodates adjectives. Further on, [105] studied syllogistic logic as a tool for Natural Logic that is to be also used to reason with adjectives.

The rest of the Chapter is organized as follows: Section 5.1 presents a discussion of the theme and introduces the goals of the study, Section 5.2 introduces the basic definitions needed for $\mathscr{A}$-Log, the *Adjective Logic*, in terms of syntax and semantics. Section 5.3 describes the proof system of $\mathscr{A}$-Log and Section 5.4 proves the basic properties of this system: soundness and completeness. In Section 5.5 we introduce a set of algorithms that are used to solve the consistency checking problem for theories in $\mathscr{A}$-Log.

5.1 Discussion

Adjectives are a specific and relevant part of speech in natural language. They belong to several distinct categories, such as *Descriptive* Adjectives, also known as Adjectives of quality (e.g., big, large, strict, poor). All non-descriptive adjectives are denominated *Definitive*, and include *Quantitative* or Adjectives of quantity (e.g., enough, all, whole); *Numeric* or Adjectives of number (e.g., one, two, three); *Demonstrative* Adjectives (e.g., this, that); *Distributive* Adjectives (e.g., every, both); *Possessive* Adjectives (e.g., mine, her); or *Interrogative* Adjectives (e.g., what, whose).

In this Chapter we focus upon **descriptive** adjectives, that are used to express the qualities of nouns. As shown in linguistics, and in particular in Klein's foundational work on the semantics of descriptive adjectives ([90]), that is in turn based on the theoretical investigation by [87], and further analysed in numerous investigations, in particular those of Sassoon in the field of semantics ([120, 121, 117]), the basic notion needed to represent adjectives is *vagueness*, the property of these terms to denote something that is neither true nor false. A large number of adjectives have this intrinsic property that distinguishes them from other categorical parts of speech, apart from adverbs, that often enjoy this property as well.

Consider for instance the adjective *pretty*. When we consider the admissible judgments on a particular object, for instance *a pretty house*, the expression of some basic judgments are:

- The house is pretty;
- The house is awful;
- The house is neither pretty nor awful.

Moreover, it is rather natural to consider that *not pretty* is implied by *awful*. The third judgment above represents a behaviour that cannot be accounted for crisp terms. For instance, being a dog is either true or false, and an assertion such as *that animal is neither a dog nor a non-dog* does not make sense. Bennett et al. ([18, 19]) have discussed aspects regarding vagueness with the explicit focus on vague geographic terms, including nouns that result to have a vague semantics (e.g., *forest*). In this study we do not approach these façades of vague nouns that are very diverse and different from the nature of adjectives, leaving judgment expression based on vague nouns to further studies.

Reasoning with adjectives is the ability of obtaining the above derivations in a formal way. Klein's linguistic account for adjectives is the basis for the development of such a theoretical framework.

We restrict ourselves onto the *gradable intersective adjectives*: those that either on a nominal sentence, or in force of a rewriting that is legitimate by the nature of the considered token, generate a nominal sentence when attributed to a noun. We also consider analogously those locutions that mimic these adjectives. In particular, this holds for provenance locutions (*Frank is from England*, that we may translate into *John is English*), genitive expressions (*Mom's cat* that translates into *The cat is mom's*), and other similar ones.

We focus on the investigation on which Sassoon semantics is applied to: the mechanics of derivation for *intersective adjectives*, with the notion of *subjective adjective*, derived by the notion of subjective judgment by [117], and the related problems for attributions to nouns (see [119] and [118]). Subjective adjectives generalize the notion of *multidimensional adjectives*, that are, in Sassoon, a category of adjectives with more than one single measure. On the other hand, adjectives that are measured on only one single dimension are *unidimensional adjectives*. For instance, *height* is singularly measured in the adjective *tall*, that is therefore a unidimensional adjective. This does not happen for the adjective *beautiful*, that are measured in different respects.

To approach the logical structure of adjectives in languages, we chose to use *partial orders*. We adopted one partial order for each objective adjective (the order may be total for unidimensional adjectives), and one partial order for each speaker involved in the expression of a judgment for each subjective adjective. This method accommodates several different theories into the same logical framework. We do not commit to the assumption by [120] that multidimensionality is the source of subjectivity, nor to the vision that assumes subjectivity to be an effect of subjective ordering as introduced by [14]. We simply assume that gradable adjectives are related to a partial order, possibly one order for each individual expressing a judgment. Therefore, we accommodate also objective adjectives that are defined by a proper partial order, where two objects are uncomparable. An example of an objective adjective that is not unidimensional (and therefore with a partial order that is not total) is *close*: a place x may be closer than place y according to the spatial dimension, but y may be closer than x because it takes less time to get there, and therefore the two places are not comparable.

We provide here a few examples to clarify the notions we use in the rest of this Chapter, and introduce, through them, the taxonomy of the adjectives' classes that Sassoon uses, for the sole purpose of accurately framing the investigation. In particular, a very general distinction lies on the ability of an adjective to *modify* the noun it is referred to. Adjectives that change by refining, shifting, generalising and other possible operations on the nouns are named *descriptive*, as we stated above, and those adjectives that change the *reference* of the noun are named *definitive* (possessive, numeral, and so for). For instance, a definitive adjective is *this*, while a descriptive adjective is *genetic*.

Among descriptive adjectives we distinguish those that are subsective, and describe a subset of the set represented by the noun, and those that are not. An example of the first group is the adjective *semantic*, that when referred to a noun such as *rule* describes the subset of the rules that have a semantic nature. Subsective adjectives attribute the nouns to subsets, and every object obtained with such an adjective is necessarily of the type of the noun. For instance *an Italian researcher* is a researcher. Non-subsective adjectives, instead, provoke a shift in meaning. For instance, *fake gun* generates a class of objects that are not guns, whilst, as a further example, *alleged hairdresser* generates a class that contains both objects that are hairdressers and objects that are not.

Subsective adjectives are to be divided by three different criteria: polarised vs. unpolarised, intersective vs. non-intersective, and gradable vs. ungradable. The distinction between *polarised* and *unpolarised* adjectives lies upon the existence of the *antonym*. Examples of unpolarised adjectives are colour adjectives, such as *yellow*: there is no adjective that represents the antonym of yellow. However, unpolarised adjectives are often also *gradable*, and for instance we state that an object is *more yellow* than another one[2].

Subsective adjectives are distinguished in *intersective* and *non-intersective*, where the second group is exemplified by terms such as *clever*, or *smart*. Non-intersective adjectives, though establishing a subset of the noun, do not allow derivation (in the sense of Sassoon) of the corresponding nominal sentence. For instance, *John is a clever surgeon and a reluctant traveller* does not allow to derive the single assertions *John is clever* and *John is reluctant* alone, for this derivation would permit to conclude assertions such as *John is a reluctant surgeon* or *John is a clever traveller* that are clearly not consequences of the previous assertions. Among subsective adjectives, we finally distinguish *gradable* and *ungradable* adjectives, many of the ungradable ones being obtained as past participle, for instance *married* (with numerous exceptions, such as *sunken*, which is a past participle but is gradable), but also in other ways (those qualifying belonging to an ethnic or national group, like *Canadian*).

Furthermore, gradable adjectives distinguish in *subjective* and *objective*, where objective adjectives allow to generalize conclusions regarding comparatives. For instance, someone who claims "*Andrew is taller than Adam*" is in conflict with someone asserting "*Adam is taller than Andrew*". This does not happen for subjective adjectives such as *elegant*: someone states "*Andrew is more elegant than Adam*" without being in conflict with someone asserting that "*Adam is more elegant than Andrew*". Specifically, the two speakers *disagree* but the assertion of their judgments does not generate a contradiction.

There is an open debate about the previous groups and the relationship with subjective and objective adjectives that has driven to possible cases of special interest arisen by Sassoon [117]. The hierarchy of adjectives' classes is presented in Figure 5.1.

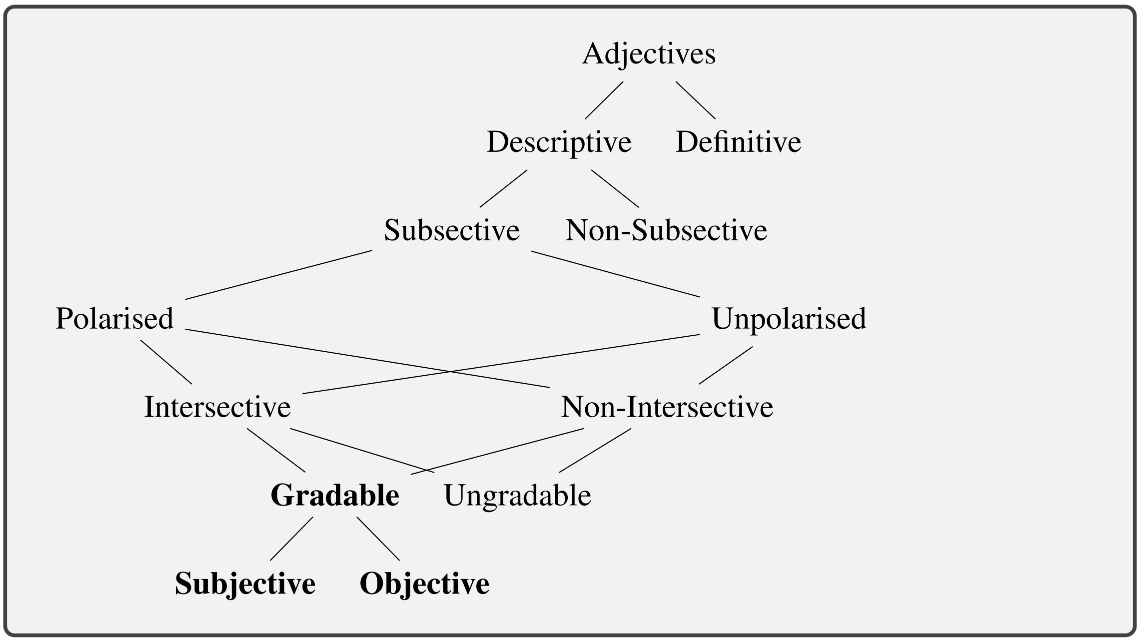

Figure 5.1: The hierarchy of adjectives' classes adopted as reference by [120]. In boldface the adjectives treated in this study.

[2]The discussion about qualities and nouns in relation with colours, but also with tastes such as *sweet* or *salted*, is considered by [106], and more in general by [62].

1. John is tall	A John is not tall
2. John is taller than Frank	B Frank is taller than John
3. Beggy is a large guinea pig	C Beggy is a tiny animal
4. Lat is a tiny elephant	D Lat is quite big
5. Barbara is beautiful	E Barbara is not very beautiful
6. Loraine is very intelligent	F Loraine is intelligent
7. Mark is more brilliant than Joe	G Mark is as brilliant as Joe
8. Jill's doily is nicer than Joe's song	H I can't say whether Jill's doily is nicer than Joe's song

Table 5.1: Example of nominal sentences in English.

Example 1. *Sentences with typical usages of adjectives.*
Consider the nominal sentences in Table 5.1, where we assume that the sentences in the left column are believed by an individual named Alice, and those on the right are believed by an individual named Bob.

The meta-judgments that we may express on these sentences are numerous. In particular, in sentences numbered 1 and A we note that the disagreement between Alice and Bob is explainable by assuming that Alice has the belief that being tall means being taller than a given threshold τ_1, for instance 1.76 m, while Bob has the belief that being tall means being taller than a threshold τ_2 that is instead 1.80 m, and John's height is 1.78 m.

Regarding sentences 2 and B, the disagreement is not explainable. One of the two is judging erroneously, for John being taller than Frank or vice versa is an objective matter. Therefore, either one is wrong, or both (in the case in which John and Frank have the same height).

When looking at sentence 3 and its relationship with sentence C we may have a different explanation for the (apparent) disagreement between Alice and Bob. The judgments can be reduced one to the other based on the fact that the category "animal" is a very general one, and guinea pigs are relatively small as animals. Tiny is a strong version of small, and therefore the judgment of Bob expresses a fact that is possibly true for Alice as well, because Beggy is a rather robust exemplar among other guinea pigs.

Analogously for Lat's measure (sentences 4 and D), but in this case we do not refer to a category and a subcategory (arguably the judgment on Lat as *quite big* refers to the *largest natural attribution* of a category, and it is clearly a correct judgment, but does not get account to the fact that Lat is much smaller than, for instance, the Everest).

The disagreement on sentences 5 and E lies on the arbitrariness of the judgments expressed with the adjective *beautiful*. It is obviously possible that the two judgments result compatible, as Bob may judge Barbara as either beautiful (but not very beautiful), awful, or neither beautiful nor awful. Even if the two judgments were opposite, there would be no contradiction in the sentence sets, as Alice and Bob may have different beliefs on what is beautiful and what is not.

Sentences 6 and F do not provide any disagreements, for Alice's opinion is that Loraine is intelligent (as a consequence of being very intelligent), and this is also the opinion of Bob.

Also the disagreement expressed in the sentences 7 and G is explainable, based on the fact that the adjective *brilliant* cannot be judged unanimously.

Judgments expressed in sentences 8 and H are different, but again this disagreement is explainable because the adjective *nice* is subjective. The first is the expression of a direct positive ordering of two artworks of different nature, whilst the second judgment is expressed in form of *nec iudicatur*. In this thesis we do not deal with the issues of nec iudicatur that emerge by expressing judgments on objects that belong to distinct classes, as in this case.

In the remainder of this Chapter, we shall deal with the problem of formalising judgments of the same type of those above by means of a propositional system. The study is a first step on adjectives that are *gradable* and on the assertions made with them. We do not consider here several other aspects, for instance those related to reference to specific categories and relationships among adjectives acting on the same measurements. Therefore, the study of the consistency of expressions such as 3 and C or 4 and D is left to further investigations.

5.2 Definitions

We adopt an extension of the formal language of first-order logic, restricted to *ground formulae*, that is, formulae in which no variables nor quantifiers appear, that constitutes a propositional fragment.

In Section 5.2.1 we define well-formed formulae, while in Section 5.2.2 we introduce interpretation and semantic evaluation of formulae.

5.2.1 Syntax

We define the Adjective Logic $\mathscr{A}$-Log as composed of a language $\mathscr{L}$ and a set of rules $\mathscr{R}$, that is, $\mathscr{A}$-Log $= \langle \mathscr{L}, \mathscr{R} \rangle$, where the language $\mathscr{L}$ is specified as follows. The signature Σ of $\mathscr{L}$ is

$$\Sigma = Adj \cup C \cup \Lambda \cup L \cup \{O,\ U,\ *,\ ^\circ,\ +,\ \sim,\ (,\ ),\ :\},$$

with $l, m, n > 0$, where:

- $Adj = \{p_1, \ldots, p_n\}$ are the adjective symbols;
- $C = \{a_1, \ldots, a_m\}$ are the object constant symbols (that we also name *nouns*);
- $\Lambda = \{\lambda_1, \ldots, \lambda_l, \varepsilon\}$ are the agent symbols;
- $L = \{\wedge, \neg, \bot\}$ are the logical connective symbols;
- O is the objective adjective symbol;
- U is the unpolarised adjective symbol;
- $*$ is the degree symbol;
- $^\circ$ is the dualisation symbol;
- $+$ is the majority comparative symbol;
- $\sim$ is the equality comparative symbol;
- $(,), :$ are the auxiliary symbols.

The adjective predicates are obtained by using the adjective symbols and some of the operators introduced above. We employ the classical syntactic notion of comparative of equality and comparative of majority (and do not consider the minority comparatives: they are not necessary because of the symmetry with the majority comparatives). On the other hand, in this step of the investigation we are not interested in devising the notion of *superlative*, that needs to be treated apart because superlatives include a reference to a set of elements (e.g., *the best actor* is referred in a context, such as the best in the world, or the best in a specific movie).

Since subjective adjectives are more frequent than objective ones, the default in $\mathscr{A}$-Log is that an adjective is subjective, while it is objective only when this is asserted with the symbol O. In the same manner, polarised adjectives are more frequent than unpolarised ones, so that the default in $\mathscr{A}$-Log is that an adjective is polarised, while it is unpolarised only when this is asserted with the symbol U. Adjectives are also to be modified, as exemplified above, by using *intensifiers* and *mitigators*. Intensifiers include *very, really, extremely*, whilst mitigators include *somewhat, rather, pretty, to some extent*. We commit ourselves, in this first step of the investigation, to a language with *only one intensifier*. We also assume that all adjectives are modified by this intensifier, that is therefore generic.

The majority comparative adjective predicates $\{p_1^+,\dots,p_n^+\}$, equality comparative adjective predicates $\{p_1^\sim,\dots,p_n^\sim\}$, and *intensified* adjective predicates $\{p_1^*,\dots,p_n^*\}$ are built by combining the adjective symbols with the majority comparative modifier $+$, the equality comparative modifier $\sim$, and the degree modifier $*$, respectively[3]. Moreover, the dual adjective predicates $\{\mathring{p}_1,\dots,\mathring{p}_n\}$ and the dual intensified adjective predicates $\{\mathring{p}_1^*,\dots,\mathring{p}_n^*\}$ are built by combining the adjective symbols with the dualisation modifier $^\circ$ and the degree modifier $*$. Note that we do not need sets of dual majority comparative adjective predicates $\{\mathring{p}_1^+,\dots,\mathring{p}_n^+\}$, $\{\mathring{p}_1^\sim,\dots,\mathring{p}_n^\sim\}$, because the meaning of a dual majority comparative adjective can be represented through a majority comparative adjective. Given objects a and a', the intuitive interpretation of dual majority comparative adjective $\mathring{p}^+$ is $\mathring{p}^+(a,a')$ holds if and only if $p^+(a',a)$ holds, while literals $\mathring{p}^\sim(a,a')$ and $\mathring{p}^\sim(a',a)$, obtained with the equality comparative adjectives, are both equivalent to $p^\sim(a,a')$ and to $p^\sim(a',a)$.

Definition 2 (Set of uwff). The set of unlabelled well-formed formulae Ψ is the smallest set Y with the properties:

- if $p \in Adj$ and $a \in C$, then $p(a) \in Y$, $\mathring{p}(a) \in Y$, $p^*(a) \in Y$, and $\mathring{p}^*(a) \in Y$;
- if $p \in Adj$ and $a,b \in C$, then $p^+(a,b) \in Y$ and $p^\sim(a,b) \in Y$;
- if $\phi,\psi \in Y$, then $(\phi \wedge \psi) \in Y$;
- if $\phi \in Y$, then $(\neg\phi) \in Y$.

We are now ready to introduce the syntactic category of well-formed formulae (wff) Φ.

Definition 3 (Set of wff). The set of well-formed formulae Φ is the smallest set X with the properties:

- $\bot \in X$;

[3] The approach we adopt here avoids the representation of properties by means of the *reification* operator as discussed in [80]. This process will be of interest when we shall further investigate applications to taxonomies.

– if $\lambda \in \Lambda$, $\alpha \in \Psi$, then $\lambda : \alpha \in X$;
– if $p \in Adj$, then $O(p) \in X$ and $U(p) \in X$;
– if $\phi, \psi \in X$, then $(\phi \wedge \psi) \in X$;
– if $\phi \in X$, then $(\neg\phi) \in X$.

For simplicity, when symbol λ is ε, we omit it, that is, $\varepsilon : p(a)$ is shortened in $p(a)$. Formulae not containing $\wedge$ are named *literals*. A set only formed by literals is named an *atomic set of formulae* or also an *atomic theory*[4].

The next example illustrates some formulae in the language of $\mathscr{A}$-Log by translating the sentences expressed in natural language in Example 1.

Example 2. *Expressing nominal sentences in $\mathscr{A}$-Log.*
Consider the sentences believed by Alice and Bob in Example 1. We adopt a signature Σ where

$$Adj = \{tall,\ large\_guinea\_pig,\ tiny\_elephant,\ beautiful,\\ intelligent,\ brilliant,\ nice,\ tiny\_animal,\ quite\_big\},$$

$$C = \{john,\ frank,\ beggy,\ lat,\ barbara,\ loraine,\\ mark,\ joe,\ jill\_doily,\ joe\_song\},\ and$$

$\Lambda = \{alice,\ bob,\ \varepsilon\}$. The sentences are translated in Table 5.2 (constants **joe_song** and **jill_doily** are abbreviated into *song* and *doily*, respectively, **tiny_elephant** into *teleph*, **large_guinea_pig** into *lgpig*, **quite_big** into *qbig*, and **tiny_animal** into *tan*):

1. $alice : tall(john)$	A $bob : \neg tall(john)$
2. $alice : tall^{+}(john, frank)$	B $bob : tall^{+}(frank, john)$
3. $alice : lgpig(beggy)$	C $bob : tan(beggy)$
4. $alice : teleph(lat)$	D $bob : qbig(lat)$
5. $alice : beautiful(barbara)$	E $bob : \neg beautiful^{*}(barbara)$
6. $alice : intelligent^{*}(loraine)$	F $bob : intelligent(loraine)$
7. $alice : brilliant^{+}(mark, joe)$	G $bob : brilliant^{\sim}(mark, joe)$
8. $alice : nice^{+}(doily, song)$	H $\neg bob : nice^{+}(doily, song)$ $\neg bob : nice^{+}(song, doily)$ $\neg bob : nice^{\sim}(doily, song)$

Table 5.2: Example of nominal sentences in $\mathscr{A}$-Log.

[4]With a little abuse of language we henceforth substitute the term *theory* for the term *set of formulae*. Correspondingly, we denominate *deductive closure* of a theory T the set of formulae equivalent to T that contains every formula ϕ that is derived from T (these sets are usually named theories in the mathematical logic literature). This is justified by the focus of this study on semantic aspects, more than on specific syntactic structures.

In Example 1, point H of Table 5.1, we introduced a sentence that cannot be mapped directly into a simple $\mathscr{A}$-Log formula. This sentence illustrates the case of incomparability, that has to be allowed to guarantee the representation of the cases in which adjectives are naturally associated to proper partial orders. This solution appropriately works with adjectives associated to a single measure as well (for instance *tall*).

We systematically avoid to introduce aspects of communication and language that pertain to the *epistemological* dimension of the labels, as these aspects are amply discussed as a general problem in labelled logics in several foundational works ([66, 54, 123]). Intuitively, an unlabelled formula ϕ should contain only objective adjectives, and the agent ε does not hold subjective beliefs. A set of formulae is called an apodeictic theory when this does not happen. In practice, an apodeictic theory is the assertion of a set of assertions that may be disagreed by agents, but the labels are however omitted.

Definition 4 (Apodeictic theory). A set S of wff is called an apodeictic theory if, for a given $p \in Adj$, S contains an uwff ϕ, p appears in ϕ, but $O(p) \notin S$. The formula ϕ is an apodeictic formula of S.

Definition 5 (Non-apodeictic transformation). Given an apodeictic theory S, a non-apodeictic transformation of S yields a theory S' that contains (1) the apodeictic formulae of S labelled by a fresh label μ and (2) the formulae of S that are not apodeictic.

5.2.2 Semantics

In general, the adjectives that we treat are intersective and gradable and they may be *subjective* or *objective*. Objective adjectives are a subset of the adjectives, and their complement is the set of subjective adjectives. Moreover, gradable adjectives may be *polarised* or *unpolarised*. Also unpolarised adjectives are a subset of the adjectives, and their complement is the set of polarised adjectives. Given set Adj of adjective symbols, we identify a set $Obj \subseteq Adj$ of objective adjective symbols and a set $Unp \subseteq Adj$ of unpolarised adjective symbols.

The intuition behind the adjective predicates is that $\lambda : p(a)$ holds if agent λ believes that the object represented by constant a has the property represented by adjective p; $\lambda : p^+(a, b)$ holds if λ believes that object a is more p than object b; $\lambda : p^\sim(a,b)$ holds if λ believes that object a is as p as object b; $\lambda : p^*(a)$ holds if λ believes that object a is very p; $O(p)$ holds if p is an objective adjective; $U(p)$ holds if p is an unpolarised adjective, that is, there is no dual (or antonym) $\mathring{p}$ of p. For instance, $O(heavy)$ holds because the adjective *heavy* is objective, and $U(yellow)$ holds because there is no antonym of the adjective *yellow*.

Definition 6 (Subjective partial order). For each agent $\lambda \in \Lambda$ and each adjective $p \in Adj$, the subjective partial order $\leq_{\lambda,p}$ is a partial order on the objects corresponding to symbols in C.

We specialize the definition for objective adjectives as follows:

Definition 7 (Objective partial order). For each adjective $p \in Obj$, the objective partial order $\leq_p$ is a partial order on the objects corresponding to symbols in C.

The objective partial order $\leq_p$ is a special case of the subjective partial order $\leq_{\lambda,p}$ that is the same for every agent $\lambda \in \Lambda$. Given non-strict partial order $\leq_{\lambda,p}$, the corresponding strict partial order and equivalence relation are $<_{\lambda,p}$ and $=_{\lambda,p}$, respectively.

Definition 8 (Interpretation). An interpretation $\mathscr{I} = \langle D, F \rangle$ of an $\mathscr{A}$-Log formula ϕ consists of a non-empty domain D and an interpretation function F such that

- for each $a \in C$, $F(a) = \overline{a} \in D$;
- for each $p \in Adj$ and each $\lambda \in \Lambda$, $F(p,\lambda,0,0) = P_\lambda \subseteq D$, $F(p,\lambda,0,1) = \mathring{P}_\lambda \subseteq D$, $F(p,\lambda,1,0) = P^*_\lambda \subseteq D$, $F(p,\lambda,1,1) = \mathring{P}^*_\lambda \subseteq D$;
- for each $p \in Adj \setminus Obj$ and each $\lambda \in \Lambda$, $\leq_{\lambda,p} \subseteq D^2$ is a subjective partial order on the elements of D; and
- for each $p \in Obj$, $\leq_p \subseteq D^2$ is an objective partial order on the elements of D (the order is the same for each $\lambda \in \Lambda$, i.e., $\leq_{\lambda,p} = \leq_p$).

Moreover, the following conditions hold for each $p \in Adj$ and each $\lambda \in \Lambda$:

- $P_\lambda \cap \mathring{P}_\lambda = \emptyset$;
- $P^*_\lambda \subseteq P_\lambda$ and $\mathring{P}^*_\lambda \subseteq \mathring{P}_\lambda$;
- if $p \in Unp$, $F(p,\lambda,0,1) = \emptyset$ (an unpolarised adjective has no dual);
- if $\overline{a} \in F(p,\lambda,x,0)$ and $\overline{a'} \notin F(p,\lambda,x,0)$, with $x \in \{0,1\}$, then $\overline{a'} <_{\lambda,p} \overline{a}$;
- if $\overline{a} \in F(p,\lambda,x,1)$ and $\overline{a'} \notin F(p,\lambda,x,1)$, with $x \in \{0,1\}$, then $\overline{a} <_{\lambda,p} \overline{a'}$;
- if $\overline{a} \in F(p,\lambda,x,0)$ and $\overline{a} <_{\lambda,p} \overline{a'}$, then $\overline{a'} \in F(p,\lambda,x,0)$, with $x \in \{0,1\}$;
- if $\overline{a} \in F(p,\lambda,x,1)$ and $\overline{a'} <_{\lambda,p} \overline{a}$, then $\overline{a'} \in F(p,\lambda,x,1)$, with $x \in \{0,1\}$;
- if $\overline{a} \in F(p,\lambda,x,y)$ and $\overline{a'} =_{\lambda,p} \overline{a}$, then $\overline{a'} \in F(p,\lambda,x,y)$, with $x,y \in \{0,1\}$;
- if $\overline{a} \notin F(p,\lambda,x,0)$ and $\overline{a'} <_{\lambda,p} \overline{a}$, then $\overline{a'} \notin F(p,\lambda,x,0)$, with $x \in \{0,1\}$;
- if $\overline{a} \notin F(p,\lambda,x,1)$ and $\overline{a} <_{\lambda,p} \overline{a'}$, then $\overline{a'} \notin F(p,\lambda,x,1)$, with $x \in \{0,1\}$;
- if $\overline{a} \notin F(p,\lambda,x,y)$ and $\overline{a'} =_{\lambda,p} \overline{a}$, then $\overline{a'} \notin F(p,\lambda,x,y)$, with $x,y \in \{0,1\}$.

In Definition 8 we assume that the interpretation P^*_λ (or $\mathring{P}^*_\lambda$) of an intensified adjective is a subset of the interpretation P_λ (or $\mathring{P}_\lambda$) of that adjective, but not necessarily a *proper subset*. This leaves space for interpreting gradable adjectives that do not exhibit intensified versions because they are already extreme to some extent. Consider, for instance, the adjective *gigantic*. Though it is uncommon to use the term *very gigantic*, formally the adjective interpretation coincides with the intensified interpretation.

Definition 9 (Semantic valuation). Given an interpretation $\mathscr{I} = \langle D, F \rangle$, the semantic valuation $[\![\cdot]\!] : \Phi \to \{0,1\}$ is a mapping with the following properties, where $q^{00} = p$, $q^{01} = \mathring{p}, q^{10} = p^*, q^{11} = \mathring{p}^*$, and $x,y \in \{0,1\}$:

- $[\![\bot]\!] = 0$;
- $[\![\lambda : q^{xy}(a)]\!] = 1$ if and only if $F(a) \in F(p,\lambda,x,y)$;
- $[\![\lambda : p^+(a,b)]\!] = 1$ if and only if $F(b) <_{\lambda,p} F(a)$;
- $[\![\lambda : p^\sim(a,b)]\!] = 1$ if and only if $F(b) =_{\lambda,p} F(a)$;
- $[\![O(p)]\!] = 1$ if and only if $p \in Obj$;
- $[\![U(p)]\!] = 1$ if and only if $p \in Unp$;
- $[\![\phi \wedge \psi]\!] = min([\![\phi]\!], [\![\psi]\!])$;
- $[\![\neg\phi]\!] = 1 - [\![\phi]\!]$.

Example 3. *An interpretation of a set of formulae in $\mathscr{A}$-Log based on a partial order devised on purpose.*
Consider the following set of formulae believed by agent *dan*
$S = \{\ dan : tall(frank),\ dan : tall^{*}(helen),$
$dan : \mathring{tall}(edward),\ dan : \neg tall(mira),\ dan : \mathring{tall}^{*}(mira),$
$dan : \neg tall(mark) \wedge \neg \mathring{tall}(mark),\ dan : tall^{\sim}(mark, nicholas),$
$dan : tall^{+}(john, frank),$
$dan : tall^{+}(nicholas, mira),\ dan : \neg tall^{+}(nicholas, helen)\ \}.$
We have that $Adj = \{tall\}$,
$C = \{frank,\ helen,\ edward,\ mira,\ mark,\ nicholas,\ john\}$, and
$\Lambda = \{dan,\ \varepsilon\}$.
Given domain $D = \{\overline{frank},\ \overline{helen},\ \overline{edward},\ \overline{mira},\ \overline{mark},\ \overline{nicholas},\ \overline{john}\}$, a possible partial order $\leq_{dan,tall}$ on the elements of D is represented in the graph of Figure 5.2. Each node represents an element of D and there is an arc from node x to node y if and only if $x <_{dan,tall} y$. The nodes corresponding to elements in $F(tall, dan, 0, 0)$ are filled with green, while those corresponding to elements in $F(tall, dan, 1, 0)$ are filled with red.

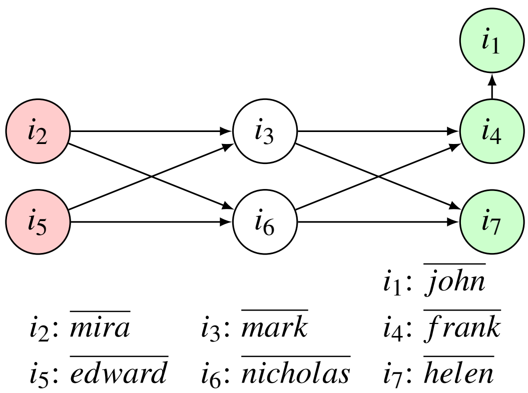

Figure 5.2: A possible partial order $\leq_{dan,tall}$ on the elements of D, where arcs obtained by transitivity have been omitted.

5.3 Proof system

The set $\mathscr{R}$ of $\mathscr{A}$-Log $= \langle \mathscr{L},\ \mathscr{R} \rangle$ comprises three kinds of rules: natural rules, structural rules, and adjectival rules.

Natural rules are the part of Prawitz's presentation of natural deduction that manages the connectives that are also employed in $\mathscr{A}$-Log. We omit to introduce explicitely the rule of *reiteration* that allows to re-employ indefinitely any formula derived in any preceding point. We also omit to explicitly define the rules for $\perp$ introduction and elimination (also known as *Non-contradiction principle* and *Ex falso sequitur quodlibet*), that can be considered implicit in the usage of a natural deduction system.

Structural rules are derived from the label modification operators as presented in labelled non classical logics by [9], and are used to move labels (representing agents asserting unlabelled formulae) in and out of the negation and conjunction symbols. Rule 5 may appear puzzling, because we assume that the system manages vague predicates, that in general do not obey the *tertium non datur* rule. For instance, beautiful and awful are antonyms, but not negations of each other, so that when something is claimed not to be beautiful, we cannot therefore conclude that it is awful (whilst we may do the opposite). This aspect is not, however, contradictory with this rule. When we assert, for instance, that an agent λ does not claim p on a given object a ($\neg\lambda : p(a)$), we *correctly* conclude that the agent (implicitly) claims that a is not p ($\lambda : \neg p(a)$), because this is compatible with the very notion of negation in classical propositional logic, while it differs with the natural semantics of antonyms of adjectives, that are treated in adjectival rules.

Adjectival rules represent the core of the system, and are used to manage the main logical mechanisms. They handle the difference between objective and subjective adjectives, the one between polarised and unpolarised adjectives, the interferences among comparatives, and conflicts in claims by different agents.

Natural rules

Rule 1 (Conjunction Introduction). *Given formulae $\phi, \psi \in \Phi$, if both ϕ and ψ hold, then their conjunction holds:*

$$\frac{\phi, \ \psi}{\phi \wedge \psi} \wedge I$$

Rule 2 (Conjunction Elimination). *Given formulae $\phi, \psi \in \Phi$, if their conjunction holds, then both ϕ and ψ hold:*

$$\frac{\phi \wedge \psi}{\phi} \wedge E$$

$$\frac{\phi \wedge \psi}{\psi} \wedge E$$

Rule 3 (Negation Introduction). *Given formula $\phi \in \Phi$, if the contradiction $\perp$ can be derived from premise ϕ, then $\neg\phi$ holds:*

$$\begin{array}{c} [\phi] \\ \vdots \\ \dfrac{\perp}{\neg\phi} \neg I \end{array}$$

Rule 4 (Negation Elimination). *Given formula $\phi \in \Phi$, if $\neg\neg\phi$ holds, then ϕ holds:*

$$\frac{\neg\neg\phi}{\phi} \neg E$$

Structural rules

Rule 5 (Negation Embedding). *Given unlabelled formula $\phi \in U$, if agent λ does not believe that ϕ holds, then λ believes that ϕ does not hold:*

$$\frac{\neg\lambda : \phi}{\lambda : \neg\phi}\ NE$$

Rule 6 (Negation Extraction). *Given unlabelled formula $\phi \in U$, if agent λ believes that ϕ does not hold, then λ does not believe that ϕ holds:*

$$\frac{\lambda : \neg\phi}{\neg\lambda : \phi}\ NEx$$

Rule 7 (Conjunction Embedding). *Given unlabelled formulae $\phi, \psi \in U$, if agent λ believes that both ϕ and ψ hold, then λ believes that their conjunction holds:*

$$\frac{\lambda : \phi,\ \lambda : \psi}{\lambda : \phi \wedge \psi}\ CE$$

Rule 8 (Conjunction Extraction). *Given unlabelled formulae $\phi, \psi \in U$, if agent λ believes that their conjunction holds, then λ believes that both ϕ and ψ hold:*

$$\frac{\lambda : \phi \wedge \psi}{\lambda : \phi}\ CEx$$

$$\frac{\lambda : \phi \wedge \psi}{\lambda : \psi}\ CEx$$

Adjectival rules

Rule 9 (Degree Introduction). *Given object a, if agent λ does not believe that a is π, with $\pi \in \{p, \mathring{p}\}$, then λ does not believe that a is* very *π:*

$$\frac{\neg\lambda : \pi(a)}{\neg\lambda : \pi^*(a)}\ VI$$

Rule 10 (Degree Elimination). *Given object a, if agent λ believes that a is* very *π, with $\pi \in \{p, \mathring{p}\}$, then λ believes that a is π:*

$$\frac{\lambda : \pi^*(a)}{\lambda : \pi(a)}\ VE$$

Rule 11 (Comparative Introduction). *Given objects a and a', if agent λ believes that a is π and that a' is not π, with $\pi \in \{p, p^*\}$, then λ believes that a is* more *p than a':*

$$\frac{\lambda : \pi(a),\ \neg\lambda : \pi(a')}{\lambda : p^+(a, a')}\ CI$$

Rule 12 (Comparative Introduction 2). *Given objects a and a', if agent λ believes that a is π and that a' is not π, with $\pi \in \{\mathring{p}, \mathring{p}^*\}$, then λ believes that a' is* more *p than a:*

$$\frac{\lambda : \pi(a),\ \neg\lambda : \pi(a')}{\lambda : p^+(a', a)}\ CI2$$

Rule 13 (Comparative Elimination). *Given objects a and a', if agent λ believes that a is π, with $\pi \in \{p, p^*, \neg\mathring{p}, \neg\mathring{p}^*\}$, and that a' is* more *p than a, then λ believes that a' is π:*

$$\frac{\begin{array}{c}\lambda : \pi(a),\\ \lambda : p^+(a', a)\end{array}}{\lambda : \pi(a')}\ CEl$$

Rule 14 (Comparative Elimination 2). *Given objects a and a', if agent λ believes that a is π, with $\pi \in \{\mathring{p}, \mathring{p}^*, \neg p, \neg p^*\}$, and that a is* more *p than a', then λ believes that a' is π:*

$$\frac{\begin{array}{c}\lambda : \pi(a),\\ \lambda : p^+(a, a')\end{array}}{\lambda : \pi(a')}\ CEl2$$

Rule 15 (Equality Elimination). *Given objects a and a', if agent λ believes that a is π, with $\pi \in \{p, p^*, \mathring{p}, \mathring{p}^*, \neg p, \neg p^*, \neg\mathring{p}, \neg\mathring{p}^*\}$, and that a' is* as p as *a, then λ believes that a' is π:*

$$\frac{\lambda : \pi(a), \quad \lambda : p^{\sim}(a', a)}{\lambda : \pi(a')}\ EE$$

Rule 16 (Agent Label Elimination). *Given objects a and a', if agent λ, with $\lambda \neq \varepsilon$, believes that $\pi(a, a')$, with $\pi \in \{p^+, p^{\sim}, \neg p^+, \neg p^{\sim}\}$, and adjective p is* objective*, then $\pi(a, a')$:*

$$\frac{\lambda : \pi(a, a'),\ O(p)}{\pi(a, a')}\ AE$$

Rule 17 (Equality Reflexivity). *Given object a, agent λ believes that a is* as p as *itself:*

$$\frac{}{\lambda : p^{\sim}(a, a)}\ ER$$

Rule 18 (Equality Symmetry). *Given objects a and a', if agent λ believes that a is* as p as *a', then λ believes that a' is* as p as *a:*

$$\frac{\lambda : p^{\sim}(a, a')}{\lambda : p^{\sim}(a', a)}\ ES$$

Rule 19 (Comparative Permutation). *Given objects a and a', if agent λ believes that a is* more *p than a', then λ does not believe that a' is* more *p than a:*

$$\frac{\lambda : p^+(a, a')}{\neg\lambda : p^+(a', a)}\ CP$$

Rule 20 (Comp./Eq. Transitivity). *Given objects a, a' and a'', if agent λ believes that $\pi_1(a,a')$ and that $\pi_2(a', a'')$, with $\pi_1, \pi_2 \in \{p^+, p^\sim\}$, then λ believes that $\pi_3(a,a'')$, where*

$$\pi_3 = \begin{cases} p^+ & \text{if } \pi_1 = p^+ \text{or } \pi_2 = p^+ \\ p^\sim & \text{else} \end{cases}$$

$$\frac{\lambda : \pi_1(a,a'), \quad \lambda : \pi_2(a',a'')}{\lambda : \pi_3(a,a'')} \; CT$$

Rule 21 (Comp./Eq. Transitivity 2). *Given objects a, a' and a'', if agent λ believes that $p^\sim(a',a'')$ and does not believe that $p^+(a',a)$ (or that $p^+(a, a')$), then λ does not believe that $p^+(a'',a)$ (or that $p^+(a,a'')$):*

$$\frac{\neg\lambda : p^+(a',a), \quad \lambda : p^\sim(a',a'')}{\neg\lambda : p^+(a'',a)} \; CT2$$

$$\frac{\neg\lambda : p^+(a,a'), \quad \lambda : p^\sim(a',a'')}{\neg\lambda : p^+(a,a'')} \; CT2$$

Rule 22 (Polarisation Introduction). *Given object a, if agent λ believes that a is the* dual *of p, then adjective p is* polarised*:*

$$\frac{\lambda : \mathring{p}(a)}{\neg U(p)} \; PI$$

Rule 23 (Dual Introduction). *Given object a, if agent λ believes that a is p, then λ does not believe that a is the* dual *of p:*

$$\frac{\lambda : p(a)}{\neg\lambda : \mathring{p}(a)} \; DI$$

Rule 24 (Dual Elimination). *Given object a, if agent λ believes that a is the dual of p, then λ does not believe that a is p:*

$$\frac{\lambda : \mathring{p}(a)}{\neg\lambda : p(a)} \; DE$$

There are no rules for dual majority comparative adjectives because these predicates are not available in $\mathscr{A}$-Log language. In Rules 13 - 16 we slightly abuse the notation by assigning the symbol π to an adjective symbol such as p or to a negated adjective symbol such as $\neg p$. We do this for the sake of simplicity, because otherwise some of these rules should be written twice. One detail to note is that this inference system allows to represent situations in which an agent refrains from giving a judgment, and therefore for instance the formulae at point H of Table 5.2 do not produce a contradiction. This corresponds to a partial order that is not total in the semantics. As noted above, *Ex falso sequitur quodlibet* is not explicitly given as a rule, but it can be obtained by applying Rules 3 and 4 in sequence.
We have now provided a set of inference rules that, as we discuss in Section 5.4, form a sound and complete system of proof for $\mathscr{A}$-Log. A few non-basic rules can be derived from Rules 1 - 24. Some of these derived rules are introduced in Propositions 1, 2, 3, 4 and 5 as consequences of the basic ones.

Proposition 1 (Comparative Irreflexivity). *Given object a, agent λ does not believe that a is* more *p than itself:*

$$\frac{}{\neg\lambda : p^{+}(a,a)}\ CIr$$

Proof. Suppose there is an agent λ that believes that a is *more* p than itself, i.e., $\lambda : p^{+}(a, a)$. Apply Rule 19 (CP) with premise $\lambda : p^{+}(a,a)$ to get
$\neg\lambda : p^{+}(a,a)$. The premise and the conclusion are contradictory, and therefore λ does not believe that a is *more* p than itself, i.e., $\neg\lambda : p^{+}(a,a)$ holds for every agent λ, adjective p, and constant a. □

Proposition 2 (Subjectivity Introduction). *Given objects a and a', if agent λ believes that $\pi(a,a')$, while agent λ' does not believe that $\pi(a,a')$, with $\pi \in \{p^{+}, p^{\sim}\}$, then the adjective p is subjective:*

$$\frac{\lambda : \pi(a,a'),\ \neg\lambda' : \pi(a,a')}{\neg O(p)}\ SubI$$

Proof. Suppose the adjective p is not subjective, i.e., $O(p)$. Begin by applying Rule 5 (NE) to premise $\neg\lambda' : \pi(a,a')$ to get $\lambda' : \neg\pi(a,a')$. Now apply Rule 16 (AE) twice. First, apply it to premises $\lambda : \pi(a,a')$ and $O(p)$ to get $\pi(a,a')$. Second, apply it to premises $\lambda' : \neg\pi(a,a')$ and $O(p)$ to get $\neg\pi(a,a')$. These two conclusions are contradictory, and therefore the adjective p is subjective, i.e., $\neg O(p)$ holds. □

Proposition 3. *Given objects a and a', if agent λ believes that a is* more *p than a', then λ does not believe that a' is* as *p* as *a:*

$$\frac{\lambda : p^{+}(a,a')}{\neg\lambda : p^{\sim}(a',a)}$$

Proof. Suppose λ believes that a' is *as p as a*, i.e., $\lambda : p^{\sim}(a',a)$. Then apply Rule 20 (CT) with premises

$$\lambda : p^{+}(a,a'), \lambda : p^{\sim}(a',a)$$

to derive $\lambda : p^{+}(a,a)$. This is in contradiction with Proposition 1 (CIr), that gives $\neg\lambda : p^{+}(a,a)$. Therefore, λ does not believe that a' is *as p as a*, i.e., $\neg\lambda : p^{\sim}(a',a)$ holds. □

Proposition 4. *Given objects a and a', if agent λ believes that a is* as *p* as *a', then λ believes neither that a' is* more *p than a nor that a is* more *p than a':*

$$\frac{\lambda : p^{\sim}(a,a')}{\neg\lambda : p^{+}(a',a)}$$

Proof. Suppose λ believes that a' is *more p* than a, i.e., $\lambda : p^{+}(a',a)$. Then apply Rule 20 (CT) with premises $\lambda : p^{\sim}(a,a')$ and $\lambda : p^{+}(a',a)$ to get $\lambda : p^{+}(a,a)$. This is in contradiction with Proposition 1 (CIr), that gives $\neg\lambda : p^{+}(a,a)$.

On the other hand, suppose λ believes that a is *more p* than a', i.e., $\lambda : p^{+}(a,a')$. First apply Rule 18 (ES) with premise $\lambda : p^{\sim}(a,a')$ to get $\lambda : p^{\sim}(a',a)$. Then apply Rule 20 (CT) with premises $\lambda : p^{\sim}(a',a)$ and $\lambda : p^{+}(a,a')$ to get $\lambda : p^{+}(a',a')$ and the same contradiction with Proposition 1 (CIr) as above.

Therefore, λ believes neither that a' is *more p* than a nor that a is *more p* than a', i.e., $\neg\lambda : p^{+}(a',a)$ and $\neg\lambda : p^{+}(a,a')$ hold. □

Proposition 5. *Given objects a and a', if agent λ believes that a is p and that a' is the dual of p, then λ believes that a is* more *p than a':*

$$\frac{\lambda : p(a),\ \lambda : \mathring{p}(a')}{\lambda : p^{+}(a,a')}$$

Proof. Apply Rule 24 (DE) to $\lambda : \mathring{p}(a')$ to get $\neg\lambda : p(a')$. Then, apply Rule 11 (CI) to $\lambda : p(a)$ and $\neg\lambda : p(a')$ to conclude $\lambda : p^{+}(a,a')$. □

A set S of wff is *consistent* if and only if $S \nvdash \bot$. In the following examples we show inconsistent sets.

Example 4. *Inference rules highlight inconsistency on disagreement between agents.* Consider a set S of formulae containing formulae 1-8 and A-H from Example 2. The disagreement between sentences 2 and B represents a conflict, and we show why. Since *tall* is an objective adjective, $O(tall)$ can be added to set S, yielding

$$S' = S \cup \{O(tall)\}.$$

Then, Rule 16 can be applied twice to eliminate the agent label from formulae 2 and B, yielding $tall^{+}(john, frank)$ and $tall^{+}(frank, john)$, that can be added to S', obtaining

$$S'' = S' \cup \{tall^{+}(john, frank),\ tall^{+}(frank, john)\}.$$

Rule 19 can be applied with premise $tall^{+}(john, frank)$ to get $\neg tall^{+}(frank, john)$, yielding set

$$S''' = S'' \cup \{\neg tall^{+}(frank, john)\}.$$

The conflict has become evident, as S''' contains a contradiction and $S''' \vdash \bot$.

Example 5. *Inference rules show a subjective inconsistency.*
Consider the following sentences believed by Carl:

- Trevor is young
- Trevor is old

Intuitively, these sentences are inconsistent. The set of formulae

$$\{carl : young(trevor),\ carl : old(trevor)\}$$

is not expressive enough to represent this inconsistency: the correct way to translate them is by using the dualisation modifier, getting set

$$S = \{carl : young(trevor),\ carl : \mathring{young}(trevor)\}.$$

The adjective *young* is an objective adjective, but the reasoning holds also for a subjective adjective, as Carl himself is holding conflicting beliefs: one cannot believe that a person is simultaneously young and old. One way to derive a contradiction is to use Rule 23 with premise $carl : young(trevor)$ and conclusion $\neg carl : \mathring{young}(trevor)$, yielding $S' = S \cup \{\neg carl : \mathring{young}(trevor)\}$. Another way is to use Rule 24 with premise $carl : \mathring{young}(trevor)$ and conclusion $\neg carl : young(trevor)$, yielding $S'' = S \cup \{\neg carl : young(trevor)\}$. Both S' and S'' are inconsistent, as $S' \vdash \bot$ and $S'' \vdash \bot$.

5.4 Proof theory

First of all, we introduce two theoretical notions that are common in logic: *derivability* and *semantic consequence*.

Definition 10 (Derivability). Let S be a set of formulae, a formula ϕ is derivable from S, written $S \vdash \phi$, if and only if there exists a derivation with conclusion ϕ and all hypotheses in S.

Definition 11 (Semantic consequence). Let S be a set of formulae, a formula ϕ is a semantic consequence of S, written $S \models \phi$, if and only if for all interpretations $\mathscr{I}$ such that $[\![\psi]\!] = 1$ for each formula $\psi \in S$, $[\![\phi]\!] = 1$ holds.

We are now able to analyse the specific properties of the proof system built in Section 5.3: *soundness* is discussed in Section 5.4.1 and *completeness* is discussed in Section 5.4.2.

5.4.1 Soundness

Theorem 1 (Soundness). *Let S be a set of formulae and ϕ a formula, if $S \vdash \phi$ then $S \models \phi$.*

Proof. The natural rules 1 - 4 are sound, as they are the traditional natural deduction rules for connectives $\wedge$ and $\neg$. Also the structural rules 5 - 8 are trivially sound, as they only allow to move connectives $\wedge$ and $\neg$ outside or inside an agent label λ. We focus on the adjectival rules, that are specific for the calculus presented here. Since $S \vdash \phi$, there is a derivation $\mathscr{D}$ with conclusion ϕ and hypotheses in S. The proof is by induction on the length of the derivation $\mathscr{D}$, and we do a case analysis of the last used rule in $\mathscr{D} = \mathscr{D}'\phi$. Recall that the interpretation of a is $F(a) = \bar{a}$.

– (Basis) If $\mathscr{D}$ has only one element ϕ and $\phi \in S$, then trivially $S \models \phi$.
– (Basis - Rule 17: ER) $S \models \lambda : p^{\sim}(a,a)$ if and only if $\overline{a} =_{\lambda,p} \overline{a}$. Since $=_{\lambda,p}$ is an equivalence relation, reflexivity holds and $\overline{a} =_{\lambda,p} \overline{a}$. Therefore, $S \models \lambda : p^{\sim}(a,a)$.
– (Rule 9: VI) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : \neg\pi(a)$ holds, with $\pi \in \{p, \mathring{p}\}$.
  $S \models \lambda : \neg\pi(a)$ if and only if $\overline{a} \notin F(p,\lambda,0,y)$, with $y \in \{0,1\}$. Since $P^*_\lambda \subseteq P_\lambda$ and $\mathring{P}^*_\lambda \subseteq \mathring{P}_\lambda$, we have that $\overline{a} \notin F(p,\lambda,1,y)$, with $y \in \{0,1\}$. $\overline{a} \in F(p,\lambda,1,y)$ if and only if $S \models \lambda : \pi^*(a)$, hence $S \models \lambda : \neg\pi^*(a)$.
– (Rule 10: VE) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : \pi^*(a)$ holds, with $\pi \in \{p, \mathring{p}\}$.
  $S \models \lambda : \pi^*(a)$ if and only if $\overline{a} \in F(p,\lambda,1,y)$, with $y \in \{0,1\}$. Since $P^*_\lambda \subseteq P_\lambda$ and $\mathring{P}^*_\lambda \subseteq \mathring{P}_\lambda$, we have that $\overline{a} \in F(p,\lambda,0,y)$, with $y \in \{0,1\}$. $\overline{a} \in F(p,\lambda,0,y)$ if and only if $S \models \lambda : \pi(a)$, hence $S \models \lambda : \pi(a)$.
– (Rule 11: CI) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \lambda : \pi(a)$ and $S \models \neg\lambda : \pi(a')$ hold, with $\pi \in \{p, p^*\}$.
  $S \models \lambda : \pi(a)$ if and only if $\overline{a} \in F(p,\lambda,x,0)$, and $S \models \neg\lambda : \pi(a')$ if and only if $\overline{a'} \notin F(p,\lambda,x,0)$, with $x \in \{0,1\}$. Since $\overline{a} \in F(p,\lambda,x,0)$ and $\overline{a'} \notin F(p,\lambda,x,0)$, then $\overline{a'} <_{\lambda,p} \overline{a}$. $\overline{a'} <_{\lambda,p} \overline{a}$ if and only if $S \models \lambda : p^+(a,a')$, hence $S \models \lambda : p^+(a,a')$.
– (Rule 12: CI2) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \lambda : \pi(a)$ and $S \models \neg\lambda : \pi(a')$ hold, with $\pi \in \{\mathring{p}, \mathring{p}^*\}$.
  $S \models \lambda : \pi(a)$ if and only if $\overline{a} \in F(p,\lambda,x,1)$, and $S \models \neg\lambda : \pi(a')$ if and only if $\overline{a'} \notin F(p,\lambda,x,1)$, with $x \in \{0,1\}$. Since $\overline{a} \in F(p,\lambda,x,1)$ and $\overline{a'} \notin F(p,\lambda,x,1)$, then $\overline{a} <_{\lambda,p} \overline{a'}$. $\overline{a} <_{\lambda,p} \overline{a'}$ if and only if $S \models \lambda : p^+(a',a)$, hence $S \models \lambda : p^+(a',a)$.
– (Rule 13: CEl) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \lambda : \pi(a)$ and $S \models \lambda : p^+(a',a)$ hold, with $\pi \in \{p, p^*, \neg\mathring{p}, \neg\mathring{p}^*\}$.
  $S \models \lambda : p^+(a',a)$ if and only if $\overline{a} <_{\lambda,p} \overline{a'}$. Consider the case where $\pi \in \{p, p^*\}$. $S \models \lambda : \pi(a)$ if and only if $\overline{a} \in F(p,\lambda,x,0)$, with $x \in \{0,1\}$. Since $\overline{a} \in F(p,\lambda,x,0)$ and $\overline{a} <_{\lambda,p} \overline{a'}$, then $\overline{a'} \in F(p,\lambda,x,0)$. $\overline{a'} \in F(p,\lambda,x,0)$ if and only if $S \models \lambda : \pi(a')$, hence $S \models \lambda : \pi(a')$.
  Consider the case where $\pi \in \{\neg\mathring{p}, \neg\mathring{p}^*\}$. $S \models \lambda : \pi(a)$ if and only if $\overline{a} \notin F(p,\lambda, x,1)$, with $x \in \{0,1\}$. Since $\overline{a} \notin F(p,\lambda,x,1)$ and $\overline{a} <_{\lambda,p} \overline{a'}$, then $\overline{a'} \notin F(p,\lambda,x,1)$. $\overline{a'} \notin F(p,\lambda,x,1)$ if and only if $S \models \lambda : \pi(a')$, hence $S \models \lambda : \pi(a')$.
– (Rule 14: CEl2) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \lambda : \pi(a)$ and $S \models \lambda : p^+(a,a')$ hold, with $\pi \in \{\mathring{p}, \mathring{p}^*, \neg p, \neg p^*\}$.
  $S \models \lambda : p^+(a,a')$ if and only if $\overline{a'} <_{\lambda,p} \overline{a}$. Consider the case where $\pi \in \{\mathring{p}, \mathring{p}^*\}$. $S \models \lambda : \pi(a)$ if and only if $\overline{a} \in F(p,\lambda,x,1)$, with $x \in \{0,1\}$. Since $\overline{a} \in F(p,\lambda,x,1)$ and $\overline{a'} <_{\lambda,p} \overline{a}$, then $\overline{a'} \in F(p,\lambda,x,1)$. $\overline{a'} \in F(p,\lambda,x,1)$ if and only if $S \models \lambda : \pi(a')$, hence $S \models \lambda : \pi(a')$.
  Consider the case where $\pi \in \{\neg p, \neg p^*\}$. $S \models \lambda : \pi(a)$ if and only if $\overline{a} \notin F(p,\lambda, x,0)$, with $x \in \{0,1\}$. Since $\overline{a} \notin F(p,\lambda,x,0)$ and $\overline{a'} <_{\lambda,p} \overline{a}$, then $\overline{a'} \notin F(p,\lambda,x,0)$. $\overline{a'} \notin F(p,\lambda,x,0)$ if and only if $S \models \lambda : \pi(a')$, hence $S \models \lambda : \pi(a')$.
– (Rule 15: EE) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S

containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \lambda : \pi(a)$ and $S \models \lambda : p^{\sim}(a',a)$ hold, with $\pi \in \{p, p^*, \mathring{p}, \mathring{p}^*, \neg p, \neg p^*, \neg\mathring{p}, \neg\mathring{p}^*\}$.
$S \models \lambda : p^{\sim}(a',a)$ if and only if $\overline{a'} =_{\lambda,p} \overline{a}$. Consider the case where $\pi \in \{p, p^*, \mathring{p}, \mathring{p}^*\}$. $S \models \lambda : \pi(a)$ if and only if $\overline{a} \in F(p,\lambda,x,y)$, with $x,y \in \{0,1\}$. Since $\overline{a} \in F(p,\lambda,x,y)$ and $\overline{a'} =_{\lambda,p} \overline{a}$, then $\overline{a'} \in F(p,\lambda,x,y)$.
Consider the case where $\pi \in \{\neg p, \neg p^*, \neg\mathring{p}, \neg\mathring{p}^*\}$. $S \models \lambda : \pi(a)$ if and only if $\overline{a} \notin F(p,\lambda,x,y)$, with $x,y \in \{0,1\}$. Since $\overline{a} \notin F(p,\lambda,x,y)$ and $\overline{a'} =_{\lambda,p} \overline{a}$, then $\overline{a'} \notin F(p, \lambda,x,y)$.

- (Rule 16: AE) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models O(p)$ and $S \models \lambda : \pi(a,a')$ hold, with $\pi \in \{p^+, p^{\sim}, \neg p^+, \neg p^{\sim}\}$.
  $S \models O(p)$ if and only if $p \in Obj$. Since $p \in Obj$, $\leq_p$ is a partial order on the elements of D such that $\leq_{\lambda,p} = \leq_p$ for each $\lambda \in \Lambda$.
  Consider the case where $\pi \in \{p^+, p^{\sim}\}$. $S \models \lambda : p^+(a,a')$ if and only if $\overline{a'} <_{\lambda,p} \overline{a}$, and $S \models \lambda : p^{\sim}(a,a')$ if and only if $\overline{a'} =_{\lambda,p} \overline{a}$. If $\overline{a'} <_{\lambda,p} \overline{a}$ for each $\lambda \in \Lambda$, then $\overline{a'} <_{\varepsilon,p} \overline{a}$ and therefore $S \models \varepsilon : p^+(a,a')$. If, on the other hand, $\overline{a'} =_{\lambda,p} \overline{a}$ for each $\lambda \in \Lambda$, then $\overline{a'} =_{\varepsilon,p} \overline{a}$ and therefore $S \models \varepsilon : p^{\sim}(a,a')$.
  Consider the case where $\pi \in \{\neg p^+, \neg p^{\sim}\}$. $S \models \lambda : \neg p^+(a,a')$ if and only if $\overline{a'} \not<_{\lambda,p} \overline{a}$, and $S \models \lambda : \neg p^{\sim}(a,a')$ if and only if $\overline{a'} \neq_{\lambda,p} \overline{a}$. If $\overline{a'} \not<_{\lambda,p} \overline{a}$ for each $\lambda \in \Lambda$, then $\overline{a'} \not<_{\varepsilon,p} \overline{a}$ and therefore $S \models \varepsilon : \neg p^+(a,a')$. If, on the other hand, $\overline{a'} \neq_{\lambda,p} \overline{a}$ for each $\lambda \in \Lambda$, then $\overline{a'} \neq_{\varepsilon,p} \overline{a}$ and therefore $S \models \varepsilon : \neg p^{\sim}(a,a')$.
- (Rule 18: ES) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : p^{\sim}(a,a')$ holds.
  $S \models \lambda : p^{\sim}(a,a')$ if and only if $\overline{a'} =_{\lambda,p} \overline{a}$. Since $=_{\lambda,p}$ is an equivalence relation, symmetry holds and $\overline{a} =_{\lambda,p} \overline{a'}$. Therefore, $S \models \lambda : p^{\sim}(a',a)$.
- (Rule 19: CP) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : p^+(a,a')$ holds.
  $S \models \lambda : p^+(a,a')$ if and only if $\overline{a'} <_{\lambda,p} \overline{a}$, and therefore $\overline{a} \not<_{\lambda,p} \overline{a'}$. Thus, $S \models \neg\lambda : p^+(a',a)$.
- (Rule 20: CT) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \lambda : \pi_1(a,a')$ and $S \models \lambda : \pi_2(a',a'')$ hold, with $\pi_1, \pi_2 \in \{p^+, p^{\sim}\}$.
  $S \models \lambda : p^+(a,a')$ if and only if $\overline{a'} <_{\lambda,p} \overline{a}$, and $S \models \lambda : p^+(a',a'')$ if and only if $\overline{a''} <_{\lambda,p} \overline{a'}$. Moreover, $S \models \lambda : p^{\sim}(a,a')$ if and only if $\overline{a'} =_{\lambda,p} \overline{a}$, and $S \models \lambda : p^{\sim}(a', a'')$ if and only if $\overline{a''} =_{\lambda,p} \overline{a'}$. Since $\leq_{\lambda,p}$ is a partial order, it is transitive. Therefore, if at least one among π_1 and π_2 is p^+, $\overline{a''} <_{\lambda,p} \overline{a}$ holds, and thus $S \models \lambda : p^+(a,a'')$. On the other hand, if both π_1 and π_2 are $p^{\sim}$, $\overline{a''} =_{\lambda,p} \overline{a}$ holds, and thus $S \models \lambda : p^{\sim}(a, a'')$.
- (Rule 21: CT2) Induction hypothesis: $\mathscr{D}'$, $\mathscr{E}'$ are derivations such that for each S containing the hypotheses of $\mathscr{D}'$ and $\mathscr{E}'$, $S \models \neg\lambda : p^+(a',a)$ (or $S \models \neg\lambda : p^+(a,a')$) and $S \models \lambda : p^{\sim}(a',a'')$ hold.
  $S \models \lambda : p^{\sim}(a',a'')$ if and only if $\overline{a'} =_{\lambda,p} \overline{a''}$. Moreover, $S \models \neg\lambda : p^+(a',a)$ if and only if $\overline{a} \not<_{\lambda,p} \overline{a'}$, while on the other hand $S \models \neg\lambda : p^+(a,a')$ if and only if $\overline{a'} \not<_{\lambda,p} \overline{a}$. Since $\overline{a'} =_{\lambda,p} \overline{a''}$, if $\overline{a} \not<_{\lambda,p} \overline{a'}$, then $\overline{a} \not<_{\lambda,p} \overline{a''}$; while on the other hand if $\overline{a'} \not<_{\lambda,p} \overline{a}$,

then $\overline{a''} \not<_{\lambda,p} \overline{a}$.

- (Rule 22: PI) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : \mathring{p}(a)$ holds.
  $S \models \lambda : \mathring{p}(a)$ if and only if $\overline{a} \in F(p,\lambda,1,0) = \mathring{P}_{\lambda}$. Therefore, $F(p,\lambda,0,1) \neq \emptyset$. Since if $p \in Unp$, then $F(p,\lambda,0,1) = \emptyset$, we can conclude that $p \notin Unp$. $S \models U(p)$ if and only if $p \in Unp$, and thus $S \models \neg U(p)$.
- (Rule 23: DI) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : p(a)$ holds.
  $S \models \lambda : p(a)$ if and only if $\overline{a} \in F(p,\lambda,0,0) = P_{\lambda}$. Since $P_{\lambda} \cap \mathring{P}_{\lambda} = \emptyset$, we have that $\overline{a} \notin F(p,\lambda,1,0) = \mathring{P}_{\lambda}$. $\overline{a} \in F(p,\lambda,1,0)$ if and only if $S \models \lambda : \mathring{p}(a)$, hence $S \models \neg\lambda : \mathring{p}(a)$.
- (Rule 24: DE) Induction hypothesis: $\mathscr{D}'$ is a derivation such that for each S containing the hypotheses of $\mathscr{D}'$, $S \models \lambda : \mathring{p}(a)$ holds.
  $S \models \lambda : \mathring{p}(a)$ if and only if $\overline{a} \in F(p,\lambda,1,0) = \mathring{P}_{\lambda}$. Since $P_{\lambda} \cap \mathring{P}_{\lambda} = \emptyset$, we have that $\overline{a} \notin F(p,\lambda,0,0) = P_{\lambda}$. $\overline{a} \in F(p,\lambda,0,0)$ if and only if $S \models \lambda : p(a)$, hence $S \models \neg\lambda : p(a)$.

Since the induction basis and the induction steps for each rule are proven, the claim is proven. □

5.4.2 Completeness

For a consistent set of wff it is always possible to obtain a corresponding consistent atomic theory, therefore, w.l.o.g., for the proof of completeness we consider consistent atomic theories. Moreover, the sets that we are interested in are closed under derivability $\vdash$.

Lemma 1. *Given an atomic theory S, its corresponding derivability-closed atomic theory T can be obtained in a finite number of steps.*

Proof. Given a signature Σ with n adjective symbols $Adj = \{p_1,\ldots,p_n\}$, m object constant symbols $C = \{a_1,\ldots,a_m\}$, and $l+1$ agent symbols $\Lambda = \{\lambda_1,\ldots,\lambda_l,\varepsilon\}$, the number of literals that can be generated is finite, therefore a derivability-closed atomic theory can be obtained by iteratively applying the inference rules until no new literal is generated. More in detail, initially S can be extended in S' by adding the literals generated by applying Rule 17, that does not interfere with other rules of the system. Then, the remaining rules can be divided in a group where the generated literal has a unary predicate, composed of Rules 4, 5, 6, 9, 10, 13, 14, 15, 22, 23, and 24; and a group where the generated literal has a binary predicate, composed of Rules 11, 12, 16, 18, 19, 20, and 21. These two groups of rules are to be applied iteratively, as the literal generated by a rule from one group may be an antecedent for a rule of the other group. □

Definition 12 (Standard valuation). Given a consistent atomic theory T, the standard valuation $[\![\cdot]\!]$ is obtained by applying Definition 9 on the interpretation $\mathscr{I} = \langle D, F\rangle$ of T that is defined as:

- $D = \{\hat{a} \mid a \in C\}$;
- for each constant symbol $a \in C$, $F(a) = \hat{a} \in D$;
- for each $p \in Adj$ and each $\lambda \in \Lambda$, define sets

- $F(p,\lambda,0,0) = P_\lambda \subseteq D$ by $\hat{a} \in P_\lambda$ if and only if $T \vdash \lambda : p(a)$,
- $F(p,\lambda,0,1) = \mathring{P}_\lambda \subseteq D$ by $\hat{a} \in \mathring{P}_\lambda$ if and only if $T \vdash \lambda : \mathring{p}(a)$,
- $F(p,\lambda,1,0) = P^*_\lambda \subseteq D$ by $\hat{a} \in P^*_\lambda$ if and only if $T \vdash \lambda : p^*(a)$,
- $F(p,\lambda,1,1) = \mathring{P}^*_\lambda \subseteq D$ by $\hat{a} \in \mathring{P}^*_\lambda$ if and only if $T \vdash \lambda : \mathring{p}^*(a)$;

– for each $p \in Adj \setminus Obj$ and each $\lambda \in \Lambda$, define $\leq_{\lambda,p} \subseteq D^2$ as a partial order on the elements of D such that
  - $F(b) <_{\lambda,p} F(a)$ if and only if $T \vdash \lambda : p^+(a,b)$, and
  - $F(b) =_{\lambda,p} F(a)$ if and only if $T \vdash \lambda : p^\sim(a,b)$;

– for each $p \in Obj$, define $\leq_p \subseteq D^2$ as a partial order on the elements of D such that
  - $F(b) <_p F(a)$ if and only if $T \vdash p^+(a,b)$, and
  - $F(b) =_p F(a)$ if and only if $T \vdash p^\sim(a,b)$.

Lemma 2. *Given a consistent atomic theory T, the standard valuation $[\![\cdot]\!]$ is such that $[\![\phi]\!] = 1$ for all $\phi \in T$.*

Proof. Note that $\phi \in T$ implies $T \vdash \phi$.

– Since T is consistent, $\perp \notin T$.
– Consider $q^{00} = p, q^{01} = \mathring{p}, q^{10} = p^*, q^{11} = \mathring{p}^*$, and $x,y \in \{0,1\}$. $[\![\lambda : q^{xy}(a)]\!] = 1$ if and only if $F(a) \in F(p,\lambda,x,y)$ by Definition 9. $T \vdash \lambda : q^{xy}(a)$ implies $F(a) \in F(p, \lambda,x,y)$ by Definition 12.
– $[\![\lambda : p^+(a,b)]\!] = 1$ if and only if $F(b) <_{\lambda,p} F(a)$ by Definition 9. $T \vdash \lambda : p^+(a,b)$ implies $F(b) <_{\lambda,p} F(a)$ by Definition 12.
– $[\![\lambda : p^\sim(a,b)]\!] = 1$ if and only if $F(b) =_{\lambda,p} F(a)$ by Definition 9. $T \vdash \lambda : p^\sim(a,b)$ implies $F(b) =_{\lambda,p} F(a)$ by Definition 12.
– $[\![\neg\phi]\!] = 1$ if and only if $[\![\phi]\!] = 0$ by Definition 9. $T \vdash \neg\phi$ implies $[\![\phi]\!] = 0$ by Definition 12, because all the constraints are biconditionals.

Since all the possible cases were considered, the claim is proven. □

Theorem 2 (Completeness). *Let S be a set of formulae and ϕ a formula, if $S \models \phi$ then $S \vdash \phi$.*

Proof. We prove the contrapositive: if $S \nvdash \phi$ then $S \not\models \phi$. Consider the atomic theory T corresponding to S. W.l.o.g., consider also ϕ to be a literal. If $T \nvdash \phi$, then $T \cup \{\neg\phi\}$ is consistent. Therefore, by Lemma 2, there is a valuation $[\![\cdot]\!]$ such that $[\![\psi]\!] = 1$ for all $\psi \in T \cup \{\neg\phi\}$. Equivalently, $[\![\psi]\!] = 1$ for all $\psi \in T$ and $[\![\phi]\!] = 0$. Hence, $T \not\models \phi$ and $S \not\models \phi$.

□

5.5 Algorithms

The idea underlying the algorithms we present in this section is to derive consistency of an atomic theory. We provide results for atomic theories and then we show how to adapt them to general theories. The principle is that if a theory T is consistent, then there exists an atomic theory AT that derives from T and is consistent.

The core element of the method is presented in Algorithm 3, and further in Algorithm 4. Essentially, Algorithm 3 searches a fixpoint in the rule propagation. However, it does not propagate part of the rules, those numbered 17-21, as these rules require, to converge, a number of steps that depends on the partial order relations appearing in the $\mathscr{A}$-Log theory. We need to prove that the number of these steps is polynomial in order to exhibit a polynomial method for nondeterministic machines. To do so, we extract these rules and treat their convergence separately. We found that it is possible to manage this part of the method by means of constraint processing techniques.
We say that a rule *r1* propagates on a rule *r2* when applying *r1* to all formulae in an atomic theory produces a set of formulae on which *r2* apply. Lemma 3 introduces the propagation relations on rules .

Lemma 3. *The following propagation relations hold:*

- *Rule 4 (¬E) propagates on Rules 6, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24.*
- *Rule 5 (NE) propagates on Rules 13, 14, 15, 16.*
- *Rule 6 (NEx) propagates on Rules 9, 11, 12, 21.*
- *Rule 9 (VI) propagates on Rules 5, 11, 12.*
- *Rule 10 (VE) propagates on Rules 11, 12, 13, 14, 15, 22, 23, 24.*
- *Rule 11 (CI) propagates on Rules 13, 14, 16, 19, 20.*
- *Rule 12 (CI2) propagates on Rules 13, 14, 16, 19, 20.*
- *Rule 13 (CEl) propagates on Rules 6, 10, 11, 13, 15, 23.*
- *Rule 14 (CEl2) propagates on Rules 6, 10, 12, 14, 15, 22, 24.*
- *Rule 15 (EE) propagates on Rules 6, 10, 11, 12, 13, 14, 15, 22, 23, 24.*
- *Rule 16 (AE) propagates on Rules 13, 14, 15, 18, 19, 20, 21.*
- *Rule 17 (ER) propagates on Rules 15, 16, 18, 20, 21.*
- *Rule 18 (ES) propagates on Rules 15, 16, 18, 20, 21.*
- *Rule 19 (CP) propagates on Rules 5, 21.*
- *Rule 20 (CT) propagates on Rules 13, 14, 15, 16, 18, 19, 20, 21.*
- *Rule 21 (CT2) propagates on Rules 16, 21.*
- *Rule 22 (PI) does not propagate on other Rules .*
- *Rule 23 (DI) propagates on Rules 5, 9, 12.*
- *Rule 24 (DE) propagates on Rules 5, 9, 11.*

The first step of the method we describe here is the transformation of the atomic theory into a finite set of constraint satisfaction problems, that represent the conditions on adjective comparatives of majority and equality. After this step we translate the results back, and put the translated objects into the theory. We apply then only a subset of the set $\mathscr{R}$ of inference rules. The two steps cited above are repeated until the execution of the second step stops changing the theory by producing new literals.

The essence of the method is to label systematically the constraint networks by the labels of partial orders. To solve the problem with abstract partial orders we show here that some of the results of Cristani ([42]) is to be employed for the partial order algebra generated by the relations that can be established between two sets, as partly dealt with in an abstract fashion in [57].

First of all, the relation algebra that we shall employ for the mentioned goal is based on the inverse Table 5.3a and the composition Table 5.3b for the basic relations of the set

$B = \{\equiv, \subset, \supset, \bowtie\}$. The relation algebra of *Subset Partial Order* (SPO) is defined as the structure $\mathscr{C} = \langle A, \circ, ^{-1}, \cap, \cup, ^{-}, 0, 1, 1' \rangle$, where we have:

- $\langle A, \cap, \cup, 0, 1, ^{-} \rangle$ is a Boolean algebra,
- $\langle A, \circ, ^{-1}, 1' \rangle$ is an Abelian group,
- $\circ$ and $^{-1}$ are distributive with respect to $\cap$, $^{-}$ and $\cup$.

We say that $B \subseteq A$ is a base for $\mathscr{C}$ when $A = 2^B$. From a semantic viewpoint we name *Interpretation Domain* a set Δ such that for every element in A there is a relation on Δ, namely, a subset of Δ^2.

Specifically, we have that:

- $A = 2^B$;
- $0 = \emptyset$;
- $1 = B$;
- $1' = \{\equiv\}$.

Semantic natural interpretation is the standard interpretation of $\equiv, \subset$ and $\supset$ on a given domain Δ.

Table 5.3: Inverse and composition tables of SPO Relation Algebra.

(a) The inverse table.

	INV
$\equiv$	$\equiv$
$\subset$	$\supset$
$\supset$	$\subset$
$\bowtie$	$\bowtie$

(b) The composition table.

	$\equiv$	$\subset$	$\supset$	$\bowtie$
$\equiv$	$\equiv$	$\subset$	$\supset$	$\bowtie$
$\subset$	$\subset$	$\subset$	$\top$	$\{\subset, \bowtie\}$
$\supset$	$\supset$	$\top$	$\supset$	$\{\supset, \bowtie\}$
$\bowtie$	$\bowtie$	$\{\subset, \bowtie\}$	$\{\supset, \bowtie\}$	$\top$

We define a *Network of Constraints* (CN) over a given Relation Algebra $\mathscr{C}$ as a directed graph labelled on vertices with variable names, and on edges with elements of the set of relations in $\mathscr{C}$. Given a CN on a Relation Algebra $\mathscr{C}$ we define the *Network Satisfaction Problem* (NSP) as the problem of determining whether it is possible to attribute a particular element of the interpretation domain to each variable, on a specific interpretation of the relations on Δ so that each relation of the network is simultaneously true.

To perform the translation into constraint processing problems we proceed as described in Algorithm 1, where the expression p.l. means "positive literal" and n.l. means "negative literal". The expression c.e. means "comparative of equality" (e.g., $p^{\sim}$) and c.max. "comparative of majority" (e.g., p^{+}). For an atomic theory T we name $\mathscr{N}(T)$ the set of networks $N_{\lambda,p}$ formed by Algorithm 1.

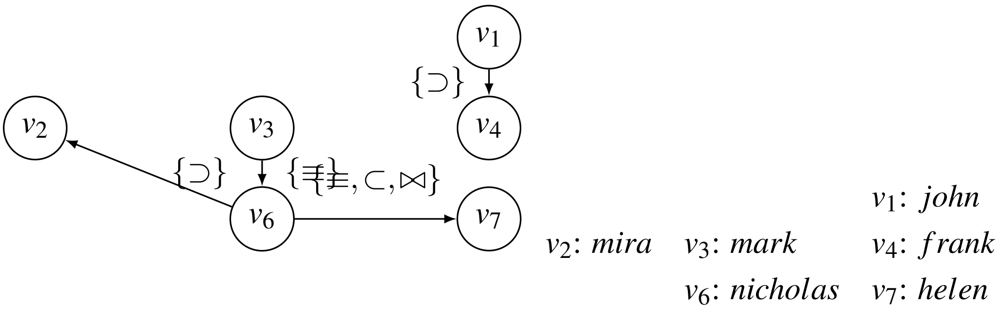

Figure 5.3: A Network Satisfaction Problem obtained by translating an atomic theory with Algorithm 1.

Algorithm 1: Translation of an atomic theory into NSPs.

Data: An atomic theory T in $\mathscr{A}$-Log.
Result: A finite set of Constraint Networks $\mathscr{N}(T)$ on SPO, one for each label and each adjective.
foreach *Assertion* α *in* T **do**
  if *Assertion* α *is* $\lambda : \pi(x,y)$ *or* $\neg\lambda : \pi(x,y)$ **then**
    Add nodes x and y to the network $N_{\lambda,p}$, if inexistent;
    Add the edge between x and y and label it with $\top$, if inexistent;
    Label the edge between x and y, previously labelled by r, with
    – $r \cap \{\equiv\}$ if π is a c.e. and α is a p.l.;
    – $r \cap \{\supset\}$ if π is a c.max. and and α is a p.l.;
    – $r \cap \{\subset, \supset, \bowtie\}$ if π is a c.e. and and α is a n.l.;
    – $r \cap \{\equiv, \subset, \bowtie\}$ if π is a c.max. and and α is a n.l.;
  end
end

In Lemma 4 we provide a computational account for this algorithm. The proof of the lemma is a straightforward consequence of the definition of the number n_λ of labels in a theory T and of the number n_p of adjectives in T, and is therefore omitted for the sake of space. We assume that n represents the maximum between n_λ and n_p.

Lemma 4. *Algorithm 1 takes in input an atomic theory T and computes the set of networks $\mathscr{N}(T)$ in $O(n^2)$, where n is the maximum between n_λ, the number of labels and n_p, the number of adjectives in T.*

Example 6. *Translation from $\mathscr{A}$-Log theory to NSP.*
Consider the set S of formulae of Example 3. By applying one instance of Rule 8 (Conjunction Extraction) we get the corresponding atomic theory T. Figure 5.3 shows the network $N_{dan,tall}$ obtained by applying Algorithm 1 to T.

The set $\mathscr{P}$ contains all expressions in SPO that can be translated directly from assertions of an atomic theory. The NSPs defined on SPO Relation Algebra can be treated polynomially on nondeterministic machines, as proven by Cristani ([42]), and constitute an NP-complete problem. However, the closure $\mathscr{P}^*$ of the set $\mathscr{P}$ is a subset of one of the maximal tractable subsets of *MC*-4, the constraint algebra introduced in the above mentioned study. This is claimed in Lemma 5 referring to the complete classification of the relation algebras generated by the operators of congruence *CG*, *CGPP*, $CGPP^{-1}$ and *CNO*, and in particular to Table 10 on page 387 of the paper from [42] cited above. The proof is a direct straightforward consequence of the above reasoning and is therefore omitted for the sake of space.

In particular, [42] denominates a list of tractable subclasses, among which some are maximal (i.e., they are not contained in any other maximal tractable subclass). One of these maximal tractable subalgebras is M_{99}, the only one containing the basic relations of *MC*-4.

Lemma 5. *$\mathscr{P}^*$ is equivalent to M_{94}, a subalgebra of M_{99}, and therefore the NSP is solved in $O(n^2)$, where n is the number of variables of the NSP.*

The method of constraint processing defined above consists of two steps: translating T into the set $\mathscr{N}$ and then processing $\mathscr{N}$ by the constraint processing method defined for *MC*-4. Algorithm 1 may be easily inverted. Algorithm 2 corresponds to this operation.

Algorithm 2: Translation from NSPs to a theory.

Data: A finite set of Constraint Networks $\mathscr{N}$ on SPO.
Result: An atomic theory T in $\mathscr{A}$-Log.
foreach *Pair of nodes* (x,y) *in a NSP* $N_{\lambda,p}$ *of* $\mathscr{N}$ **do**
  Translate the label on the edge between nodes x and y:
  – If the label is $\{\equiv\}$, add $\lambda : p^{\sim}(x,y)$;
  – If the label is $\{\subset\}$, add $\lambda : p^{+}(y,x)$;
  – If the label is $\{\supset\}$, add $\lambda : p^{+}(x,y)$;
  – If the label is $\{\subset,\supset,\bowtie\}$, add $\neg\lambda : p^{\sim}(x,y)$;
  – If the label is $\{\equiv,\supset,\bowtie\}$, add $\neg\lambda : p^{+}(y,x)$;
  – If the label is $\{\equiv,\subset,\bowtie\}$, add $\neg\lambda : p^{+}(x,y)$;
  – If the label is $\{\supset,\bowtie\}$, add $\neg\lambda : p^{\sim}(x,y)$, $\neg\lambda : p^{+}(y,x)$;
  – If the label is $\{\subset,\bowtie\}$, add $\neg\lambda : p^{\sim}(x,y)$, $\neg\lambda : p^{+}(x,y)$;
  – If the label is $\{\equiv,\bowtie\}$, add $\neg\lambda : p^{+}(y,x)$, $\neg\lambda : p^{+}(x,y)$;
  – If the label is $\{\bowtie\}$, add $\neg\lambda : p^{\sim}(x,y)$, $\neg\lambda : p^{+}(y,x)$, $\neg\lambda : p^{+}(x,y)$.
end

At this point we identify the subset of rules that have not been processed by the algorithm specific for NSPs. The iterative application of these rules can be computed in polynomial time by following the propagations introduced in Lemma 3. The method is illustrated in Algorithm 3.

Algorithm 3: Rewriting rules application.

Data: An atomic theory T in $\mathscr{A}$-Log.

Result: An atomic theory T in $\mathscr{A}$-Log rewritten to a state with no further applicability of Rules 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 22, 23, 24.

```
while At least one new assertion is added to T do
    foreach Assertion α in T do
        Apply Rule 4 with premise α;
        Apply Rule 5 with premise α;
        Apply Rule 6 with premise α;
        Apply Rule 9 with premise α;
        Apply Rule 10 with premise α;
        Apply Rule 11 with premise α;
        Apply Rule 12 with premise α;
        Apply Rule 13 with premise α;
        Apply Rule 14 with premise α;
        Apply Rule 15 with premise α;
        Apply Rule 16 with premise α;
        Apply Rule 22 with premise α;
        Apply Rule 23 with premise α;
        Apply Rule 24 with premise α;
    end
end
```

Rules 17 - 21 are not treated by Algorithm 3 because they are handled by the algorithm specific for NSPs. Complexity of the method is proven in Lemma 6.

Lemma 6. *Algorithm 3 takes in input an atomic theory T and computes the rewritten theory in $O(m^2 \cdot n^2)$, where m is the number of adjective symbols and n is the number of constant symbols in the theory.*

Proof. The inner cycle of the algorithm is performed once and takes a maximum of $O(m \cdot n)$ steps. The algorithm keeps iterating if at least one literal has been added, and the rules that can be involved are the following:

- Rules 4, 5, 6, 9, 10, 13, 14, 15, 22, 23, and 24, that introduce single assertions;
- Rules 11, 12, 16, that introduce comparative assertions.

Since for every literal there is only one adjective symbol, and the number of constants for each single and comparative assertion is respectively 1 and 2, the total number of literals involved in each rewriting is $O(m \cdot n)$. Thus, the maximum number of steps of the method is $O(m^2 \cdot n^2)$. □

Therefore, at most, the algorithm is executed a number of times that corresponds to the application of the rules introducing single or comparative assertions. In other terms, if l is the number of literals in the theory, Algorithm 3 takes $O(l^2)$ steps to complete. Algorithm 4 introduces the full method.

Algorithm 4: Consistency checking for an $\mathscr{A}$-Log Theory.

Data: An atomic theory T in $\mathscr{A}$-Log.
Result: **Yes**, if the theory is consistent; **No**, if it is not.
while *At least one change is performed by Step 2 or 4.* **do**
  1. Execute Algorithm 1;
  2. Execute Constraint propagation by Algorithm of M_{99}-consistency on the computed NSPs;
  3. Execute Algorithm 2;
  4. Execute Algorithm 3.
end

Algorithm 4 works appropriately, as shown in Theorem 3.

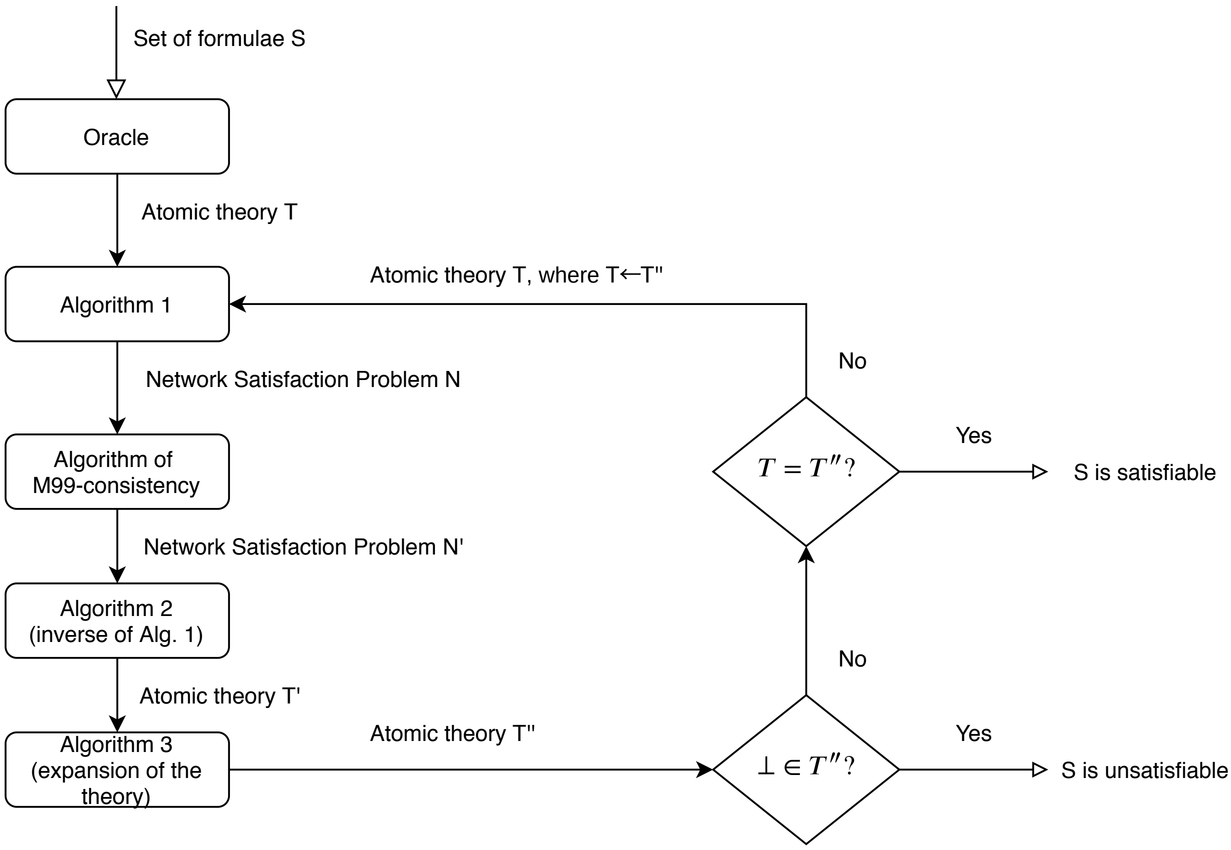

Figure 5.4: Working schema of Algorithm 4 for consistency checking in $\mathscr{A}$-Log.

Theorem 3. *Algorithm 4 correctly computes consistency of an $\mathscr{A}$-Log atomic theory in $O(l^3)$, where l is the number of literals that can be produced with symbols in the theory.*

Proof. We prove the claim in two steps. First of all we show that the steps within the outer while cycle guarantee the propagation of the set of Rules 6-24. This is straightforward for the presence of the two techniques executed in Steps 2 and 4, and because the execution of the checks required for providing these rules is repeated until no change occurs.

Secondly, the algorithm propagates all the rules until no rule applies again, as at the end of a cycle it repeats if and only if at least one rule applied. Every rule is activated only when the premises result valid, and this can only happen if we have, in the worst case, both an assertion of type $\pi(a)$ and an assertion of type $\pi(a,a')$, where a and a' are constants. No rules involve more than two assertions, and therefore we assume that the number of possible activations is $O(l^2)$. The inner cycle generates in the worst case one of these elements at a time, and therefore the outer cycle repeats at most $O(l)$. Therefore the computational account for the algorithm is $O(l^3)$. This proves the claim. □

We now employ the polynomial method developed for the atomic theories to determine the computational complexity class of the general consistency checking problem in $\mathscr{A}$-Log. We shall do this in two steps. First of all, we prove that the problem can be reduced to a NP-complete problem, by showing that a method that solves consistency checking in $\mathscr{A}$-Log is able to solve consistency checking in classical propositional logic (Lemma 7).

Lemma 7. *$\mathscr{A}$-Log consistency checking is NP-hard.*

Proof. Consider a finite set S of propositional formulae in which only negation and conjunction symbols appear. Assume that the distinct literals appearing in the set are n. Construct a dummy predicate p, and a finite set of dummy objects $o_1, o_2, ..., o_n$. Now map each of the positive literals in S into a literal of $\mathscr{A}$-Log $p(o_i)$ and each negative literal into the corresponding negative literal $\neg p(o_j)$. Finally map each expression of S by using conjunction and negation from the language of $\mathscr{A}$-Log. This process takes a number of steps that is linear in the number of distinct literals appearing in the set (with $O(n)$ complexity). The obtained set of formulae is an $\mathscr{A}$-Log apodeictic theory.

We polynomially map a problem of consistency checking on classic propositional calculus into the consistency checking of an $\mathscr{A}$-Log apodeictic theory. Since the consistency checking problem for classic propositional calculus is not polynomially solvable on deterministic machines (provided that $P \neq NP$), the problem of consistency checking in $\mathscr{A}$-Log is NP-hard.

□

As usual in propositional systems, when dealing with a two-valued interpretation, we speak of the *assignment* of the letters of $\mathscr{A}$-Log, that are bound to be true or false. We say that an atomic theory AT derives from a general $\mathscr{A}$-Log theory T when the following hold:

- All the assignments of AT are also assignments of T;
- If the theory AT is satisfied for all the literals appearing in AT, then T is satisfied.

Atomic theories are to be thus used to prove consistency of non-atomic theories. Note that we do not claim *bijective correspondence* between satisfiability of atomic theory AT and satisfiability of theory T that AT derives from. An atomic theory may be inconsistent and derive from a theory that is consistent. More precisely we may generally state the following correspondence: if an $\mathscr{A}$-Log theory T is consistent, then there exists an $\mathscr{A}$-Log atomic theory AT that derives from T and is consistent.

Since we can check consistency of an atomic theory in polynomial time, we are able to formulate a solution for consistency checking in $\mathscr{A}$-Log that takes polynomial time on nondeterministic machines, as proven in Lemma 8.

Lemma 8. *$\mathscr{A}$-Log consistency checking is NP.*

Proof. Consider an oracle, able to select a consistent atomic theory from an $\mathscr{A}$-Log theory *if and only if* it exists. The oracle gives a positive answer, providing a theory that it checks to be consistent in polynomial time; or a negative answer, concluding that T is inconsistent because there is no consistent atomic theory that derives from T. If such an oracle exists, we can solve the problem polynomially. Thus the problem is polynomially solvable on nondeterministic machines. □

The summary of Lemmata 7 and 8 is expressed in Theorem 5:

Theorem 4. *The $\mathscr{A}$-Log consistency checking problem in NP-complete.*

Proof. By Lemma 7 we know that the problem of consistency checking is NP-hard, and by Lemma 8 we know that it is also polynomially solvable on deterministic machines. Therefore, by definition, the problem is NP-complete. □

Chapter 6

An Ontology of Changes in Normative Systems

Defeasible deontic logic has shown to be expressive enough to represent a normative system, and therefore compliance to such a system can be automatically checked by means of classical model checking techniques of logical systems. However, normative systems are not static, as they can be actively changed by the legislator over time, directly, by changing one norm. Moreover norms can change passively, either by effect of the change of another piece of the normative system, or by means of the change of meaning that affects terms employed in the norm.

Although some efforts have been carried out by scholars in the field of legal reasoning about norm change, there is a lack of uniformity in the representation of these changes, and this is an issue when we aim at deploying the law as an automated platform: we need to introduce changes as effects in the semantics of derivation in a logical system, when the unified viewpoint admits a unified representation as well. We adopt the logical paradigm of agency and provide a classification of changes from an agentive viewpoint that allows a unified representation within the language of defeasible deontic logic.

Defeasible deontic logic is one of the most successful frameworks for the representation of normative systems. To determine the exact application of a norm we need to identify the stucture of the norm from the viewpoint of its *temporal* and *territorial* extensions and its application field, namely the individuals who are subject to the norm.

The usage of deontic defeasible logic is common for two reasons: since the normative systems are represented in a declarative fashion it is relatively easy to manage tests for compliance [73], the process of revision in defeasible logic is easier than it is in classical logic, especially when considered from a conflict-resolution perspective [72, 76, 122, 77, 122].

Many efforts are known in the literature of deontic defeasible logic for denoting normative systems, and some efforts have been also put on the problem of managing change in normative systems as captured by the above mentioned scheme [78, 45, 70]. In this chapter we deal with the problem of classifying in a general framework the ways in which a norm can change, either actively, namely by means of the intervention of a legislator, or passively, as a consequence of the change in the meaning of terms.

The rest of the Chapter is organized as follows: Section 6.1 presents the framework used for the definition of the problem, Section 6.2 is devoted to discuss ways in which such a framework provides room for many different explicit changes, and Section 6.3 shows the implicit changes.

6.1 Norms in defeasible deontic logic

We assume that norms are represented in defeasible deontic logic by the Definition that follows.

Definition 13 (Norm). A norm n is a finite set of rules in defeasible deontic logic, where each rule takes one of the following forms:

– A definition $l_1, ..., l_n \rightarrow l$,
– A fact l,
– An unconditional rule in the norm body $\langle X,\ t_i,\ t_f,\ \tau\rangle : \sim \mathscr{M}\ l$, or
– A conditional rule in the norm body $\langle X,\ t_i,\ t_f,\ \tau\rangle : l_1, ..., l_n \Rightarrow \sim \mathscr{M}\ l$,

where $l, l_1, ..., l_n$, with $n \geq 1$, are propositional literals that represent a state, an action, or an event; $\mathscr{M}$ is a deontic operator indicating an obligation O or a prohibition F; X is a string indicating the application group of the rule; t_i is a positive finite number indicating the start date of the rule; t_f is the expiration date of the rule, that is a positive finite number for a temporary rule or $+\infty$ for a non-temporary rule; and τ is a string indicating the territory the rule is applicable in. The operator $\sim$ represents either the empty string or the logical negation $\neg$.

As common in modal logic, the modals are dualized, in the specific case of deontic logic, $\neg Ol \Leftrightarrow \neg F \neg l$, and corresponding duals for $\neg O \neg l$ or $\neg F l$. There has been a long debate in the community of deontic logic about the form that the logic should have in terms of modal axioms, for different and complex reasons. Moreover, an explicit focus has been posed on the problem of *explicit permissions*. Usual representation of permits in deontic logic is based on the structural dualisation of prohibition: you have the permission P to do something when you have no duty to do its opposite $Pl \Leftrightarrow \neg O \neg l$. However, as shown in [74], explicit permissions are a more complex matter, that will be treated, for what regards changes, in further studies.

When a norm n changes, the set of the worlds that satisfy n, the *legal worlds*, and the set of the worlds that do not satisfy n, the *illegal worlds*, vary accordingly. In particular, if n is modified by introducing liberalizations or decriminalizations, the set of legal worlds grows and the set of illegal worlds shrinks because some of the illegal worlds become legal, while if n is modified by introducing restrictions, the set of illegal worlds grows and the set of legal worlds shrinks because some of the legal worlds become illegal. The situation is illustrated in Figures 6.1 and 6.2.

The legal effects of n can apply also in the past or in the future of its temporal validity. For this reason, whenever a norm n is changed, there may be a transition phase in which its old legal effects still hold. For example, if the norm change includes a restriction, and the action that is being restricted has a prolonged duration, the people involved in the restriction should be allowed some time to change their behaviour. Indeed, since decisions that were taken before the norm change are not contrary to the norm, they have to be protected. This is done by issuing a *transitory norm* that holds during the transition phase.

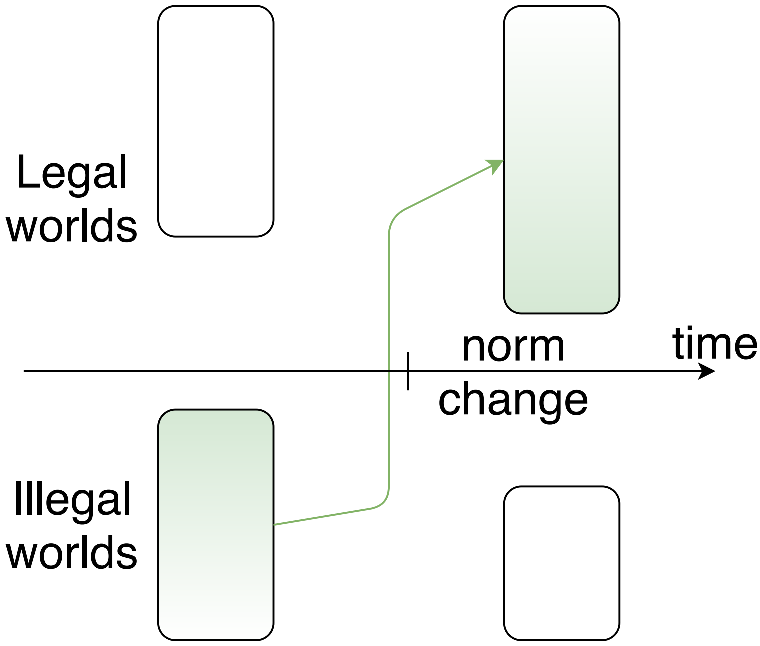

Figure 6.1: The effects of changing a norm by extension.

Typically, such a norm consists of one or more *reparation chains* of the form $Ol\ Ol'$, where O is the obligation operator and l, l' are propositional literals indicating states or actions. A reparation chain $Ol\ Ol'$ is read as "Ol' holds whenever Ol is not fulfilled".

6.2 Explicit norm change

We say that norm change is *explicit* when the actual code of a norm n is changed by the legislator.

Temporal change and misalignment of legal consequences We show here six operators that change the temporal validity of a rule in two different combined ways. The intuitive idea is to decrease or increase the start date t_i of the rule or its expiration date t_f. One limitation is that the start date t_i can be anticipated only if it is in the future. If the validity of a norm n has an *expiration date* attached to it, that is, $t_f \neq +\infty$, n is called a *temporary norm* (the problem of modifying temporary norms is addressed by Cristani et al. [45]), whilst in a wider scenario, the problem of abrogations and annulments have been addressed by Governatori et al. [78] from a technical viewpoint and further in details by Governatori and Rotolo [70]. A change to t_i may be an *anticipation*, when t_i is decreased, or a *postposition*, when t_i is increased. For a temporary norm n, two other kinds of temporal modifications, *anticipation* and *extension* of the expiration date t_f, may occur. Temporal anticipation of t_f consists in decreasing t_f, while temporal extension of t_f consists in increasing t_f. Moreover, for a norm n that is a *non-temporary norm*, i.e., $t_f = +\infty$, modifications to the start date t_i are done by the same operator that works on temporary norms, while changes to t_f are done by a specialized operator, that intuitively corresponds to annulments or abrogations.

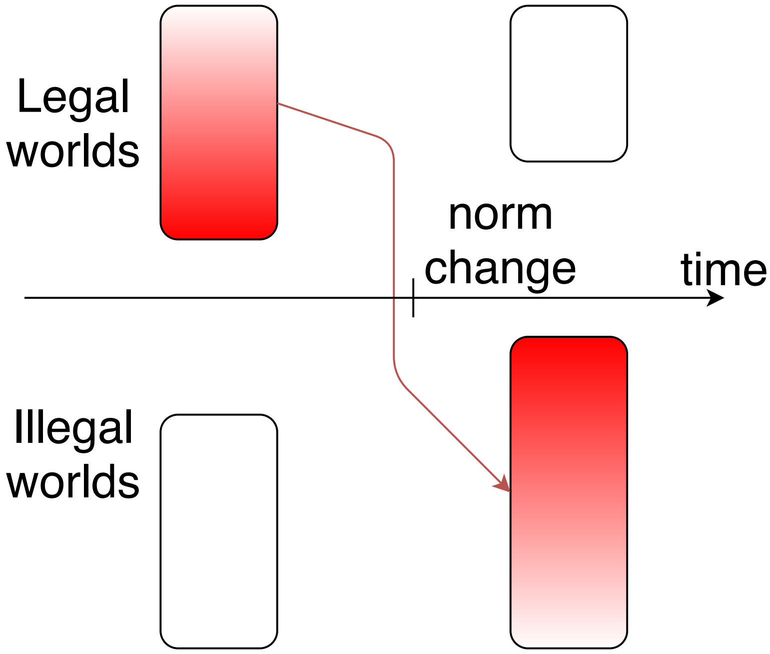

Figure 6.2: The effects of changing a norm by restriction.

These temporal changes for a labelled rule $\langle X, t_i, t_f, \tau\rangle : r$, where r may be conditional or unconditional, are defined through the following norm revision operators, with δ a finite positive number:

- Anticipation of the start date from t_i to $t_i' = t_i - \delta$, with $t_i > t_i'$. For real-world legal systems t_i' is typically in the future with respect to the issuing moment

$$\swarrow_{t_i,\delta} (\langle X, t_i, t_f, \tau\rangle : r) = \langle X, t_i - \delta, t_f, \tau\rangle : r$$

- Postposition of the start date from t_i to $t_i' = t_i + \delta$, with $t_i < t_i'$

$$\searrow_{t_i,\delta} (\langle X, t_i, t_f, \tau\rangle : r) = \langle X, t_i + \delta, t_f, \tau\rangle : r$$

- Anticipation of the expiration date from t_f to $t_f' = t_f - \delta$, with $t_f > t_f'$ and $t_f \neq +\infty$

$$\swarrow_{t_f,\delta} (\langle X, t_i, t_f, \tau\rangle : r) = \langle X, t_i, t_f - \delta, \tau\rangle : r$$

- Anticipation of the expiration date from $t_f = +\infty$ to $t_f' = t_i + \delta$, with $t_f' \neq +\infty$

$$\swarrow_{+\infty,\delta} (\langle X, t_i, t_f, \tau\rangle : r) = \langle X, t_i, t_i + \delta, \tau\rangle : r$$

- Extension of the expiration date from t_f to $t_f' = t_f + \delta$, with $t_f < t_f'$ and $t_f' \neq +\infty$

$$\searrow_{t_f,\delta} (\langle X, t_i, t_f, \tau\rangle : r) = \langle X, t_i, t_f + \delta, \tau\rangle : r$$

- Extension of the expiration date from $t_f \neq +\infty$ to $t_f' = +\infty$

$$\searrow_{t_f,+\infty} (\langle X, t_i, t_f, \tau\rangle : r) = \langle X, t_i, +\infty, \tau\rangle : r$$

The operators are six, and not eight, because we assume that $t_i > 0$ always has to hold in our framework. In the current doctrinal language postposition (and in some cases also anticipation) is named *derogation*. In this chapter we employ a more *agentive* term set for the reason that we mean to focus upon the *taxonomies* much more than the legal structures. The main difference among the revision operators is therefore defined on the way in which they modify the norms. The admissibility in the legal system is treated separately.

Another peculiarity to be aware of is that after a norm n is *issued*, there can be a temporal interval before n is enacted and becomes *valid*, where the temporal validity of n is defined as the interval in which n can be applied. In this chapter we do not analyze effects on the modifiability of the issuing and validity interval, focusing on the modification of temporal validity so far.

We employ t as the issue time instant, t_i as the initial validity time instant and t_f as the final validity time instant. Operators can move either the initial validity time instant forward or backward, and the final validity time instant forward or backward, but cannot move the issue time instant t, as that instant is a fact. There are systems subject to the principle of *natural issue* where it is not possible to provide issue of a norm in the past (lex posterior derogat priori). However, in many systems, at least for certain matters, and in nature of certain operations, it is possible (for instance when a norm is issued in a retroactive form in order to substantiate some principles expressed in other norms, or when some form of discontinuity in the normative system is implemented, in presence, for instance, of the suppression of some sort of previous regime). Therefore, we assume that the original ordering of the three values is free, provided that $t_i < t_f$ is assumed, thereby instituting three ontological classes of norms in terms of temporal issue:

- *Future temporary regulation (FTR)* $t \leq t_i < t_f$;
- *Past temporary repair (PTR)* $t_i < t < t_f$
- *Past temporary norm (PTN)* $t_i < t_f \leq t$.

Legal systems admitting past norms are quite unusual, and generally considered *unfair* as they attribute guilt to behaviours that have not been regulated before. However, there are cases in which past repairs have been introduced, often in the form of regular permanent norms, establishing that something that was *unregulated* in the past is now regulated. The pre-existing norm is issued for foundational principles, as they forbid behaviours that have not been considered in the past. The most commonly known example is the introduction *in judicio* of the notion of *war crime* during the Nuremberg Trials or the Tokyo Trial. Notice that when a norm is *permanent* the types of norms are reduced to two (*Future regulations (FR)* and *Past repairs (PR)*).

A summary of the ontological configurations that are possible in this temporal framework is presented in Table 6.1, where t' is the *reform time*, the time in which the norm is changed. As intuition suggests, we assume that the reform time is after the issue time, that is, $t < t'$. By using one of the four operators defined above we can also substantiate *ontological changes to a norm*, and an overview of the ontological modifications that are provided by the temporal operators is presented by the graphs in Figures 6.3, 6.4, and 6.5. For instance, Figure 6.3 shows the transitions on temporal configurations after the application of the anticipation of the start date operator $\swarrow_{t_i,\delta}$ or the postposition of the start date operator $\searrow_{t_i,\delta}$. Notice how the graph exemplifies that the two operators $\swarrow_{t_i,\delta}$ and $\searrow_{t_i,\delta}$ are symmetric, as the corresponding directed edges are symmetric. In an analogous

CONFIGURATION NAME	CONFIGURATION	CONFIGURATION CLASS
T_1	$t \leq t' \leq t_i < t_f$	FTR
T_2	$t \leq t_i \leq t' \leq t_f$	FTR
T_3	$t \leq t_i < t_f \leq t'$	FTR
T_4	$t_i \leq t < t' \leq t_f$	PTR
T_5	$t_i \leq t \leq t_f \leq t'$	PTR
T_6	$t_i < t_f \leq t < t'$	PTN
P_1	$t < t' \leq t_i$	FR
P_2	$t \leq t_i \leq t'$	FR
P_3	$t_i \leq t < t'$	PR

Table 6.1: The nine possible temporal configurations.

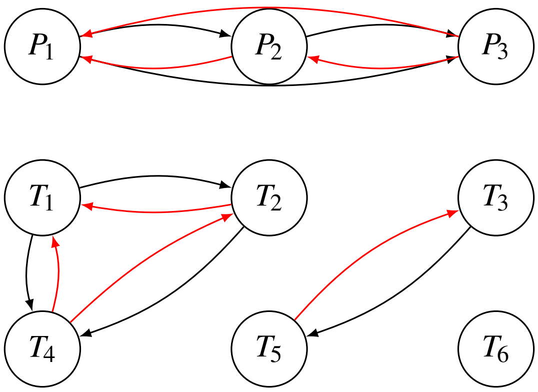

Figure 6.3: The transitions on configurations after applying the anticipation of the start date operator $\swarrow_{t_i,\delta}$ or the postposition of the start date operator $\searrow_{t_i,\delta}$.

manner, the graphs in Figures 6.4 and 6.5 show that also the pairs $\swarrow_{t_f,\delta}$ and $\searrow_{t_f,\delta}$ and $\swarrow_{+\infty,\delta}$ and $\searrow_{t_f,+\infty}$ are symmetric.

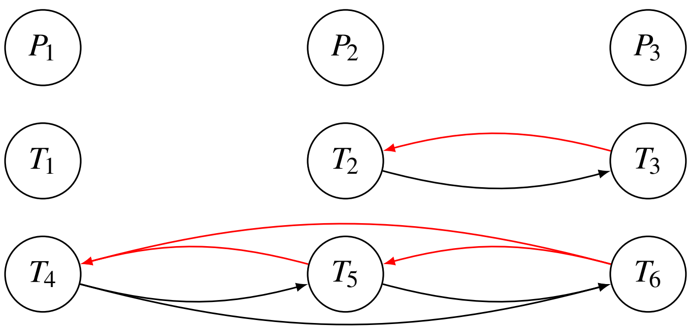

Figure 6.4: The transitions on configurations after applying the anticipation of the expiration date operator $\swarrow_{t_f,\delta}$ or the extension of the expiration date operator $\searrow_{t_f,\delta}$.

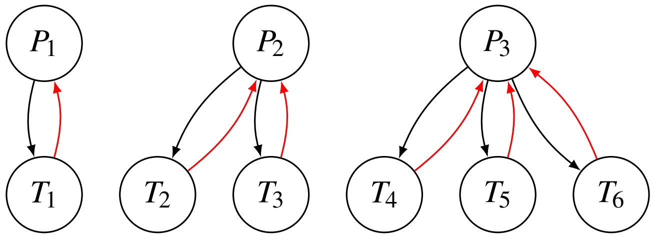

Figure 6.5: The transitions on configurations after applying the anticipation of the expiration date operator $\swarrow_{+\infty,\delta}$ or the extension of the expiration date operator $\searrow_{t_f,+\infty}$, that is, the operators that change a finite t_f to $+\infty$, and vice-versa.

Given the three ontological classes and the temporal configurations and having shown how the operators make the configurations transition between each other, we can give the definition of a *pure operator*.

Definition 14 (Pure operator). A norm revision operator OP is a pure operator if either

- for every labelled rule $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$ of a given ontological class C, when OP is applied to $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$, the resulting labelled rule $OP(\langle X,\ t_i,\ t_f,\ \tau\rangle : r)$ always has ontological class C', with $C' \neq C$, or
- for every labelled rule $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$ of a given ontological class C, when OP is applied to $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$, the resulting labelled rule $OP(\langle X,\ t_i,\ t_f,\ \tau\rangle : r)$ always has ontological class C.

In Examples 7, 8 and 9 we analyze whether the operators presented thus far are pure operators or not.

Example 7. Consider the operators $\swarrow_{t_i,\delta}$ and $\searrow_{t_i,\delta}$. We have that for a rule s in configuration P_1, i.e., with class FR, $\swarrow_{t_f,\delta}(s)$ may be either in configuration P_2, i.e., with class FR, or in configuration P_3, i.e., with class PR. Therefore, the operator $\swarrow_{t_i,\delta}$ is not pure. Indeed, the operator $\searrow_{t_i,\delta}$ is not pure either because it is symmetric to $\swarrow_{t_i,\delta}$.

Example 8. Consider the operators $\swarrow_{t_f,\delta}$ and $\searrow_{t_f,\delta}$. We have that for a rule s in configuration T_4, i.e., with class PTR, $\swarrow_{t_f,\delta}(s)$ may be either in configuration T_5, i.e., with class PTR, or in configuration T_6, i.e., with class PTN. Therefore, the operator $\swarrow_{t_f,\delta}$ is not pure. Indeed, the operator $\searrow_{t_f,\delta}$ is not pure either because it is symmetric to $\swarrow_{t_f,\delta}$.

Example 9. Consider the operators $\swarrow_{+\infty,\delta}$ and $\searrow_{t_f,+\infty}$. These operators always change the ontological class of a norm, as they make a permanent norm temporary, and vice-versa. Therefore, these two operators are pure.

A norm n may be *removed* from a normative system. In general, n may be removed by *annulment* or *abrogation*. Annulment makes a norm n invalid by removing it from the legal system, and usually n is suppressed in a *retroactive* fashion, that is, n is not valid since a time prior to its suppression. Abrogation suppresses only the legal effects of a norm n, and typically it is not retroactive.

- EU member states
  - Austria
  - Belgium
  - Bulgaria
  - Croatia
  - Cyprus
  - Czechia
  - Denmark
  - Estonia
  - Finland
  - France
  - Germany
  - Greece
  - Hungary

- EU member states
  - Italy
    - Abruzzo
    - Aosta Valley
    - Apulia
    - Basilicata
    - Calabria
    - Campania
    - Emilia-Romagna
    - Friuli-Venezia Giulia
    - Lazio
    - Liguria
    - Lombardy
    - Marche
    - Molise
    - Piedmont
    - Sardinia
    - Sicily
    - Trentino-Alto Adige
    - Tuscany
    - Umbria
    - Veneto

- EU member states
  - Latvia
  - Lithuania
  - Luxembourg
  - Malta
  - Netherlands
  - Poland
  - Portugal
  - Romania
  - Slovakia
  - Slovenia
  - Spain
  - Sweden

Figure 6.6: The classification of the 27 states that are members of the European Union (EU), with a focus on Italy and its 20 regions.

Territorial change A rule $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$ of a norm n is valid only in the specified territory τ. The territory τ is classified as an element in a hierarchy of territories, where some territories may include τ (the *superterritories* of τ), and other territories may be included in τ (the *subterritories* of τ). For instance, see Figure 6.6, which shows a classification of the states that are members of the European Union. The legislator may change the rule $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$ by making it valid in a different territory τ', that may be a superterritory of τ or a subterritory of τ. We give here two operators for this purpose:

– Extension of the territory of validity from τ to one of its superterritories Λ

$$\uparrow_{\tau,\Lambda} (\langle X,\ t_i,\ t_f,\ \tau\rangle : r) = \langle X,\ t_i,\ t_f,\ \Lambda\rangle : r$$

– Shrinkage of the territory of validity from τ to one of its subterritories V

$$\downarrow_{\tau,V} (\langle X,\ t_i,\ t_f,\ \tau\rangle : r) = \langle X,\ t_i,\ t_f,\ V\rangle : r$$

Change in application topic A norm n may regulate a specific action or topic t, for instance $t = fishing$. The legislator may change n by expanding t to t', for instance $t' = fishing\ or\ hunting$; by restricting t to t', for instance $t' = carp\ fishing$; or by changing t to t', for instance $t' = hunting$.

```
STRUCTURE OF THE INTERNATIONAL STANDARD CLASSIFICATION OF
OCCUPATIONS (ISCO-08)
 |
 |__1 Managers
 |   |__11 Chief Executives, Senior Officials and Legislators
 |   |   |__111 Legislators and Senior Officials
 |   |   |   |__1111 Legislators
 |   |   |   |__1112 Senior Government Officials
 |   |   |   |__1113 Traditional Chiefs and Heads of Villages
 |   |   |   |__1114 Senior Officials of Special-interest Organizations
 |   |   |__112 Managing Directors and Chief Executives
 |   |__12 Administrative and Commercial Managers
 |   |__13 Production and Specialized Services Managers
 |   |__14 Hospitality, Retail and Other Services Managers
 |__2 Professionals
 |__3 Technicians and Associate Professionals
 |__4 Clerical Support Workers
 |__5 Services and Sales Workers
 |__6 Skilled Agricultural, Forestry and Fishery Workers
 |__7 Craft and Related Trades Workers
 |__8 Plant and Machine Operators and Assemblers
 |__9 Elementary Occupations
 |__0 Armed Forces Occupations
```

Figure 6.7: The ISCO-08 classification of occupational titles, with a focus on the unit group of Legislators and Senior Officials.

Change in application group A rule $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$ of a norm n is valid only for the specified group X of people. The group X is classified as an element in a hierarchy of groups, where some groups may include X (the *supergroups* of X), and other groups may be included in X (the *subgroups* of X). For instance, see Figure 6.7, which shows how different professional groups are classified by the International Labour Organization (ILO) in the International Standard Classification of Occupations (ISCO-08). Notice that a classification based on professional categories is just one among the many ways to classify groups of people, other examples may be classifications by marital status or by native language. The legislator may change the rule $\langle X,\ t_i,\ t_f,\ \tau\rangle : r$ by making it valid for a different group X', that may be a supergroup of X or a subgroup of X. We give here two operators for this purpose:

– Extension of the application group from X to one of its supergroups Δ

$$\Uparrow_{X,\Delta} (\langle X,\ t_i,\ t_f,\ \tau\rangle : r) = \langle \Delta,\ t_i,\ t_f,\ \tau\rangle : r$$

– Restriction of the application group from X to one of its subgroups ∇

$$\Downarrow_{X,\nabla} (\langle X,\ t_i,\ t_f,\ \tau\rangle : r) = \langle \nabla,\ t_i,\ t_f,\ \tau\rangle : r$$

An action that is usually forbidden, but permitted to a group X of people, is called a *privilege* for X. An action that is usually obligatory, but not obligatory for a group X of people, is called an *exemption* for X. An obligation or a prohibition that is applied only to a group X of people is called a *proper obligation* for X or a *proper prohibition* for X, respectively.

Change in definitions The legislator may change a norm n by changing the definitions that are adopted by n. Given a definition represented by a strict rule $l_1,...,l_n \to l$, the definition may be *strengthened*, as in $l_1,...,l_{n-1} \to l$, *weakened*, as in $l_1,...,l_n,l_{n+1} \to l$, or *removed* from the normative system.

Change in applicability conditions The legislator may change a conditional rule $\langle X, t_i, t_f, \tau\rangle : l_1,...,l_n \Rightarrow \sim \mathscr{M}\ l$ by changing the conditions that make it applicable. The rule may be *strengthened*, as in $\langle X, t_i, t_f, \tau\rangle : l_1,...,l_{n-1} \Rightarrow \sim \mathscr{M}\ l$, or *weakened*, as in $\langle X, t_i, t_f, \tau\rangle : l_1,...,l_n,l_{n+1} \Rightarrow \sim \mathscr{M}\ l$.

6.3 Implicit norm change

We say that norm change is *implicit* when the actual code of a norm n itself has not changed, but its effects have changed because its context has changed: the region of space or the group of people that n can be applied to have changed independently of it. Norm n may also change because the semantics of legal terminology has changed, for instance because there has been a shift in how some topics are perceived by the public.

Territorial change A labelled rule $\langle X, t_i, t_f, \tau\rangle : r$ of a norm n is subjected to an implicit territorial change when the region of space τ that it refers to has changed. Spatial boundaries may be *bona fide* boundaries or *fiat* boundaries. Bona fide boundaries are marked by physical features of the area they constrain, while fiat boundaries are induced through human demarcation.

A rule r that is valid only in a specific region τ of space changes its effects when τ changes. Bona fide boundaries may change because of a change in the physical features of the area, or they may be converted to fiat boundaries. Fiat boundaries of a region τ of space may be changed in order to capture a different region τ' of space, or they may be converted to bona fide boundaries.

Change in application group A labelled rule $\langle X, t_i, t_f, \tau\rangle : r$ of a norm n is applicable only to a specific group X of people and implicitly changes its effects whenever X changes independently of r. This may happen because an external norm that defines group X has changed, or because the perception of what people constitute group X has changed in society.

Change in semantics of legal terminology Whenever the meaning of legal terminology adopted in norm n changes, the effects of n change accordingly. This implicit change may happen because of a modification in other norms that give the legal terminology, similarly to an explicit change in the definitions of the norm, or because society has changed its perception of the meaning of some words. For example, the Eighth Amendment to the United States Constitution states that a *cruel and unusual* punishment should not be inflicted, but what makes a punishment cruel and unusual is something that gets revised as acceptance in society evolves.

Change in the norms hierarchy Each norm n is part of a norms hierarchy, where norms are ordered by the body of law that issued them and by application topic. The *lex superior* doctrine states that, when in conflict, a law issued by a body of law of greater legitimacy overrides a law issued by an inferior body of law. The *lex specialis* doctrine states that, when in conflict, a law governing a specific topic overrides a law that governs a more general subject. Therefore, the effects of norm n may implicitly change also because a norm that overrides n in the hierarchy has been added, changed or removed from the normative system.

Chapter 7

Acquisition and Learning of Principles and Norms

7.1 Basic Principles and Issues

Consider an agent acting in a hybrid society, i.e., a society where both natural and artificial components play an important role. The norms that govern the behaviour of such an agent can be either ethical or legal, and in this context new regulatory challenges emerge and need to be investigated. While an agent may follow some rules that stem from its internal state of mind, many rules are externally imposed on it.

The rules of the first kind, the internal ones, can be described by the *rule of justice* principle, while the external rules can be described by the *rule of law* principle. The rule of justice principle establishes that on a general basis the internal rules prevail against the external ones. The rule of law warrants the opposite.

These two principles can clash, that is, a rule of one kind may be conflicting with a rule of the other kind. When such situations arise, assuming that we are representing the normative system by using defeasible deontic logic [108], a superiority relation on rules can be used to solve the conflict and determine which is the rule to be applied.

There exist however many cases in which an agent itself has to formulate a new normative principle, and possibly solve a conflict that arises per se when the normative principle clashes with other existing ones (see [77] for issues regarding the introduction of new preferences in nonmontonic reasoning frameworks). Consider, for instance, the case in which an artificial agent has yet no way to manage a new token of information that appears out of nowhere during its acting. In particular, imagine a driverless car that is passing the boundary of a right-driving country towards a left-driving country (for instance when passing towards the channel tunnel coming from France to the UK). Many of the ethical rules the agent has been used with at that stage have to be changed with other rules the agent may have not yet incorporated for it has been designed for right-driving countries. What would be the solution of this? One possibility would be that the agent acquires the new rules by an internet connection, and establishes the correct application of the rules (namely by devising a correct interpretation of the situation as provisional).

Now consider the case of an industrial robot that is operating in an unsupervised way within a factory. One worker, alone, enters the room and suddenly he fell on the floor, because of a heart attack. The robot has not been provided with a specific rule for this kind of emergency, but it has an ethical apparatus that is specified to work with novel situations by analogy with other situations. The robot has recorded a situation in which humans have to act in a similar context and they have proceeded by calling an emergency number. By analogy the agent computes a new rule that says: when an intelligent agent sees an emergency situation, she call an emergency number. Here, the analogy is obtained by substituting intelligent agent for human.

Both the examples shown above require the ability to introduce *new* rules in the system, and possibly solve contradictions arising that way. Defeasible deontic logic can accommodate the solution to the emerging conflict when we introduce new *specific* revision operators that act on the base of actions to be carried out to match situations that are not covered by the existing rules.

In this work we focus on the problem of an agent generating (by acquisition or learning) new legal or ethical rules in the language of DDL, and on the effect that this has on the agent's normative system.

Throughout the Chapter we use the basic definitions of Defeasible Deontic Logic (DDL) introduced in Section 2.1.2 and the formalism structure for norms in DDL introduced in Section 6.1 and Definition 13. In this Chapter we focus on rules in the norm body.

The rest of the Chapter is organized as follows. Section 7.2 is devoted to provide a general framework for acquisition of new norms from external sources and provide room for the different concepts of acquisition and learning. In Section 7.3 we discuss two revision operators able to modify an ethical/normative system and show how to accommodate the types of processes discussed in Section 7.2 with these operators.

7.2 Norm Acquisition and Learning

The new rule formulated by the agent may be acquired from the outside or learned by analogy or homology. The process can be one of the kinds described below.

External Acquisition The rule is acquired from the outside and is incorporated into the agent's normative system. An agent may incorporate an external rule because it is legally bound to do so (for instance because the institution that issued the rule has power over the agent), or because it trusts the issuing institution or the content of the rule. In the first case, we say that the rule is *externally imposed*, while in the second case the acquisition is voluntary and is based on the concept of *trustability*. The revision process provoked by this method can issue negotiation strategies, that we take into consideration in further work.

The incorporation has to be made in relation with the rules that are already in the system, and the existing superiority relation may have to be enriched in case of conflicts. Typically, the externally-imposed rules are legal rules, the trusted rules are ethico-moral rules, the already existing rules can be of either kind, and the agent may use the principles of rule of law or rule of justice to solve emerging conflicts. If the conflict is between two rules of the same kind (legal or ethico-moral), the agent has to use other principles to define the superiority, for instance the principles of *lex superior derogat inferiori*, *lex specialis derogat generali*, or *lex posterior derogat priori* can be adopted for legal rules.

Learning by Analogy: Abstraction The rule is learned after observing a situation that is not regulated but is similar to a situation where an existing rule applies. The learned rule is formulated by analogy to the existing rule. For instance, consider an existing rule $a,\ b \Rightarrow c$ and a rule to be learned $a',\ b' \Rightarrow c'$: one rule can be obtained from the other by analogy if there is an abstraction function α such that $\alpha(a) = \alpha(a')$, $\alpha(b) = \alpha(b')$, and $\alpha(c) = \alpha(c')$.

Learning by Homology: Generalization The rule is learned after observing a situation that is not regulated but is a generalization of a situation where an existing rule applies, and also the rule can be generalized. For instance, consider an existing rule $a,\ b \Rightarrow c$ and a rule to be learned $a \Rightarrow c'$: the second rule can be learned if c is a logical consequence of c', that is, $c' \models c$.

Learning by Homology: Specialization The rule is learned after observing a situation that is regulated only in a general form, and a more specific rule can be formulated. For instance, consider an existing rule $a \Rightarrow c$ and a rule to be learned $a,\ b \Rightarrow c'$: the second rule can be learned if c' is a logical consequence of c, that is, $c \models c'$.

Learning by Averaging The rule is learned after observing a situation for which there are a general rule and a rule that is too specific to apply. The learned rule is formulated by "averaging" these two existing rules. For instance, consider existing rules $a \Rightarrow c$ and $a,\ b,\ d \Rightarrow c''$: the rule to be learned $a,\ b \Rightarrow c'$ can be obtained if c is a logical consequence of c' and c' is a logical consequence of c'', that is, $c' \models c$ and $c'' \models c'$.

The incorporation processes depend on the legal context, and we can exemplify this by the process of legislative acquisition and incorporation of GDPR as an EU *regulation* in a national legislation.

Example 10. The GDPR has been issued as an EU law that unifies the regulation on data protection and privacy within the EU. Since it is a regulation and not a directive, it is directly applicable in EU states, that is, in our framework, the GDPR rules are externally acquired in the national normative systems. Moreover, the GDPR provides some flexibility for the norms to be adjusted by individual member states, and it is used as a model for many national laws outside EU (a phenomenon known as the Brussels effect), therefore it is also an example of a combination of external acquisition and norm learning.

On the other hand, let us consider the following specific application in healthcare and robot ethics.

Example 11. A nurse robot is active in a hospital to treat patients. Suppose it is of paramount importance that the robot can provide a specific treatment drug d, among others. An operator may choose to add an ethic rule for the robot to stop everything it is doing if it runs out of the drug in question, and immediately start to refill its supply. Based on this data, we may be interested for the robot to automatically learn a new rule, analogous to the one written by the operator, for a treatment drug that is at least as important as d.

See Figure 7.1 for External Acquisition, Figure 7.2 for Learning by Analogy: Abstraction, Figure 7.3 for Learning by Homology: Generalization and Specialization, and Figure 7.4 for Learning by Averaging, that are the processes that are formalized below.

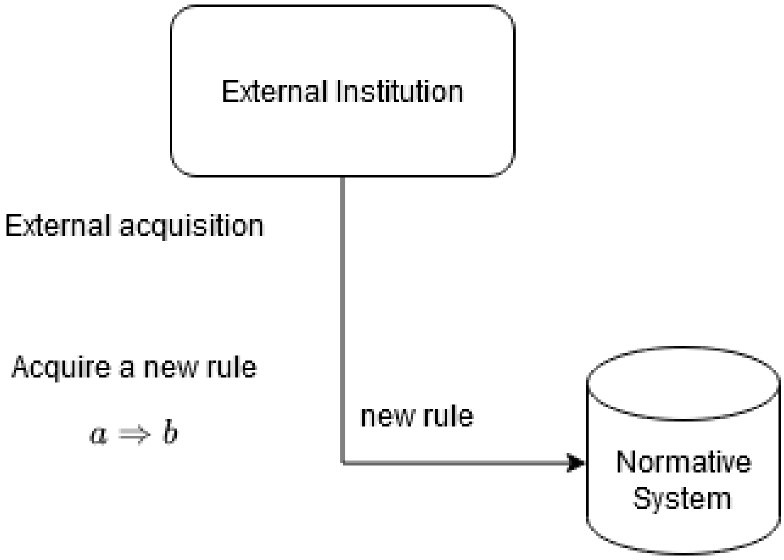

Figure 7.1: External Acquisition.

7.3 Revision Operators for Norm Acquisition and Learning

In this Section we formalize some of the processes described above with the introduction of a revision operator that expands a theory D, composed of norms as in Definition 13, with new information to obtain theory D', where new norms are learned or acquired.

Definition 15 (Norm creation operator). Given a defeasible theory D, the *norm creation operator NC* is a revision operator that can be used to obtain a new theory $D' = NC(D, H, I)$, where

- the set $H = \{h_1, ..., h_t\}$ contains rules that D is expanded with, and
- the set $I = \{i_1, ..., i_u\}$ contains pieces of information that are used to create new rules in D.

In particular, an element of set I can be of two kinds:

- a redundancy $\mathscr{R}(l,\ r)$ of literal l for a rule r, or
- a similarity $\mathscr{S}(l_1,\ l_2,\ r)$ between literals l_1 and l_2 for a rule r.

The theory D is expanded with rules in H by the usual expansion operation known in belief revision. On the other hand, the processing of set I is introduced for the first time in this thesis: the idea is that a literal l such that $\mathscr{R}(l,\ r)$ can be removed from the premises $A(r)$ of rule r, while literals l_1 and l_2 such that $\mathscr{S}(l_1,\ l_2,\ r)$ can be replaced one for the other in $A(r)$.

Given theory $D = (F, R, >)$, we formalize the process to obtain $D' = (F', R', >')$ for each of the operations described above.

- *External Acquisition* $NC(D, H, \emptyset) = (F, R', >') = D'$, where $R' = R \cup H$ and $>'$ is $>$ enriched to handle conflicts.

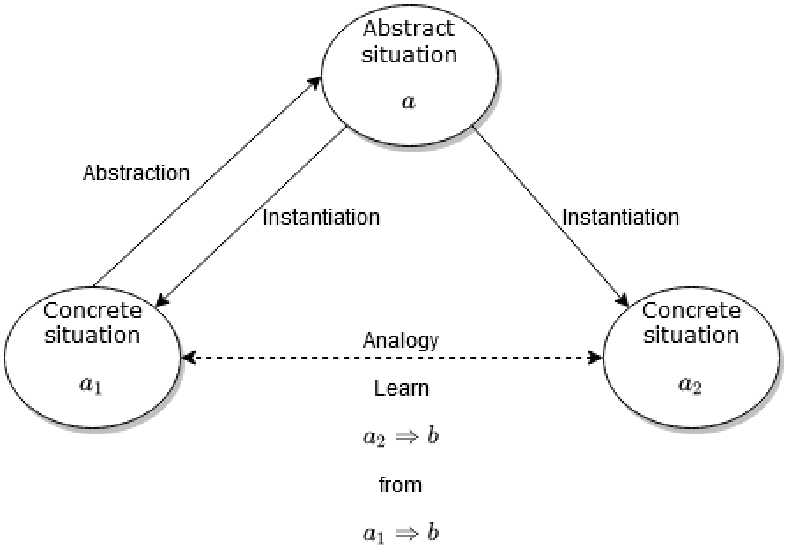

Figure 7.2: Learning by Analogy: Abstraction.

- *Learning by Analogy: Abstraction* $NC(D,\emptyset,I) = (F,R',>') = D'$, where $R' = R \cup G$ and $>'$ is an enrichment of $>$. The set G is obtained in this way: for each similarity $\mathscr{S}(l_1,\ l_2,\ r)$ between literals l_1 and l_2 for a rule r, a rule r', corresponding to r with l_1 and l_2 swapped when they appear in the premises $A(r)$, is added to G.
- *Learning by Homology: Generalization* $NC(D,\emptyset,I) = (F,R',>') = D'$, where $R' = R \cup G$ and $>'$ is an enrichment of $>$. The set G is obtained in this way: for each redundancy $\mathscr{R}(l,\ r)$ of literal l for a rule r, a rule r', corresponding to r with l removed when it appears in the premises $A(r)$, is added to G.
- *Learning by Homology: Specialization* $NC(D,\emptyset,I) = (F,R',>') = D'$, where $R' = R \cup G$ and $>'$ is an enrichment of $>$. The set G is obtained in this way: for each redundancy $\mathscr{R}(l,\ r)$ of literal l for a rule r, a rule r', corresponding to r with l added when it does not appear in the premises $A(r)$, is added to G.
- *Learning by Averaging* $NC(D,\emptyset,I) = (F,R',>') = D'$, where $R' = R \cup G$ and $>'$ is an enrichment of $>$. The set G is obtained in this way: given two rules $r,\ r'' \in R$, with $A(r) \subset A(r'')$ and $C(r'') \models C(r)$, add to G a rule r' such that $A(r) \subset A(r')$, $A(r') \subset A(r'')$, $C(r'') \models C(r')$ and $C(r') \models C(r)$. Moreover, each literal l such that $l \in A(r'')$ and $l \notin A(r')$ has to be a redundancy $\mathscr{R}(l,\ r'')$ for r''.

A part the external acquisition revision operator, the other four methods are *observational*. In other terms, we can see them as *base revision* operators with an inductive reasoning mechanics.

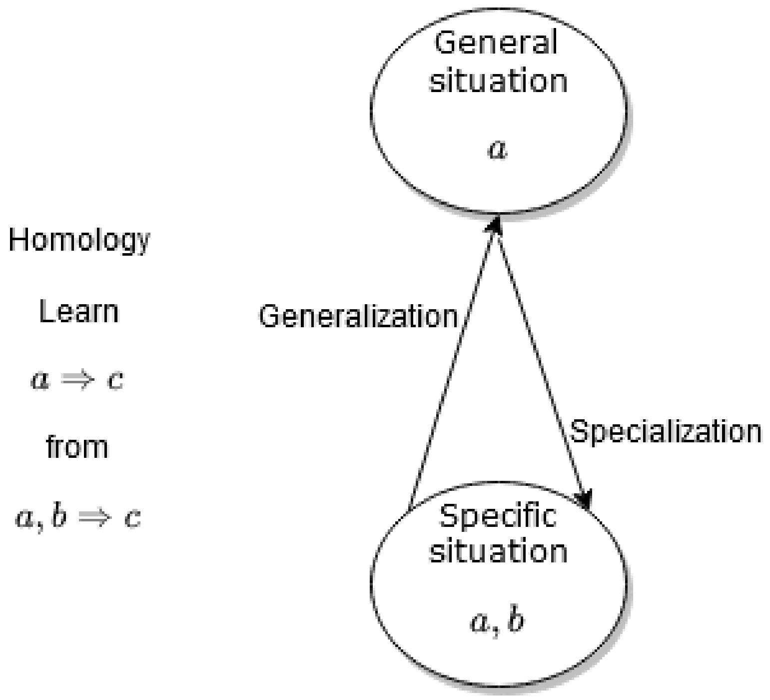

Figure 7.3: Learning by Homology: Generalization and Specialization.

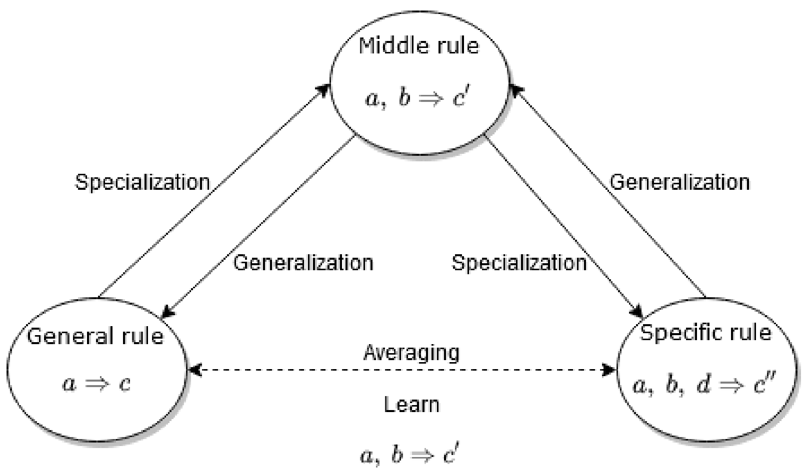

Figure 7.4: Learning by Averaging.

Chapter 8

Measuring Pledges in Defeasible Logic

In this Chapter we describe how pledges between institutions can be encoded in defeasible logic and we frame the problem of measuring the advancement of a pledge as a revision problem. For our regards, we consider a pledge to be a voluntary commitment, or a promise, that is made between organizations or institutions. The fulfillment of such a promise is not required by legislation and is guaranteed only by the faith in the other organization. Typically, actions that go towards the realization of the pledge are measurable but their effects may not be immediate. The goal that is to be realized by the pledge may be long-term and it might require also an effort by other subjects. An organization can either pledge on its own or together with another organization. Examples of pledges are those made to the European Union institutions:

– the *product safety pledge*, that is a commitment of online marketplaces to keep online shoppers safe from harmful products
– the *EU pledge*, that is a commitment of food and beverage companies to change the way they advertise to children in order to support parents in making the right diet and lifestyle choices for their children,
– pledge made in the context of the *European Climate Pact* in order to contribute to fighting climate change.

The hypotheses of this study are that pledges can be viewed as *intentions* and that the validity of a pledge can be measured through a *measure of advancement* on a theory representing the pledge. In the two sections of this Chapter we describe two different approaches that in the future we aim at combining together. With the first approach we give a theoretical notion of what it means when a defeasible theory can achieve a pledge and we describe some problems that we deem of interest and that we aim at tackling in the future. The notions given in the first section can be used to create a framework to compute the advancement of pledges. In the second section we treat the problem of revising a defeasible theory by only changing its facts. The motivation for studying this problem is that the problems of checking the possibility and the advancement of a pledge can be injected in the fact revision problem.

8.1 Achieving Pledges in Defeasible Logic

We start by identifying two sets:

Definition 16. Given a defeasible theory S, a *candidate set* C of S is a subset of the propositional variables appearing in S, while a *goal set* G of S is a subset of the literals appearing in S.

The notion of *potential achievement* that we employ is introduced in Definition 17:

Definition 17. Given a defeasible theory S, a *candidate set* C of S, and a *goal set* G of S, S *potentially achieves* G when it is possible to conclude every literal $L \in G$ by adding a subset A of C to S, with $G \cap C = \emptyset$.

In the following, we simply say that S *achieves* goal set G instead of *potentially achieves* when no confusion arises. Moreover, by abusing the terminology, when the goal set G is a singleton $G = \{L\}$, we say that S achieves literal L instead of set G. The idea is that S *achieves* L when it concludes L while assuming other literals, and without assuming L or the premises of one rule used to derive it. A candidate set is *plausible* for a given defeasible theory when injecting the literals in the theory does not generate a contradiction in the strict part of the system. The set A added to S to conclude L is called an *achievement set*. An achievement set is *minimal* when no subset of it is an achievement set. An achievement set is *maximal* when no superset of it is an achievement set. The set of all minimal achievement sets for a system S and goal set G is named the *minimal configuration set* and denoted by $mcs(G,S)$. The set of all maximal achievement sets for a system S and goal set G is named the *maximal configuration set* and denoted by $MCS(G,S)$.

Example 12. Consider the defeasible theory S below:

$$\begin{array}{rcl} b, \neg e & \Rightarrow & c \\ a, \neg b & \Rightarrow & \neg c \\ a & \Rightarrow & b \\ b & \Rightarrow & \neg e \end{array} \qquad \begin{array}{rcl} \neg a, b & \Rightarrow & \neg c \\ \neg a, \neg e & \Rightarrow & b \\ a, c & \Rightarrow & b \\ b, c & \Rightarrow & d \end{array}$$

	Achiev.	Minimal	a	$\neg a$	b	c	d	$\neg e$
c_1	Yes	Yes	•					
c_2	No			•				
c_3	Yes	Yes			•			
c_4	No							•
c_5	Yes	No	•					•
c_6	Yes	Yes		•				•
c_7	Yes	Yes					•	
c_8	Yes	No	•			•		

Since there is the only one rule able to derive d, and its premises are b and c, a candidate set for d can be obtained only by ensuring that it does not contain b, c and d. Literals whose letters appear in S, excluding the above, are a, $\neg a$, $\neg b$, $\neg c$, e, $\neg e$. Obviously, we cannot assume $\neg b$ or $\neg c$ for this will certainly lead not to conclude b and c, making therefore the obtained subset of literals not plausible. Moreover, the assumptions must be consistent, that is, we cannot assume a and $\neg a$ (or e and $\neg e$) simultaneously. Again, we cannot assume e in any configuration (namely $\{e\}, \{a,e\}, \{\neg a, e\}$) because the only rule with consequent c has $\neg e$ as a premise, and therefore assuming e would block the derivation of c. The table above on the right represents 7 plausible configurations, some achieving d and some not, and among those achieving d some minimal and some not.

Given a defeasible theory S, a *candidate set* C of S, and a *goal set* G of S, the reasoning problems that we find of interest are:

1. Decide whether there exists an achievement set for G in S.
2. Find a minimal (or maximal) achievement set for G in S.
3. Find the set of all minimal (or maximal) achievement sets $mcs(G,S)$ (or $MCS(G,S)$) for G in S.
4. If there is no achievement set for G in S, find a maximal subset of G for which there is an achievement set in S.

8.2 The Facts Changing Problem

The problems

- of determining whether a desired consequence of a specific norm is derived under general conditions, and
- of determining whether an undesired consequence can be avoided

are typical speculations *de iure condendo* (that is, reflections that are made while elaborating new norms) for law and policy makers. In this Chapter we analyze the behavior of non monotonic logic with respect to the proposed scenarios, and show that the problem of computing the effectiveness of norms, as devised above, is computationally hard. We also show that, under some limitations, the problem becomes tractable in a polynomial fashion, and some of these cases indeed correspond to the configurations where the problem intuitively appears simpler.

8.2.1 Fact Revision Complexity

We analyse the problem of revising a defeasible theory by only changing its facts. We start by focusing on the corresponding decision problem: is it possible to modify the extension of a defeasible theory by only changing the set of facts? We call such a problem the *Facts Changing Problem*, shortly *FC Problem*, and we now formalize it.

Definition 18 (FC Problem). Given a defeasible theory $D = (F,R)$, a literal p such that $D \vdash +\partial p$ ($D \vdash -\partial p$), and a set of revisable facts $C = \{f_1, \ldots, f_n\}$, with $n \geq 0$, is it possible to obtain $D' = (F',R)$ such that $D' \vdash -\partial p$ ($D' \vdash +\partial p$), where F' is obtained by adding or removing from F facts appearing in C?

In this section we show that the FC Problem is NP-complete.

Lemma 9. *The FC Problem is NP-hard.*

Proof. Consider the variant of the FC Problem in Definition 18 where given a defeasible theory $D = (F,R)$ and a literal p such that $D \vdash +\partial p$, we ask whether it is possible to obtain $D' = (F',R)$ such that $D' \vdash -\partial p$. To prove NP-hardness, consider an unsatisfiable set S of propositional clauses. The set S does not derive a literal p, written $S \not\vdash p$, if and only if $S \cup \{\neg p\}$ is satisfiable. The set S derives a literal p, written $S \vdash p$, if and only if $S \cup \{\neg p\}$ is unsatisfiable.

Consider the Propositional Revision Problem (PR Problem) of revising a propositional set S such that $S \vdash p$ to a propositional set S' such that $S' \not\vdash p$. Notice that the PR Problem can be solved by iteratively adding combinations of unit clauses to set S in order to get set S' and then checking whether $S' \not\vdash p$. Of course, it is not allowed to simply add the unit clause p to S. Since classical propositional logic is monotonic, the only way to derive something new is by adding constraints to the set of clauses. The problem of checking that $S' \not\vdash p$ is SAT, therefore the PR Problem is at least as hard as SAT. In order to make the problem not trivial, we define also for the propositional case a set C that contains a subset of all literals with atoms appearing in S. The unit clauses to be added to S have to appear in C.

We now build a reduction from the PR Problem for S to the FC Problem for a defeasible theory $D = (F, R)$ and set of revisable facts C. For simplicity, we consider a set S where there are no unit clauses. Therefore, in the reduction we leave the set F empty. The set C remains unchanged. The non-unit clauses of S are transformed in defeasible rules and put in R in the following manner, where we distinguish three kinds of clauses.

1. *Clauses where p appears*: these clauses are translated in rules where p is in the head and the other literals are negated and become antecedents.
2. *Clauses where $\neg p$ appears*: these clauses are translated in rules where $\neg p$ is in the head and the other literals are negated and become antecedents.
3. *Clauses where neither p nor $\neg p$ appears*: each one of these clauses is translated in a number of rules equal to the number of literals in the clause. In each one of these rules, one of the literals is in the head and the other literals are negated and become antecedents.

Notice that rules obtained from clauses of kind 2 will not be used when trying to derive p and therefore can be safely ignored. The idea to convert the results of the FC Problem to results of the PR Problem is that the facts in F' correspond to the unit clauses that can be used to get S' from S.

Let us show that the results of the reduction hold both when the answer to the FC Problem is *"yes"* and when it is *"no"*.

- If the FC Problem has a negative answer, the facts in F' are used to form a deductive chain that ends with $D' \vdash +\partial p$. The facts in F' are not contradictory because they are strict. Consider the propositional set U built with the unit clauses corresponding to the facts in F'. The set S' is obtained as $S' = S \cup U$, and we have that $S' \vdash p$ and that $S' \cup \{\neg p\}$ is unsatisfiable.
- On the other hand, if the FC Problem has a positive answer, there is no coherent set of facts F' such that $D' \vdash +\partial p$. Therefore, there is no way of adding a coherent set U of unit clauses to S to obtain S' such that $S' \vdash p$. In other words, $S' \not\vdash p$ and $S' \cup \{\neg p\}$ is satisfiable for every coherent set U of unit clauses such that $S' = S \cup U$.

Thus, this instance of the FC Problem can be used to answer the PR Problem for S, and therefore the FC Problem is NP-hard. □

Lemma 10. *The FC Problem is in NP.*

Proof. Consider the FC Problem as in Definition 18. Consider an oracle that selects which facts of C to remove from F and which to add to F, guessing F' in output. Then, checking that $D' = (F', R)$ is such that $D' \vdash -\partial p$ ($D' \vdash +\partial p$) is polynomial in R because computing the extension of a defeasible theory is polynomial. Thus, the FC Problem is in NP. □

We get Theorem 5 by Lemma 9 and Lemma 10:

Theorem 5. *The FC Problem is NP-complete.*

We give a couple of examples of the reduction in order to make it more clear.

Example 13. Consider the set of propositional clauses $S = \{(1)\ a \vee b \vee c \vee p,\ (2)\ \neg a \vee c,\ (3)\ \neg b \vee c\}$. S is such that $S \not\vdash p$ because $S \cup \{\neg p\}$ is satisfiable by a model containing c and $\neg p$. We solve the PR Problem of revising from set S to a set S' such that $S' \vdash p$ with $C = \{a,\ c,\ \neg c\}$ by reduction to the FC Problem, following Lemma 9.

We build a defeasible theory $D = (F,R)$ where $F = \emptyset$ and $R = \{r_1 :\ \neg a, \neg b, \neg c \Rightarrow p,\ r_2 :\ a \Rightarrow c,\ r_3 :\ \neg c \Rightarrow \neg a,\ r_4 :\ b \Rightarrow c,\ r_5 :\ \neg c \Rightarrow \neg b\}$. Since clause (1) is of kind 1, it is translated in rule r_1. Clauses (2) and (3) are of kind 3, and therefore each of them is translated in two clauses.

The FC Problem asks to revise from $D = (F,R)$ such that $D \vdash -\partial p$ to $D' = (F',R)$ such that $D' \vdash +\partial p$. The set C remains unchanged, that is, $C = \{a,\ c,\ \neg c\}$.

The solution of the FC Problem is $F' = \{\neg c\}$. The deductive chain starts with $\neg c$ and applies r_3 and r_5, obtaining $D' \vdash +\partial \neg a$ and $D' \vdash +\partial \neg b$. Then r_1 can be applied, ending the chain with $D' \vdash +\partial p$.

The set of unit clauses $U = \{\neg c\}$ is used to get $S' = S \cup U = \{(1)\ a \vee b \vee c \vee p,\ (2)\ \neg a \vee c,\ (3)\ \neg b \vee c,\ (4) \neg c\}$ We have that $S' \vdash p$ and $S' \cup \{\neg p\}$ is unsatisfiable, as represented with the resolution proof tree of Figure 8.1.

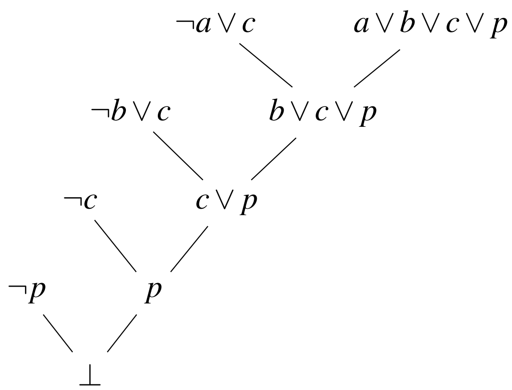

Figure 8.1: The resolution refutation of Example 13.

Definition 19 (Generalized FC Problem). Given a defeasible theory $D = (F,R)$, a set P of literals such that $D \vdash +\partial p$ ($D \vdash -\partial p$) for each $p \in P$, and a set of revisable facts $C = \{f_1,\ \dots,\ f_n\}$, with $n \geq 0$, is it possible to obtain $D' = (F',R)$ such that $D' \vdash -\partial p$ ($D' \vdash +\partial p$) for each $p \in P$, where F' is obtained by adding or removing from F facts appearing in C?

We now identify two trivial sub-cases of the FC Problem. If the set C is empty, the answer is always negative. Consider now the instance of the problem where we ask, given a defeasible theory $D = (F,R)$ and a literal p such that $D \vdash -\partial p$, whether it is possible to obtain $D' = (F',R)$ such that $D' \vdash +\partial p$. In this case, the answer to the problem is trivially positive if the literal $p \in C$ or if there is a rule $a_1, \dots, a_n \Rightarrow p$ such that $a_1, \dots, a_n \in C$.

Chapter 9

Explainable Belief Revision by Non-Monotonic Meta-Theories

The mechanical revision of logical theories has been a hot research topic since the pioneering studies of the twentieth century on computational logic. Moreover, handling change in logical theories is a concrete problem for numerous applications, such as planning, scheduling and, more generally speaking, knowledge representation. On this issue, Van Benthem [130] has been systematically approaching the emerging problems within a view that has been very fruitful in devising effective methods for belief revision.

There are two aspects of revision that need to be distinguished. One is the basic *manipulation* operations that can be performed on logical systems, as discussed in an ample stream of studies, especially when approaching Logic Programming, often under the name of *assert and retract* (see [116] for a technical view on the most updated issues). Basically, the operators studied in this research line consist in *manipulating* a theory by deleting or adding (or substituting, or modifying) elements in the theory itself. It can be proved that the so-called *impure* Prolog operators (assert and retract) act as theory change methods in the corresponding clausal form of a Prolog program. These manipulations of the theory are in some sense *non-declarative* as they operate on the logical theory directly. Actual belief revision, instead, should exhibit a *declarative* form, as discussed in the pioneering and foundational work by Alchourrón, Gärdenfors and Makinson [2]. Clearly, the two issues cannot be fully distinguished: manipulation frequently has some form of declarativity, because it is not possible to freely add items to a theory without any control, while on the other hand developing an algorithm that controls the effects of theory change may introduce de facto manipulations.

There exist a number of approaches to revise a theory written in the language of defeasible deontic logic, for instance [51] gives an ontology that models many different kinds of norm change, [77] illustrates a method for revising a theory by only changing its preferences and proves its complexity, [70] gives a framework to model abrogations and annulments, and [45] provides a framework for changing the temporal extension of temporary norms. In this work, we provide a system of meta-theories in which the theory at the super-level infers what to revise in the theory at the sub-level.

Throughout the Chapter we use the basic definitions of Defeasible Deontic Logic (DDL) introduced in Section 2.1.2 and the formalism structure for norms in DDL introduced in Section 6.1 and Definition 13.

9.1 Revision by Meta-theories

One of the objectives of this Chapter is that of providing a deductive system for revision, that is, reasoning on a theory in order to decide how to revise another theory. This allows to represent the reasoning processes of an agent that changes another logical agent, for instance a legislator that works on revising a piece of the law in order to perform a simplification of it or to obtain some specific logical consequences.

The revision processes that we are interested in are of two kinds: *manipulative revisions* or *declarative revisions*, and we describe them below.

9.1.1 Manipulative Revisions

The revisions that we call manipulative are those that manipulate directly the set of facts, the set of rules, or the set of superiorities of the theory. In this sense, they can explicitly expand or contract the theory:

Definition 20 (Manipulative expansion operator). Given defeasible theory $D = (F, R, >)$, a *manipulative expansion operator* $\mathscr{E}$ for D is an operator that gives $D' = (F', R, >)$ or $D' = (F, R', >)$ or $D' = (F, R, >')$, where

- $F' = F \cup \{f\}$, with $f \notin F$;
- $R' = R \cup \{r\}$, with $r \notin R$;
- $>' = > \cup \{p\}$, with $p \notin >$.

Definition 21 (Manipulative contraction operator). Given defeasible theory $D = (F, R, >)$, a *manipulative contraction operator* $\mathscr{C}$ for D is an operator that gives $D' = (F', R, >)$ or $D' = (F, R', >)$ or $D' = (F, R, >')$, where

- $F' = F \setminus \{f\}$, with $f \in F$;
- $R' = R \setminus \{r\}$, with $r \in R$;
- $>' = > \setminus \{p\}$, with $p \in >$.

9.1.2 Declarative Revisions

The revisions that we call declarative are those that state what the result of the revision should be, but do not explicitly specify the steps needed to obtain it. Contrarily to manipulative revisions, they do not modify directly the set of facts, the set of rules, or the set of superiorities of the theory. The idea is that these revisions give in output a specification that the components of the extension of the revised theory need to satisfy. In this sense, they can expand or contract the theory, but more than one revision process may correspond to a given extension specification.

In particular, given defeasible theory $D = (F, R, >)$, a literal l can be concluded in three disjunct manners:

1. $D \vdash +\Delta l$ (and obviously $D \vdash +\partial l$)
2. $D \vdash +\partial l$, but $D \vdash -\Delta l$
3. $D \vdash -\partial l$ (and obviously $D \vdash -\Delta l$)

There are six types of change for the declarative revision of the extension, as illustrated in Figure 9.1:

- From ($D \vdash +\Delta l$) to ($D' \vdash +\partial l$ and $D' \vdash -\Delta l$) (*weak contraction 1*)
- From ($D \vdash +\partial l$ and $D \vdash -\Delta l$) to ($D' \vdash -\partial l$) (*weak contraction 2*)
- From ($D \vdash +\Delta l$) to ($D' \vdash -\partial l$) (*strong contraction*)
- From ($D \vdash -\partial l$) to ($D' \vdash +\partial l$ and $D' \vdash -\Delta l$) (*weak expansion 1*)
- From ($D \vdash +\partial l$ and $D \vdash -\Delta l$) to ($D' \vdash +\Delta l$) (*weak expansion 2*)
- From ($D \vdash -\partial l$) to ($D' \vdash +\Delta l$) (*strong expansion*)

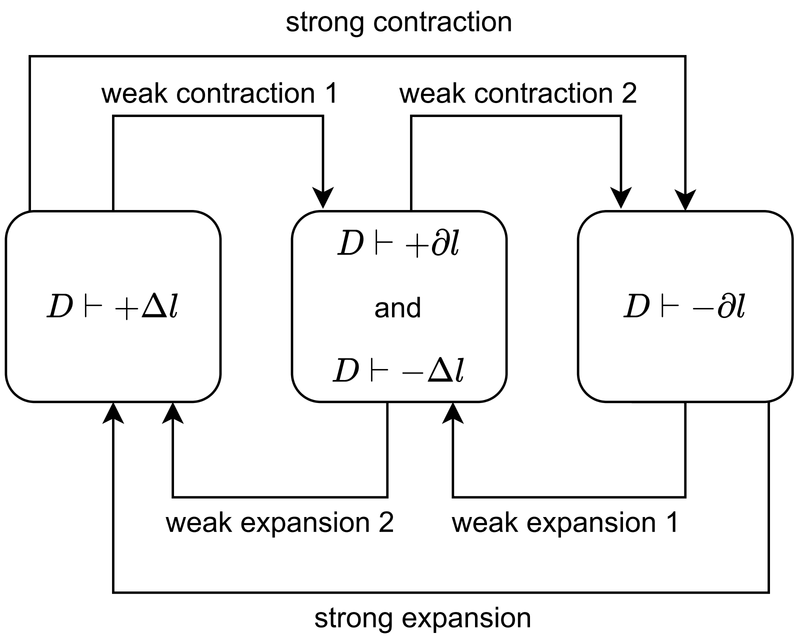

Figure 9.1: The six types of declarative revision on the extension of theory D for a given literal l.

9.1.3 Revisions and Meta-theories

We describe here the concept of meta-theory, and connect it to manipulative and declarative revisions illustrated above, and to the combination of them. We consider theories at a certain level, for instance i, and indicate this level with a subscript, for instance D_i is a theory at level i. Moreover, we consider set Lit_i as the Herbrand Base of theory D_i.

Definition 22 (Set of lifted literals). Given a defeasible theory $D_i = (F_i, R_i, >_i)$ and its Herbrand Base Lit_i, the set of lifted literals from D_i is the smallest set Y with the properties:

- if $A \subseteq Lit_i$ and $C \in Lit_i$, then $(A \hookrightarrow C) \in rules_i$, for each $\hookrightarrow \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}$;
- if $r \in rules_i$ and $s \in rules_i$, then $(r, s) \in sup_i$;
- if $l \in Lit_i$, then $(\#(l)) \in ext_i$, for each $\# \in \{\Delta, \partial\}$;
- if $x \in rules_i$ or $x \in sup_i$ or $x \in ext_i$, then $x \in Y$ and $\neg x \in Y$.

The set of lifted literals from a theory D_i at level i represents all the possible combinations of rules, superiorities and extension for that theory, and it will be used by a theory at level $i+1$ to reason on revising D_i. In order to correctly describe the defeasible reasoning process of the theory at the meta-level, it is necessary to constrain the lifted literals used at the meta-level to follow the defeasible mechanisms of the level. We achieve this by always inserting a special set of strict rules in the meta-theory:

Definition 23 (Set of logical coherency). Given a defeasible theory $D_i = (F_i, R_i, >_i)$ and its Herbrand Base Lit_i, the set of logical coherency of D_i is the smallest set S of strict rules with the properties:

- if $l \in Lit_i$, then
  $\{\Delta(l) \rightarrow \partial(l),\ \neg\partial(l) \rightarrow \neg\Delta(l),\ \partial(l) \rightarrow \neg\partial(\sim l)\} \subseteq S$;
- if $r \in >_i$ and $s \in >_i$, then $\{(r,s) \rightarrow \neg(s,r)\} \subseteq S$.

Notice that also the rules $\Delta(l) \rightarrow \neg\partial(\sim l)$ and $\Delta(l) \rightarrow \neg\Delta(\sim l)$ can be derived from S.

We can now give the definition of a meta-theory:

Definition 24 (Meta-theory). Given a defeasible theory $D_i = (F_i, R_i, >_i)$, its Herbrand Base Lit_i, the set Y of lifted literals from D_i, and the set S of logical coherency of D_i, we say that a defeasible theory $D_{i+1} = (F_{i+1}, R_{i+1}, >_{i+1})$ is a *revision meta-theory* of D_i when

- if $y \in Y$, then $y \in Lit_{i+1}$;
- if $s \in S$, then $s \in R_{i+1}$;
- if $r : A(r) \hookrightarrow l \in R_i$, then $\Rightarrow r \in R_{i+1}$;
- if $p : (r,s) \in >_i$, then $\Rightarrow p \in R_{i+1}$;
- if $l \in Lit_i$ and $D_i \vdash +\Delta l$, then $\Rightarrow \Delta(l) \in R_{i+1}$;
- if $l \in Lit_i$ and $D_i \vdash -\Delta l$, then $\Rightarrow \neg\Delta(l) \in R_{i+1}$;
- if $l \in Lit_i$ and $D_i \vdash +\partial l$, then $\Rightarrow \partial(l) \in R_{i+1}$;
- if $l \in Lit_i$ and $D_i \vdash -\partial l$, then $\Rightarrow \neg\partial(l) \in R_{i+1}$;
- if $(t,u) \in >_{i+1}$, then
  - t and u have opposite conclusions, or
  - there exists $l \in Lit_i$ such that one between t and u concludes $\Delta(l)$ or $\partial(l)$ and the other concludes $\neg\partial(l)$ or $\partial(\sim l)$ or $\Delta(\sim l)$, or
  - there exist $r \in >_i$ and $s \in >_i$ such that one between t and u concludes (r,s) and the other concludes (s,r).

Definition 24 shows how to *lift* a theory from level i to level $i+1$. The lifted literals that correspond to the rules, superiorities and extension of D_i are included as hypotheses in D_{i+1}. They represent the starting state of theory D_i before revision and they can be changed by reasoning at the meta-level. For instance, one such hypothesis can be defeated by a stronger rule or by a fact at level $i+1$.

Besides these hypotheses, also the strict rules of S have to be in the meta-theory, as these are necessary to represent the logical coherency of the theory at level i. For instance, these are used to constrain that if $D_i \vdash +\Delta l$, then $D_i \vdash +\partial l$, and if $D_i \vdash -\partial l$, then $D_i \vdash -\Delta l$. Moreover, we also constrain the form of the pairs in the superiority relation $>_{i+1}$. We allow pairs where the two rules have opposite conclusions, as it is standard in defeasible logic; and also pairs where the two rules have conclusions that are not opposite, but one is not coherent with the other: starting from these conclusions it is possible to derive opposite assertions by applying the rules in the set S of logical coherency.

After lifting a theory, a reasoning process takes place at the meta-level, so that the extension of D_{i+1} can be used to revise D_i. For what regards manipulative revisions, that represent a syntactic point of view, we need literals that correspond to to expansion and contraction operators for rules and superiorities in D_i. The literals that we use for this aim are the lifted literals from D_i that correspond to rules and superiorities at level i. If one such literal is positive, it represents an expansion operator; if it is negative, it represents a contraction operator.

On the other hand, expansion and contraction operators for facts in D_i are handled by declarative revisions. More in general, for declarative revisions, that represent a more semantic point of view, the literals $\Delta_i(l)$, $\neg\Delta_i(l)$, $\partial_i(l)$, $\neg\partial_i(l) \in Lit_{i+1}$ are to be used at level $i+1$ to revise the conclusion of literal l at level i.

The reasoning process that we just described corresponds to a *drop* from the meta-theory to the theory and is used to perform the revision of the theory:

Definition 25 (Revision by Meta-theory). Given a defeasible theory $D_i = (F_i, R_i, >_i)$ and revision meta-theory $D_{i+1} = (F_{i+1}, R_{i+1}, >_{i+1})$, the revision D'_i of D_i is computed on the extension of D_{i+1} as follows

- if $D_{i+1} \vdash +\partial r$, then $r \in R'_i$;
- if $D_{i+1} \vdash +\partial\neg r$, then $r \notin R'_i$;
- if $D_{i+1} \vdash +\partial p$, then $p \in >'_i$;
- if $D_{i+1} \vdash +\partial\neg p$, then $p \notin >'_i$;
- if $D_{i+1} \vdash +\partial\Delta(l)$, then $D'_i \vdash +\Delta l$;
- if $D_{i+1} \vdash +\partial\neg\Delta(l)$, then $D'_i \vdash -\Delta l$;
- if $D_{i+1} \vdash +\partial\partial(l)$, then $D'_i \vdash +\partial l$;
- if $D_{i+1} \vdash +\partial\neg\partial(l)$, then $D'_i \vdash -\partial l$.

The first four items in Definition 25 correspond to manipulative revisions, while the last four correspond to declarative revisions. In this framework we do not distinguish what's inside the strict portion of the extension: a literal that is definitely provable is equivalent to a fact. We only use the partial extension $+\partial$ of D_{i+1} as it is sufficient to represent the changes that we are interested in.

The example that follows shows some of the basic mechanisms of the framework:

Example 14. Consider a defeasible theory $D_0 = (F_0, R_0, >_0)$, with $F_0 = \{a,\ c\}$, $R_0 = \{\alpha : a \Rightarrow b,\ \beta : c \Rightarrow \neg b,\ \gamma : d \Rightarrow e\}$, and $>_0 = \{(\alpha,\ \beta)\}$. The Herbrand Base Lit_0 is $\{a,\ b,\ c,\ d,\ e,\ \neg a,\ \neg b,\ \neg c,\ \neg d,\ \neg e\}$. We have that $D_0 \vdash +\Delta a$, $D_0 \vdash -\Delta b$, $D_0 \vdash +\Delta c$, $D_0 \vdash -\Delta d$, $D_0 \vdash -\Delta e$, $D_0 \vdash +\partial a$, $D_0 \vdash +\partial b$, $D_0 \vdash +\partial c$, $D_0 \vdash -\partial d$, and $D_0 \vdash -\partial e$.

In order to build a meta-theory $D_1 = (F_1, R_1, >_1)$, the lifted literals corresponding to rules, superiorities and extension at level 0 have to be hypotheses at level 1. Moreover, R_1 must contain the set S of logical coherency. In this example, we have $S = \bigcup_{l \in Lit_0}\{\Delta(l) \to \partial(l),\ \neg\partial(l) \to \neg\Delta(l),\ \partial(l) \to \neg\partial(\sim l)\} \cup \bigcup_{r,s \in >_0}\{(r,s) \to \neg(s,r)\}$. Consider the meta-theory $D_1 = (F_1, R_1, >_1)$, with $F_1 = \emptyset$, $R_1 = \{\Rightarrow \alpha,\ \Rightarrow \beta,\ \Rightarrow \gamma,\ \Rightarrow (\alpha,\beta)\} \cup \{\Rightarrow \Delta(a),\ \Rightarrow \neg\Delta(b),\ \Rightarrow \Delta(c),\ \Rightarrow \neg\Delta(d),\ \Rightarrow \neg\Delta(e),\ \Rightarrow \partial(a),\ \Rightarrow \partial(b),\ \Rightarrow \partial(c),\ \Rightarrow \neg\partial(d),\ \Rightarrow \neg\partial(e)\} \cup S \cup \{(\alpha,\beta) \to \neg\beta,\ \neg\partial(d) \Rightarrow \neg\gamma\}$, and $>_1 = \emptyset$.

The only rules that are proper to R_1 are $(\alpha,\beta) \to \neg\beta$ and $\neg\partial(d) \Rightarrow \neg\gamma$, as they give new information that is not present in D_0: the first one is a manipulative contraction rule that says that β is contracted when it is defeated by α, while the second one is a manipulative contraction rule that says that γ is contracted when it is not activated. We compute the extension of D_1 in order to drop it and revise D_0. In particular, we have that $D_1 \vdash +\partial\neg\beta$ and $D_1 \vdash +\partial\neg\gamma$: by reasoning in theory D_1 we deduce that we should contract rules β and γ from D_0.

Consider now a theory D and a revision by meta-theory that concludes that b should not be a fact and that all rules concluding in b should be removed from D, but also that b should still be derived. This situation is unsatisfiable: there are cases in which the resulting revision by meta-theory is unfeasible. For this reason, we introduce a special notion of satisfiability:

Definition 26 (D_i-satisfiability). Given a defeasible theory $D_i = (F_i, R_i, >_i)$ and the extension $\{+\Delta_{i+1},\ -\Delta_{i+1},\ +\partial_{i+1},\ -\partial_{i+1}\}$ of revision meta-theory $D_{i+1} = (F_{i+1}, R_{i+1}, >_{i+1})$, we say that D_{i+1} is D_i-satisfiable if it is possible to revise D_i in D_i' that satisfies the constraints in $+\partial_{i+1}$.

9.1.4 Properties of the System

The framework described above can express a number of revisions that exist in the scientific literature and here we show that it is complete in this sense. The operators will be represented with a combination of manipulative and declarative revisions.

Lemma 11. *The standard revision operators for the extension of a defeasible theory can be represented in the meta-theoretical framework.*

Proof. We show the standard (canonical) revision operators for the extension and then we map them to the meta-theoretical framework of this chapter. Given a defeasible theory $D = (F, R, >)$, a literal l, and a set of revisable facts $C = \{f_1,\ \ldots,\ f_n\}$, with $n \geq 0$, the canonical operators are

- (Contraction) if $D \vdash +\partial l$, obtain $D' = (F', R', >')$ such that $D' \vdash -\partial l$;
- (Revision) if $D \vdash +\partial l$, obtain $D' = (F', R', >')$ such that $D' \vdash +\partial \sim l$;
- (Expansion) if $D \vdash -\partial l$, obtain $D' = (F', R', >')$ such that $D' \vdash +\partial l$;

where F' is obtained by adding or removing from F facts appearing in C.

The operators to revise a theory $D_i = (F_i, R_i, >_i)$ are mapped in our framework as follows:

- (Contraction) if $\Rightarrow \partial(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\neg\partial(l)$;
- (Revision) if $\Rightarrow \partial(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\partial(\sim l)$;
- (Expansion) if $\Rightarrow \neg\partial(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\partial(l)$;

where parts of the extension (comprising facts), rules, and precedences of D_i that are not revisable are to be added as facts in D_{i+1}. □

Lemma 12. *Belief revision for classical logic can be represented in the meta-theoretical framework.*

Proof. We assume that belief revision for classical logic follows the standard AGM model [2]. Given a classical theory $D = (F, R)$, a literal l, and a set of revisable facts $C = \{f_1,\ \ldots,\ f_n\}$, with $n \geq 0$, the AGM model gives three operators for revision:

– (Contraction) if $D \vdash +\Delta l$, obtain $D' = (F', R', >')$ such that $D' \vdash -\Delta l$;
– (Revision) if $D \vdash +\Delta l$, obtain $D' = (F', R', >')$ such that $D' \vdash +\Delta \sim l$;
– (Expansion) if $D \vdash -\Delta l$, obtain $D' = (F', R', >')$ such that $D' \vdash +\Delta l$;

where F' is obtained by adding or removing from F facts appearing in C.

The operators to revise a theory $D_i = (F_i, R_i, >_i)$ are mapped in our framework as follows:

– (Contraction) if $\Rightarrow \Delta(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\neg\Delta(l)$;
– (Revision) if $\Rightarrow \Delta(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\Delta(\sim l)$;
– (Expansion) if $\Rightarrow \neg\Delta(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\Delta(l)$;

where parts of the extension (comprising facts), rules, and precedences of D_i that are not revisable are to be added as facts in D_{i+1}.

Since the AGM model works with classical logic, we are using the $\pm\Delta$ proof tags instead of $\pm\partial$ at level i. □

Lemma 13. *Revision of defeasible preferences can be represented in the meta-theoretical framework.*

Proof. The Defeasible Superiority Changing Problem (DSC Problem) is a process of revision in a defeasible theory where no changes to rules and facts are allowed: the revision acts only on the superiority relation. Given a defeasible theory $D = (F, R, >)$ and a literal l, the DSC Problem is formalized in [77] as

– (Contraction) if $D \vdash +\partial l$, obtain $D' = (F, R, >')$ such that $D' \vdash -\partial l$;
– (Expansion) if $D \vdash -\partial l$, obtain $D' = (F, R, >')$ such that $D' \vdash +\partial l$;

where in D' only the superiority relation $>'$ has been modified.

The operators to revise a theory $D_i = (F_i, R_i, >_i)$ are mapped in our framework as follows:

– (Contraction) if $\Rightarrow \partial(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\neg\partial(l)$;
– (Expansion) if $\Rightarrow \neg\partial(l) \in R_{i+1}$, derive $D_{i+1} \vdash +\partial\partial(l)$;

where the strict extension (representing facts) and rules of D_i are to be added as facts in D_{i+1} because they are not revisable. Moreover, each element of the subset of lifted literals that corresponds to a rule that does not appear in R_i is to be added negated as a fact in D_{i+1}, so that the corresponding rule will not be added during the revision. □

Theorem 6. *The meta-theoretical framework is complete with respect to defeasible revision, classical belief revision, and revision of defeasible preferences.*

Proof. The proof follows straightforwardly from Lemmas 11, 12 and 13. □

Chapter 10

Discussion

In this Discussion Chapter we we apply some of the contents of the thesis to solve the problems presented for the legal text given in Chapter 2. Let us consider again the regulation thereby described:

Access to the bus

1. It is mandatory to board from the front door
2. It is forbidden to board when the vehicle is moving
3. It is mandatory to have a travel document or to purchase a ticket on board of the bus
4. A travel document can be a ticket or a subscription
5. Tickets purchased on board have an extra charge of 1.00 €
6. It is mandatory to validate the ticket
7. A ticket which is not validated is equivalent to travelling without a travel document
   Travel behaviour
8. It is forbidden to smoke
9. It is forbidden to hold disrespectful or indecorous behaviour
10. It is forbidden to hold behaviour that can harm the safety of the service
    Penalties
11. Travel documents must be exhibited at the request of drivers or inspectors
12. Passengers without a travel document shall pay a fine of 100.00 €
13. Passengers without a travel document that pay while on the bus shall pay a fine of 80.00 €
14. Passengers that smoke shall pay a fine of 150.00 €
15. Passengers that hold disrespectful or indecorous behaviour shall leave the bus
16. Passengers that hold behaviour that can harm the safety of the service shall pay a fine of 250.00 €
    Animal transport
17. Pets travel for a fee
18. Pets shall wear a muzzle and be kept on a leash
19. Pets accompanying visually impaired persons are an exception, they travel free of charge and can be without a muzzle
    Luggage transport
20. Standard luggage does not exceed weight of 10 kg
21. Heavy luggage exceeds weight of 10 kg and does not exceed weight of 20 kg

22. Very heavy luggage exceeds weight of 20 kg
23. It is permitted to carry standard luggage free of charge
24. It is permitted to carry heavy luggage with payment of fare #1
25. It is permitted to carry very heavy luggage with payment of fare #2
26. One extra bag can be transported with payment of a special fare
27. For passengers with an extra bag, it is mandatory to put one bag in the luggage compartment: the bag in the luggage compartment has to be heavier than the other bag

10.1 Translation from Natural to Formal Language

In order to employ Defeasible Deontic Logic or Adjective Logic, it is first necessary to translate the norm from natural language to one of the two formal languages.

For the translation to Defeasible Deontic Logic (see Chapters 3 and 4), we need to analyse the norm and detect in the text the linguistic structures that correspond to the norm body, definitions, obligations, prohibitions, permissions, and hierarchy of rules:

– Definitions usually appear at the beginning of the norm and are typically introduced by using the verbs *to mean* or *to be* or keywords such as *equivalent* and *interpretation*. Definitions are translated to strict rules with $\rightarrow$.
– Rules in the norm body can be detected by structures such as *if... then*, *when... then*. These are translated to defeasible rules with $\Rightarrow$.
– Obligations can be introduced by keywords such as *mandatory, obligatory, shall, must*. Obligations are translated to the deontic operator $\mathscr{O}$.
– Prohibitions can be introduced by keywords such as *forbidden, prohibited, shall not, can not, may not*. Prohibitions are translated to the deontic operator $\mathscr{F}$.
– Permissions can be introduced by keywords such as *allowed, permitted, consented, tolerated, accepted, can, may*. Permissions are translated to the negation $\neg\mathscr{F}$ of the deontic operator $\mathscr{F}$.
– A hierarchy of rules can be introduced by keywords such as *except*, *exception*, *unless* and it is translated into a superiority relation $>$.

For the translation to Adjective Logic, we need to analyse the norm and detect in the text the linguistic structures that correspond to adjectives, comparatives, degree modifiers, dualization, nouns, and agents:

– Adjectives can be detected by employing standard NLP tools that identify different POS (Parts of Speech) and are translated to adjective predicates.
– Comparatives can be detected by structures such as *more... than*, *less... than*, *-er... than* and are translated to comparative adjective predicates. Equal comparatives can be detected by structures like *as... as* or by adverbs such as *equally* and are translated to equal comparative adjective predicates.
– Degree modifiers can be detected by adverbs such as *very* and are translated to degree adjective predicates by using the adjective that comes after *very*.
– Dualizations can be detected by keywords such as *opposite* and are translated to dual adjective predicates.

– Nouns can be detected by employing standard NLP tools that identify different POS and are translated to constants or to agents, that are the nouns used with verbs such as *to believe* or *to think*.

Some parts of the translation process from natural to formal language can still be challenging because natural language can be convoluted or ambiguous.

10.2 The Regulation in Defeasible Deontic Logic

In this section we give a possible formalization for the text. The regulation can be translated to defeasible theory $D = (F, R, >)$ such that the set of facts is $F = \emptyset$ and R is a set composed as follows, where $\rightarrow$ is used for strict rules, $\Rightarrow$ is used for defeasible rules, and $\mathcal{O}$ and $\mathcal{F}$ are deontic operators representing something that is *obligatory* or *forbidden*, respectively:

1. $\mathcal{O}\ frontdoor\_board$
2. $\mathcal{F}\ moving\_board$
3. $a:\ \mathcal{O}\ travel\_doc$,
   $b:\ \neg ticket \Rightarrow \mathcal{O}\ onboard\_purchase$,
   $c:\ \neg subscription \Rightarrow \mathcal{O}\ onboard\_purchase$
4. $a:\ ticket, validate \rightarrow travel\_doc$
   $b:\ subscription \rightarrow travel\_doc$
5. $a:\ onboard\_purchase \rightarrow extra\_1$
   $b:\ onboard\_purchase \rightarrow travel\_doc$
6. $ticket \Rightarrow \mathcal{O}\ validate$
7. $ticket,\ \neg validate \rightarrow \neg travel\_doc$

8. $\mathcal{F}\ smoke$
9. $a:\ \mathcal{F}\ disrespectful$
   $b:\ \mathcal{F}\ indecorous$
10. $\mathcal{F}\ unsafe$

11. $a:\ request \Rightarrow \mathcal{O}\ show\_doc$
    $b:\ travel\_doc \rightarrow show\_doc$
    $c:\ \neg travel\_doc \rightarrow \neg show\_doc$
12. $request,\ \neg show\_doc \Rightarrow \mathcal{O}\ fine\_100$
13. $a:\ request,\ \neg show\_doc,\ instant\_pay \Rightarrow \mathcal{O}\ fine\_80$,
    $b:\ request,\ \neg show\_doc,\ instant\_pay \Rightarrow \neg\mathcal{O}\ fine\_100$
14. $smoke \Rightarrow \mathcal{O}\ fine\_150$
15. $a:\ disrespectful \Rightarrow \mathcal{O}\ leave$
    $b:\ indecorous \Rightarrow \mathcal{O}\ leave$
16. $unsafe \Rightarrow \mathcal{O}\ fine\_250$

17. $pet \Rightarrow \mathcal{O}\ fee$
18. $a:\ pet \Rightarrow \mathcal{O}\ muzzle$
    $b:\ pet \Rightarrow \mathcal{O}\ leash$

19. $a:\ pet,\ visually\_imp \Rightarrow \neg\mathscr{O}\ fee$
 $b:\ pet,\ visually\_imp \Rightarrow \neg\mathscr{O}\ muzzle$

20. $less\_10 \rightarrow standard$
21. $more\_10,\ less\_20 \rightarrow heavy$
22. $more\_20 \rightarrow very\_heavy$
23.
24. $heavy \Rightarrow \mathscr{O}\ fare\_1$
25. $very\_heavy \Rightarrow \mathscr{O}\ fare\_2$
26. $extra\_bag \Rightarrow \mathscr{O}\ special\_fare$
27. $extra\_bag \Rightarrow \mathscr{O}\ heavier\_in\_compartment$

The superiority relation $>$, defined on rules with opposite consequences, is as follows:

- $r_{13b} > r_{12}$
- $r_{19a} > r_{17}$
- $r_{19b} > r_{18a}$

A rule in R can take one of the following forms, where $l_1,...,l_n,l$ are propositional literals and $\mathscr{M} \in \{\mathscr{O},\mathscr{F}\}$:

- A definition $l_1,...,l_n \rightarrow l$,
- An unconditional rule $\mathscr{M}\ l$, or
- A conditional rule $l_1,...,l_n \Rightarrow\ \mathscr{M}\ l$

10.3 Reasoning in Defeasible Deontic Logic

In this section we give an example of reasoning with the formalized text. We describe a situation and we check for compliance to the regulation. If the situation is not compliant, we want to compute which are the penalties of the case.

Suppose now that we are in the situation where a passenger boards form the front door, has a ticket but does not validate it, and smokes on the bus. The inspector gets on the bus and the passenger is ready to immediately pay the fine. The situation can be represented by the set of facts

$$F' = \{frontdoor\_board,\ ticket,\ \neg validate,\ smoke,\ request,\ instant\_pay\}.$$

We want to derive the logical extension of $D' = (F',R,>)$ in order to get the legal consequences for the passenger. Recall that $+\Delta\ q$ is used when q is definitely provable, while $+\partial\, q$ is used when q is defeasibly provable.

Since r_1,r_2,r_8,r_9 and r_{10} are unconditional rules, the corresponding literal can be definitely proved: $+\Delta\ \mathscr{O}\ frontdoor\_board$, $+\Delta\ \mathscr{F}\ moving\_board$, $+\Delta\ \mathscr{F}\ smoke$, $+\Delta\ \mathscr{F}\ disrespectful$, $+\Delta\ \mathscr{F}\ indecorous$, and $+\Delta\ \mathscr{F}\ unsafe$.

Since the passenger has a ticket, he has an obligation to validate it by r_7, that allows to derive $+\partial\ \mathscr{O}\ validate$.
Since the passenger has a ticket but does not validate it, we can apply r_7 to get $+\Delta\ \neg travel\_doc$: a ticket that is not validated is not a proper travel document.
When the inspector gets on the bus she requests the passenger to exhibit a travel document,

so r_{11a} yields $+\partial\ \mathscr{O}\ show\_doc$, but since the passenger does not have a proper travel document, he cannot show it by rule r_{11c}, giving $+\Delta\ \neg show\_doc$.
Since the inspector asked for a document and the passenger could not provide it, he should be fined 100.00 € by r_{12}. However, the passenger is able to pay immediately and by the precedence relation $r_{13b} > r_{12}$ we have that $+\partial\ \mathscr{O}\ fine\_80$ and $+\partial\ \neg\mathscr{O}\ fine\_100$. That is, the passenger has to pay a fine of 80.00 € and not one of 100.00 €. Moreover, the passenger has smoked on the bus, and by r_{14}, that yields $+\partial\ \mathscr{O}\ fine\_150$, he has to pay another fine of 150.00 €. In summary, we say that the passenger is uncompliant because he does not have a proper travel document and because he has smoked on the bus.

10.4 Reasoning with Adjectives

In general, adjectival structures are hard to treat, especially when they are vague. For instance, this happens when the legislator has to qualify some behaviours, and in our example regulation we see this with the adjectives *disrespectful*, *indecorous* and *unsafe*.
The adjective *unsafe* can have different interpretations, as a transport company may consider boarding a dog as *safe* behaviour while another may consider it *unsafe*. Moreover, the meaning of these adjectives depends on common sense and changes with time as society evolves, and a model to represent this implicit norm changes becomes necessary.

A famous example is that of Rosa Parks: she was found guilty of *disorderly conduct* in 1955 for failing to give up her seat to a white passenger on a Montgomery, Alabama city bus, and her behavior fitted the definition at the time. However, perception of society was changing and the Supreme Court overturned the decision in 1956 by declaring that segregation on public transportation was unconstitutional, and in today's society the behaviour of Rosa Parks would not be considered *disorderly conduct*.

10.4.1 Extraction to Adjective Logic

We now focus on the *Luggage transport* section of the regulation and specifically on the last rule.
We translate to the Adjective Logic $\mathscr{A}$-Log of Chapter 5 the concept that the bag in the luggage compartment is heavier than the other bag. Suppose we have a language composed of objective adjective symbol *heavy*, constant symbols *compartment_bag* and *other_bag*. Then the concept can be represented by the set

$$S = \{heavy^{+}(compartment\_bag, other\_bag),\ O(heavy)\}.$$

Suppose then that the inspector agent, that we represent with the agent symbol *inspector*, boards the bus and checks the weights of the two bags. She finds out that the compartment bag is heavy, but not very heavy, while the other bag is very heavy. This can be represented by the set

$$T = \{inspector : heavy(compartment\_bag),\ inspector : \neg heavy^{*}(compartment\_bag),$$

$$inspector : heavy^{*}(other\_bag)\}.$$

We now check whether a contradiction is derivable from $S \cup T$. Since *heavy* is and objective adjective, the Agent Label Elimination inference rule can be applied, yielding

$$\frac{inspector : \neg heavy^*(compartment\_bag),\; O(heavy)}{\neg heavy^*(compartment\_bag)}\; AE$$

and

$$\frac{inspector : heavy^*(other\_bag),\; O(heavy)}{heavy^*(other\_bag)}\; AE$$

From these conclusions, the Comparative Introduction inference rule can be applied, yielding

$$\frac{heavy^*(other\_bag),\; \neg heavy^*(compartment\_bag)}{heavy^+(other\_bag, compartment\_bag)}\; CI$$

We almost derived the contradiction, since we have both $heavy^+(compartment\_bag, other\_bag)$ and $heavy^+(other\_bag, compartment\_bag)$. We apply the Comparative Permutation inference rule, yielding

$$\frac{heavy^+(other\_bag, compartment\_bag)}{\neg heavy^+(compartment\_bag, other\_bag)}\; CP$$

At this point we can derive the contradiction with

$$\frac{heavy^+(compartment\_bag, other\_bag),\; \neg heavy^+(compartment\_bag, other\_bag)}{\bot}$$

showing us that the situation detected by the inspector agent is not compliant with the regulation and specifically with $r_{27}:\ extra\_bag \Rightarrow \mathscr{O}\ heavier\_in\_compartment$.

10.5 Interaction between Defeasible Deontic Logic and Adjective Logic

The interaction between the two logical languages has to be studied more in order to be managed seamlessly. The idea is to separately perform the two translations, reason internally in either one of the logical systems and only successively share results.

For instance, an adjectival structure in $\mathscr{A}$-Log may correspond to a literal in defeasible deontic logic or vice-versa. This information sharing process constitutes a method with greater expressive power than both the logics and allows to derive conclusions and check conditions that otherwise would not be possible. Since the two languages differ, the problem is especially challenging.

10.6 Management of Explicit and Implicit Change

As seen in Chapter 6, norm change can be *explicit*, by action of the legislator, or *implicit*, when for instance another piece of the normative system or the meaning of a term employed in the norm changes. After a norm change it is is crucial to re-compute the conclusions of the logical system.

Defeasible rules can be augmented with a label with information such as temporal extension, territorial extension or application group of the norm, and such a label can be modified in order to account for explicit norm change. Other explicit changes may involve a structural modification of the rules, by removing or adding literals to the premises.

Implicit norm change caused by a change in the norms hierarchy can be represented by modifying the superiority relation $>$. On the other hand, implicit norm change caused by a change in semantics of the legal terminology is harder to represent, as we described in Section 10.4 with adjectives *disrespectful*, *indecorous* and *unsafe*. If the change in meaning is explicitly given by the legislator it is quite easy to modify the corresponding definition in defeasible deontic logic, but when the change is implicit it seems to be simpler to compute the consequences of the change in $\mathscr{A}$-Log and then propagate them back to the other system. Indeed, $\mathscr{A}$-Log is specifically designed to deal with adjectives and it may prove to be a solution to treat the linguistic problems that defeasible deontic logic is not equipped to address.

We illustrate the situation by continuing on the running example. Suppose the perception of the adjective *heavy* changes for the inspector agent because she checks the weights of the two bags with a faulty scale. She then believes that the compartment bag is heavy, while the other bag is not heavy. This can be represented by the set

$$T = \{inspector : heavy(compartment\_bag),\ inspector : \neg heavy(other\_bag)\}.$$

The set allows to derive $heavy^{+}(compartment\_bag, other\_bag)$, and therefore the situation results to be compliant with $r_{27} :\ extra\_bag \Rightarrow \mathscr{O}\ heavier\_in\_compartment$: the change in meaning of the adjective is successfully propagated from $\mathscr{A}$-Log to defeasible deontic logic.

Chapter 11

Conclusions

The contributions of this thesis can be summarized by the definition of the items in the following points:

- a pipeline for the translation of a legal text from natural to formal language and its application to the case of natural resources contracts,
- a logical system aimed at reasoning with gradable adjectives,
- an ontology to represent change in a normative system,
- some basic mechanisms by which an agent may acquire new norms,
- a study on the problem of revising a defeasible theory by only changing its facts and the application of this to the problem of measuring pledges between organizations,
- a general theory that includes as specific cases all the three previous points.

In this Chapter we take conclusions on each of these points and take a glance on possible future work related to them.

11.1 The Legal Text Analysis Pipeline

We have investigated the problem of defining a methodology for the construction of specific pipelines in order to process legal texts. This effort is part of a long-term investigation regarding different aspects of the production of analytical pipelines for specific purposes. In part, this analysis has invested the domain of social networks [30, 43] and the domain of environmental document analysis [46, 44].

The specificity of a legal text, as discussed in this thesis, is that it is at the crossroads of specific language, specific technicalities and specific purposes, making the usage of general methods not enough.

In this thesis we discussed the development of a pipeline for translating legal texts into a formal language, defeasible deontic logic. The methodology we developed is then applied to a sample case, an exploration permit for petroleum in Western Australia. The permit is part of the publicly available *resourcecontracts.org* corpus, a repository of more than 2500 legal documents related to natural resources. These documents are written in several languages and involve numerous countries, both as sources of natural resources, and as contractors' sites. While the pipeline given in this thesis is specific to resource contracts, we have also worked on another pipeline for documents involving funding opportunities.

So far, the results we obtained are very promising, as the number of basic errors is less than one might expect and the quality of the translation is rather good, although only partial at this stage. The three main issues of the methodology that we have at the moment are:

1. It is necessary to use a *context* in order to know the kind of legal document that we are treating. For instance, words such as *fruitful* or *extension* may have different meanings depending on the context.
2. It is necessary to have the availability of the *corpus* of legal documents we are referring to, that gives the context.
3. Also the *normative background* has to be available, and in certain contexts more than one normative background may apply.

The first steps of the development shall therefore relate to the extension of the method in order to capture more details of the document that are at the moment neglected. We shall then experimentally determine the behavior of methods to build the gazetteer lists in an automated (or possibly semi-automated) way, and repeat performance measures on the domain. We are also carrying out a gold standard test, where a group of legal experts shall check the validity of the translation with a specific methodology by comparing the application of the automated method with their legal reasoning to measure the difference.

11.2 Reasoning with Gradable Adjectives

In this thesis we have studied the problem of defining a framework for reasoning with assertions obtained by predicating an adjective onto a noun, called *nominal sentences*. This research has proven that it is possible to derive conclusions from a theory made up of nominal sentences in polynomial time when each sentence is formed by a sole literal. However, the algorithms presented here can be used in a general logical propositional framework, but the set of formulae needs to be previously translated into a corresponding set of atomic sentences, that is not of polynomial size. Therefore the problem of deriving consistency of adjectival formulae, each containing more than one single literal, is NP-hard.

There have been several aspects that we have not been able to deal with yet: the interaction between adjectives and nouns, the interaction among adjectives, the usage of different modifiers (different in terms of degree), and the extension of the investigation to adjectives that are not intersective. Moreover, we still need to investigate, as anticipated in Section 5.2, the behaviour of crisp adjectives (ungradable ones) and unpolarised ones in their unified semantics as related to the extension of measures associated to the adjective itself.

Another aspect that needs to be further investigated, as briefly introduced in Section 5.2, is the possibility of introducing various *intensifiers* and *mitigators*, and to further devise operators for limiting the usage of these intensifiers and mitigators to specific categories of adjectives. Apart from intensifiers and mitigators, there exist other modifiers of the scope of adjectives that can cause effects on the meaning of these and are worth studying, as in expressions *painfully heavy* or *brightly red*.

Another topic that we did not treat in this thesis is the temporal shift of meaning in nouns produced by certain adjectives such as *former* and *future*, that are *non-subsective*.

However, there are also intersective effects that are generated by these adjectives when attached to nouns: *Ian is my former employer* generates the intersective expression *Ian used to be my employer*, that has no adjective predicated. This *apparent intersectivity* due to the temporal non-subsective nature of adjectives is worth studying as well.

We are currently exploring three extensions of the theory, and also two multi-disciplinary approaches to the corresponding problems in cognitive theory along with a team of language theorists. The first extension allows to incorporate adjectives in a taxonomy of noun classes. This will make it possible to provide distinctions of judgments derived from adjectives referred to nouns as in *small elephant* or *big mouse*, but also in solving the issue of the sentence in Example 1, H "I can't say whether Jill's doily is nicer than Joe's song" for the two elements belong to distinct categories that are in different positions in the taxonomy of nouns.

The second extension is related to analogies between adjectives, that can be *strong analogies*, as for *small* and *tiny* that are on the same *measure*, but also crossed or *weak analogies*, where the meaning of one adjective overlaps the meaning of the other one, as for *large* and *wide* or for synonyms like *sad* and *miserable*. We shall also focus on the relationships between adjectives and qualities as partly investigated in [7].

The third extension consists in looking for degree modifiers other than *very*, for instance *quasi, rather, slightly, somewhat* and locutions such as *kind of* or the postponed *to some extent*. There are two cognitive aspects that we shall explore with specific efforts in multidisciplinary contexts: the *adjective learning* process, in which we assume that *reenforcement learning* cognitive processes could match the development of adjective semantics, and the *translation* process, that presents problems related to the mismatch of sentence interpretations.

A particularly puzzling theme that regards adjectives is the difference (in terms of logical consequences) in nominal sentences involving adjectives that are related to *sensorial apparatuses*. For instance, as exemplified in the Chapter, *yellow* is unpolarised, but on the contrary, shall we consider taste-related adjectives such as *sweet* and *bitter* antonyms? And what about *sharp* and *smooth*? Very often, judgments expressed by senses are easily counter-positive for some cases (sweet is not bitter), and comparable even if distinguished (sweet and sour tastes may be together). Analogously, sharp is not the antonym of smooth, but it is *rare* to see an object that is simultaneously smooth and sharp.

At the end of the definition of these processes of reasoning, there shall be one, basic, complex problem to deal with, that can be also regarded as a general issue: as we *observe* an adjective behaving in a sufficiently large, reasonably balanced and significant set of cases, is there any method, statistically well-founded, to *infer* its class, and consequently derive the meta-properties that allow to concretely reason on the assertions made up on that adjective? Consider, for instance, an objective adjective: what makes it possible to infer its class? Do the observations prevent us to use it metaphorically, creating complex language issues (as clearly exemplified by intentional misuses *very pregnant* or *more pregnant*, that have some *implicit* metaphoric meaning related to the temporal extension of the adjective)?

11.3 An Ontology of Changes in Normative Systems

Regarding the ontology of changes, we discussed a general framework within which we classify the possible types of changes that a norm can be subject to. In our work we move the focus from the revision mechanisms to the different kinds of changes, presenting an ontology of normative change. The classification is defined in an abstract way and many efforts are still required to solve the problems arising from the application of the peculiarities of the classes to single cases in order to make the model work in an appropriate way in all the specific cases. More in detail we are investigating extensions to the issue and reform models, exploring the non-trivial relationship between application group and application topic of a norm, and studying the ways in which territorial changes can interfere with temporal ones.

We acknowledge here three directions to expand the current work. The first aspect relates norm change to time intervals. When a piece of law changes, its legal effects have to change accordingly. For instance, the law may change from punishing a behavior to only punishing the achievement obtained with that behaviour. In order to model this, we propose to extend the current framework with event calculus features. This allows to specify pre-conditions and post-conditions to normative rules, and in doing so to express causation. Previous work in theory change has the limitation that in order to handle time it is necessary to work on two different levels, a concrete one and an abstract one. There exist three typologies of operations: actions, normative operations, and revision of norms. Normative operations work on the abstract level and therefore constrain action and revision of norms, that work on the concrete level. The second work direction is aimed at handling this discrepancy between abstract and concrete level. Another aspect is the epistemological one. In epistemic logics, modal operators for knowledge and belief are available to express notions other than truth. For instance in our context we may want to express that an agent believes something even if this is not true. Moreover, it is also possible to model the relationship between the agents with specific communication logics that interact with epistemic states.

In conclusion, the framework for managing changes in normative systems can be expanded in three ways:

- Representing causation by event calculus features;
- Balancing abstract and concrete level for normative operations and actions;
- Expressing the epistemological states of the agents.

11.4 Acquisition and Learning of Principles and Norms

There are different ways in which this study shall be brought to maturity. At first we shall investigate methods to integrate trustability, and consequently negotiation processes [30, 29, 31]. In particular, the purpose of this research line shall be to confirm variants of the methods employed to perform integration of new rules when derived by observations. Consider the case in which I trust someone and observe her performing some activity. I see that her behaviour has some coherence, that I regard as invariant. On the other hand I can see that someone whom I distrust, behave in ways that do not correspond to the invariant I have recorded for my trusted one. If the conditions above mentioned occur

I may distill a new principle that establishes correct and incorrect behaviours *learning by examples*. Analogous and extended aspects shall be derived from the observation of punishments and rewards associated to good and bad behaviours.

We shall also advance the study by adding functions related to the interaction processes to be modelled in this scenarios, by means of the social interactions in hybrid societies [48, 43]. That would provide room for another further aspect, similar to the one specified above: reputation. When trust is communicated in a public audience, it makes sense to value the communication, especially when it comes from someone, whom I trust in turn. The result of this combination shall result in a further extended model.

Bibliography

[1] Ruth Sara Aguilar-Savén. “Business process modelling: Review and framework”. In: *International Journal of Production Economics* 90.2 (2004). Production Planning and Control, pp. 129–149. ISSN: 0925-5273. DOI: https://doi.org/10.1016/S0925-5273(03)00102-6. URL: https://www.sciencedirect.com/science/article/pii/S0925527303001026.

[2] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. “On the Logic of Theory Change: Partial Meet Contraction and Revision Functions”. In: *The Journal of Symbolic Logic* 50.2 (1985), pp. 510–530. ISSN: 00224812. URL: http://www.jstor.org/stable/2274239.

[3] G. Antoniou et al. “Representation Results for Defeasible Logic”. In: *ACM Transactions on Computational Logic* 2.2 (2001), pp. 255–287.

[4] Grigoris Antoniou. “Defeasible logic with dynamic priorities”. In: *Int. J. Intell. Syst.* 19.5 (2004), pp. 463–472.

[5] T. Athan et al. “LegalRuleML: Design Principles and Foundations”. In: 2015. ISBN: 978-3-319-21767-3.

[6] T. Athan et al. “LegalRuleML: Design Principles and Foundations”. In: *Reasoning Web. Web Logic Rules: 11th International Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures.* Ed. by Wolfgang Faber and Adrian Paschke. Cham: Springer International Publishing, 2015, pp. 151–188. ISBN: 978-3-319-21768-0. DOI: 10.1007/978-3-319-21768-0_6. URL: https://doi.org/10.1007/978-3-319-21768-0_6.

[7] S. Bandini, S. Calegari, and P. Radaelli. “Towards fuzzy ontology handling vagueness of natural languages”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4062 LNAI (2006), pp. 693–700.

[8] R. Bartolini et al. “Automatic classification and analysis of provisions in italian legal texts: A case study”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3292 (2004), pp. 593–604. DOI: 10.1007/978-3-540-30470-8_72. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-35048890166&doi=10.1007%2f978-3-540-30470-8_72&partnerID=40&md5=b00a0134f3542973f48e6da414eaf76a.

[9] D. Basin, S. Matthews, and L. Viganò. “Labelled propositional modal logics: Theory and practice”. In: *journal of Logic and Computation* 7.6 (1997). cited By 23, pp. 685–717. DOI: 10.1093/logcom/7.6.685.

[10] M. Benerecetti, P. Bouquet, and C. Ghidini. "Contextual reasoning distilled". In: *Journal of Experimental and Theoretical Artificial Intelligence* 12.3 (2000), pp. 279–305. DOI: 10.1080/09528130050111446.

[11] V.R. Benjamins et al. "Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications: Preface". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3369 LNAI (2005), pp. IX–X.

[12] B. Bennett. "Application of supervaluation semantics to vaguely defined spatial concepts". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2205 (2001), pp. 108–123. DOI: 10.1007/3-540-45424-1_8.

[13] B. Bennett. "Modes of concept definition and varieties of vagueness". In: *Applied Ontology* 1 (2005), pp. 17–26.

[14] B. Bennett. "A theory of vague adjectives grounded in relevant observables". In: *Proceedings of the International Workshop on Temporal Representation and Reasoning*. 2006, pp. 36–44.

[15] B. Bennett. "Spatial vagueness". In: *Studies in Fuzziness and Soft Computing* 256 (2010), pp. 15–47. DOI: 10.1007/978-3-642-14755-5_2.

[16] B. Bennett. "Possible worlds and possible meanings". In: *AAAI Spring Symposium - Technical Report*. Vol. SS-11-06. 2011, pp. 23–29.

[17] B. Bennett. "Possible worlds and possible meanings: A semantics for the interpretation of vague languages". In: *10th International Symposium on Logical Formalizations of Commonsense Reasoning, Commonsense 2011 - Proceedings*. 2019.

[18] B. Bennett and P. Agarwal. "Semantic categories underlying the meaning of place". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4736 LNCS (2007), pp. 78–95. DOI: 10.1007/978-3-540-74788-8_6.

[19] B. Bennett, D. Mallenby, and A. Third. "An ontology for grounding vague geographic terms". In: *Frontiers in Artificial Intelligence and Applications* 183.1 (2008), pp. 280–293. DOI: 10.3233/978-1-58603-923-3-280.

[20] H. Bhuiyan et al. "A methodology for encoding regulatory rules". In: *CEUR Workshop Proceedings* 2632 (2020). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85094947107&partnerID=40&md5=ba8ebd6e3c2aa0bc60d250a5ab3c87dc.

[21] A. Bikakis and G. Antoniou. "Defeasible contextual reasoning with arguments in ambient intelligence". In: *IEEE Transactions on Knowledge and Data Engineering* 22.11 (2010), pp. 1492–1506. DOI: 10.1109/TKDE.2010.37.

[22] D.M. Blei. "Probabilistic topic models". In: *Communications of the ACM* 55.4 (2012), pp. 77–84. DOI: 10.1145/2133806.2133826. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84861170800&doi=10.1145%2f2133806.2133826&partnerID=40&md5=f7ccdd25a9092952bb8fb1f231102a86.

[23] Guido Boella et al. "A Logical Understanding of Legal Interpretation". In: *Proceedings of KR 2010*. 2010.

[24] Guido Boella et al. "*Lex minus dixit quam voluit*, *lex magis dixit quam voluit*: A formal study on legal compliance and interpretation". In: *AI approaches to the complexity of legal systems*. Berlin: Springer, 2010.

[25] I. Bordino et al. "GarNLP: A Natural Language Processing Pipeline for Garnishment Documents". In: *Information Systems Frontiers* (2020). DOI: 10.1007/s10796-020-09997-0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082753494&doi=10.1007%2fs10796-020-09997-0&partnerID=40&md5=5be6d34454ae6617433434709fcef129.

[26] M. Bourahla. "Reasoning over vague concepts". In: vol. 9120. 2015, pp. 591–602. DOI: 10.1007/978-3-319-19369-4_52.

[27] T. Brett. "Understanding organizations and institutions". In: *Managing Development: Understanding Inter-Organizational Relationships* (1999), pp. 17–48.

[28] J. Breuker, A. Valente, and R. Winkels. "Legal ontologies in knowledge engineering and information management". In: *Artificial Intelligence and Law* 12.4 (2004), pp. 241–277. DOI: 10.1007/s10506-006-0002-1.

[29] E. Burato and M. Cristani. "Learning as meaning negotiation: A model based on English auction". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5559 LNAI (2009), pp. 60–69.

[30] E. Burato and M. Cristani. "The process of reaching agreement in meaning negotiation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7270 LNCS (2012), pp. 1–42.

[31] E. Burato, M. Cristani, and L. Viganò. "A deduction system for meaning negotiation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6619 LNAI (2011), pp. 78–95.

[32] H. Burnett. "Penumbral connections in comparative constructions". In: *Journal of Applied Non-Classical Logics* 24.1-2 (2014), pp. 35–60. DOI: 10.1080/11663081.2014.911519.

[33] E. Cambria and B. White. "Jumping NLP curves: A review of natural language processing research". In: *IEEE Computational Intelligence Magazine* 9.2 (2014), pp. 48–57. DOI: 10.1109/MCI.2014.2307227.

[34] J.J. Camilleri, N. Gruzitis, and G. Schneider. "Extracting formal models from normative texts". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9612 (2016), pp. 403–408. DOI: 10.1007/978-3-319-41754-7_40.

[35] K. Carlson, M. Livermore, and D. Rockmore. "A Quantitative Analysis of Writing Style on the U.S. Supreme Court". In: *SSRN Electronic Journal* (2015). DOI: 10.2139/ssrn.2554516.

[36] D.J. Carter, J. Brown, and A. Rahmani. "Reading the High Court at a Distance: Topic Modelling the Legal Subject Matter And Judicial Activity of the High Court of Australia, 1903–2015". In: *University of New South Wales Law Journal* 39.4 (2016). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3564387.

[37] I. Chalkidis, I. Androutsopoulos, and A. Michos. "Extracting contract elements". In: *Proceedings of the International Conference on Artificial Intelligence and Law* (2017), pp. 19–28. DOI: 10.1145/3086512.3086515. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85038035195&doi=10.1145%2f3086512.3086515&partnerID=40&md5=40008ef8cdf22cec5a592e8600dd9719.

[38] Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Cambridge: Academic Press, 1973.

[39] R. Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12 (2011), pp. 2493–2537.

[40] G. Contissa et al. "Automated processing of privacy policies under the EU general data protection regulation". In: *Frontiers in Artificial Intelligence and Applications* 313 (2018), pp. 51–60. DOI: 10.3233/978-1-61499-935-5-51. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059624819&doi=10.3233%2f978-1-61499-935-5-51&partnerID=40&md5=d69a1bf9e604a26cdd269092454fceb9.

[41] O. Corcho et al. "Building legal ontologies with METHONTOLOGY and WebODE". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3369 LNAI (2005), pp. 142–157. DOI: 10.1007/978-3-540-32253-5_9.

[42] M. Cristani. "The Complexity of Reasoning about Spatial Congruence". In: *Journal of Artificial Intelligence Research* 11 (1999), pp. 361–390.

[43] M. Cristani, D. Fogoroasi, and C. Tomazzoli. "Measuring homophily". In: vol. 1748. 2016.

[44] M. Cristani and N. Gabrielli. "Practical issues of description logics for spatial reasoning". In: vol. SS-09-02. 2009, pp. 5–10.

[45] M. Cristani, F. Olivieri, and A. Rotolo. "Changes to temporary norms". In: pp. 39–48. DOI: 10.1145/3086512.3086517.

[46] M. Cristani, L. Pasetto, and C. Tomazzoli. "Protecting the environment: A multi-agent approach to environmental monitoring". In: vol. 176. 2020, pp. 3636–3644. DOI: 10.1016/j.procs.2020.09.336.

[47] M. Cristani and A. Rotolo. "Meaning negotiation with defeasible logic". In: *Smart Innovation, Systems and Technologies* 74 (2018), pp. 67–76. DOI: 10.1007/978-3-319-59394-4_7.

[48] M. Cristani, C. Tomazzoli, and F. Olivieri. "Semantic social network analysis foresees message flows". In: vol. 1. 2016, pp. 296–303. DOI: 10.5220/0005832902960303.

[49] M. Cristani, C. Tomazzoli, and M. Zorzi. "Automatic generation of dictionaries: The journalistic lexicon case". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11606 LNAI (2019), pp. 744–752. DOI: 10.1007/978-3-030-22999-3_63.

[50] M. Cristani et al. "A simple algorithm for the lexical classification of comparable adjectives". In: *Procedia Computer Science*. Vol. 126. 2018, pp. 626–635. DOI: 10.1016/j.procS.2018.07.297.

[51] M. Cristani et al. "An ontology of changes in normative systems from an agentive viewpoint". In: *Communications in Computer and Information Science* 1233 CCIS (2020), pp. 131–142. DOI: 10.1007/978-3-030-51999-5_11. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088503306&doi=10.1007%2f978-3-030-51999-5_11&partnerID=40&md5=905f59a36b1ea5bf2e176171ab7e4f1f.

[52] Martin Davis. *The Universal Computer. The Road from Leibniz to Turing*. Mathematics/Logic/Computing Series. Turing Centenary Edition. CRC Press, Taylor and Francis Group, 2012.

[53] E. De Maat and R. Winkels. "Automated classification of norms in sources of law". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6036 LNAI (2010), pp. 170–191. DOI: 10.1007/978-3-642-12837-0_10. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-77953781340&doi=10.1007%2f978-3-642-12837-0_10&partnerID=40&md5=3a2cb9f5205c7d298ae75ead7c2a47fb.

[54] M. Denecker, V.W. Marek, and M.S. Truszczyński. "Uniform semantic treatment of default and autoepistemic logics". In: *Artificial Intelligence* 143.1 (2003), pp. 79–122.

[55] M. Dragoni et al. "Combining Natural Language Processing Approaches for Rule Extraction from Legal Documents". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10791 (2018), pp. 287–300. DOI: 10.1007/978-3-030-00178-0_19.

[56] D. Dubois, H. Prade, and J.-P. Rossazza. "Vagueness, typicality, and uncertainty in class hierarchies". In: *International journal of Intelligent Systems* 6.2 (1991), pp. 167–183. DOI: 10.1002/int.4550060205.

[57] I. Düntsch. "Small integral relation algebras generated by a partial order". In: *Periodica Mathematica Hungarica* 23.2 (1991), pp. 129–138. DOI: 10.1007/BF02280664.

[58] G. Ferraro et al. "Automatic Extraction of Legal Norms: Evaluation of Natural Language Processing Tools". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12331 LNAI (2020), pp. 64–81. DOI: 10.1007/978-3-030-58790-1_5.

[59] J. Frankenreiter and M.A. Livermore. "Computational methods in legal analysis". In: *Annual Review of Law and Social Science* 16 (2020), pp. 39–57. DOI: 10.1146/annurev-lawsocsci-052720-121843. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85093067485&doi=10.1146%2fannurev-lawsocsci-052720-121843&partnerID=40&md5=ad229be390d750890c25856037db2242.

[60] F. Galgani, P. Compton, and A. Hoffmann. "Combining Different Summarization Techniques for Legal Text". In: *Proceedings of the workshop on innovative hybrid approaches to the processing of textual data* (2012), pp. 115–123.

[61] A. Gangemi, M.-T. Sagri, and D. Tiscornia. "A constructive framework for legal ontologies". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3369 LNAI (2005), pp. 97–124. DOI: 10.1007/978-3-540-32253-5_7.

[62] A. Gangemi et al. "Sweetening ontologies with DOLCE". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2473 (2002), pp. 166–181.

[63] A. Gangemi et al. "Adjective semantics in open knowledge extraction". In: *Frontiers in Artificial Intelligence and Applications* 283 (2016), pp. 167–180. DOI: 10.3233/978-1-61499-660-6-167.

[64] M. Gasmi and M. Bourahla. "Reasoning with vague concepts in description logics". In: *International Journal of Fuzzy System Applications* 6.2 (2017), pp. 43–58. DOI: 10.4018/IJFSA.2017040103.

[65] C. Ghidini and F. Giunchiglia. "Local Models Semantics, or contextual reasoning = locality + compatibility". In: *Artificial Intelligence* 127.2 (2001), pp. 221–259. DOI: 10.1016/S0004-3702(01)00064-9.

[66] F. Giunchiglia and L. Serafini. "Multilanguage hierarchical logics, or: How we can do without modal logics". In: *Artificial Intelligence* 65.1 (1994), pp. 29–70.

[67] P.A. Gloor et al. "Web science 2.0: Identifying trends through semantic social network analysis". In: *Proceedings - 12th IEEE International Conference on Computational Science and Engineering, CSE 2009*. Vol. 4. 2009, pp. 215–222. DOI: 10.1109/CSE.2009.186.

[68] T. Gonçalves and P. Quaresma. "Is linguistic information relevant for the classification of legal texts?" In: *Proceedings of the International Conference on Artificial Intelligence and Law* (2005), pp. 168–176. DOI: 10.1145/1165485.1165512. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-67849120190&doi=10.1145%2f1165485.1165512&partnerID=40&md5=006b8423854dca0c376361dabec7456c.

[69] G. Governatori, Z. Milosevic, and S. Sadiq. "Compliance checking between business processes and business contracts". In: 2006, pp. 221–230. DOI: 10.1109/EDOC.2006.22. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-38749096258&doi=10.1109%2fEDOC.2006.22&partnerID=40&md5=dc1a47a514d96ff13bb0eeeb7de93e34.

[70] G. Governatori and A. Rotolo. "Changing legal systems: Legal abrogations and annulments in Defeasible Logic". In: *Logic Journal of the IGPL* 1 (), pp. 157–194. DOI: 10.1093/jigpal/jzp075.

[71] G. Governatori and S. Sadiq. *The journey to business process compliance*. 2009, pp. 426–454. DOI: 10.4018/978-1-60566-288-6.ch020. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84898320543&doi=10.4018%2f978-1-60566-288-6.ch020&partnerID=40&md5=13350151ef72124f57a3bae60dcef453.

[72] G. Governatori et al. "Superiority based revision of defeasible theories". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6403 LNCS (2010), pp. 104–118. DOI: 10.1007/978-3-642-16289-3-10.

[73] G. Governatori et al. "Designing for compliance: Norms and goals". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7018 LNCS (2011), pp. 282–297. DOI: 10.1007/978-3-642-24908-2_29.

[74] G. Governatori et al. "Computing Strong and Weak Permissions in Defeasible Logic". In: *Journal of Philosophical Logic* 42.6 (2013), pp. 799–829. DOI: 10.1007/s10992-013-9295-1.

[75] G. Governatori et al. "Picking up the best goal an analytical study in defeasible logic". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8035 (2013), pp. 99–113. DOI: 10.1007/978-3-642-39617-5_12.

[76] G. Governatori et al. "The hardness of revising defeasible preferences". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8620 LNCS (2014), pp. 168–177. DOI: 10.1007/978-3-319-09870-8_12.

[77] G. Governatori et al. "Revision of defeasible preferences". In: *International Journal of Approximate Reasoning* 104 (2019), pp. 205–230. DOI: 10.1016/j.ijar.2018.10.020.

[78] G. Governatori et al. "Variants of temporal defeasible logics for modelling norm modifications". In: pp. 155–159. DOI: 10.1145/1276318.1276347.

[79] Guido Governatori and Antonino Rotolo. "Changing Legal Systems: Legal Abrogations and Annulments in Defeasible Logic". In: *Logic Journal of the IGPL* 18.1 (2010), pp. 157–194.

[80] N. Guarino and G. Guizzardi. "Relationships and events: Towards a general theory of reification and truthmaking". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10037 LNAI (2016), pp. 237–249. DOI: 10.1007/978-3-319-49130-1_18.

[81] B. Hachey and C. Grover. "Extractive summarisation of legal texts". In: *Artificial Intelligence and Law* 14.4 (2006), pp. 305–345. DOI: 10.1007/s10506-007-9039-z. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-34047209272&doi=10.1007%2fs10506-007-9039-z&partnerID=40&md5=49071e499dd91246d4a0f9c451945041.

[82] J.A. Hampton. "Typicality, graded membership, and vagueness". In: *Cognitive Science* 31.3 (2007), pp. 355–384. DOI: 10.1080/15326900701326402.

[83] L. A. Hannikainen. *Peremptory Norm (Jus Cogens) in International Law - Historical Development, Criteria, Present Status*. 1988.

[84] Lucja Iwanska. "Reasoning with intensional negative adjectivals: Semantics, pragmatics, and context". In: *Computational Intelligence* 13.3 (1997), pp. 348–390. DOI: 10.1111/0824-7935.00043.

[85] D. Jouve et al. "Conceptual framework for document semantic modelling: An application to document and knowledge management in the legal domain". In: *Data and Knowledge Engineering* 46.3 (2003), pp. 345–375. DOI: 10.1016/S0169-023X(03)00046-6.

[86] J.J. Jung and J. Euzenat. "Towards semantic social networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 4519 LNCS. 2007, pp. 267–280.

[87] J. A. W. Kamp and Edward L. Keenan. "Two theories about adjectives". In: *Formal Semantics of Natural Language*. Cambridge University Press, 1975, pp. 123–155. DOI: 10.1017/CBO9780511897696.011.

[88] D.M. Katz and II Bommarito M.J. "Measuring the complexity of the law: the United States Code". In: *Artificial Intelligence and Law* 22.4 (2014), pp. 337–374. DOI: 10.1007/s10506-014-9160-8. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84920953190&doi=10.1007%2fs10506-014-9160-8&partnerID=40&md5=7a11f76cdd20edf631e6f957af711de1.

[89] D.M. Katz et al. "Complex societies and the growth of the law". In: *Scientific Reports* 10.1 (2020). DOI: 10.1038/s41598-020-73623-x. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85094680800&doi=10.1038%2fs41598-020-73623-x&partnerID=40&md5=8bf4857e01026ee660ef2d809fcdbadc.

[90] E. Klein. "A semantics for positive and majority comparative adjectives". In: *Linguistics and Philosophy* 4.1 (1980), pp. 1–45. DOI: 10.1007/BF00351812.

[91] M. Koniaris, G. Papastefanatos, and I. Anagnostopoulos. "Solon: A holistic approach for modelling, managing and mining legal sources". In: *Algorithms* 11.12 (2018). DOI: 10.3390/a11120196. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059285468&doi=10.3390%2fa11120196&partnerID=40&md5=74d2d93a8286ca3de4fd425f77058c3b.

[92] F. Lagioia et al. "Deep learning for detecting and explaining unfairness in consumer contracts". In: *Frontiers in Artificial Intelligence and Applications* 322 (2019), pp. 43–52. DOI: 10.3233/FAIA190305. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082136211&doi=10.3233%2fFAIA190305&partnerID=40&md5=248c6e244b99b655d6a5ea6710991f05.

[93] H.-P. Lam and M. Hashmi. "Enabling reasoning with LegalRuleML". In: *Theory and Practice of Logic Programming* 19.1 (2019), pp. 1–26. DOI: 10.1017/S1471068418000339.

[94] G. Lame. "Using NLP techniques to identify legal ontology components: Concepts and relations". In: *Artificial Intelligence and Law* 12.4 (2004), pp. 379–396. DOI: 10.1007/s10506-005-4160-3.

[95] G. Lame. "Using NLP techniques to identify legal ontology components: Concepts and relations". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3369 LNAI (2005), pp. 169–184. DOI: 10.1007/978-3-540-32253-5_11.

[96] D. Lassiter. "Vagueness as probabilistic linguistic knowledge". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6517 LNAI (2011), pp. 127–150. DOI: 10.1007/978-3-642-18446-8-8.

[97] Z. Li. "A Classification Retrieval Approach for English Legal Texts". In: *Proceedings - 2019 International Conference on Intelligent Transportation, Big Data and Smart City, ICITBS 2019* (2019), pp. 220–223. DOI: 10.1109/ICITBS.2019.00059. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063902757&doi=10.1109%2fICITBS.2019.00059&partnerID=40&md5=20132d5e7bdf65ef4d76478bbf1c9a67.

[98] M. Lippi et al. "Automated detection of unfair clauses in online consumer contracts". In: *Frontiers in Artificial Intelligence and Applications* 302 (2017), pp. 145–154. DOI: 10.3233/978-1-61499-838-9-145. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85037993346&doi=10.3233%2f978-1-61499-838-9-145&partnerID=40&md5=24da4b0af24430e9732f4e9874d85e1b.

[99] R. Lu, S. Sadiq, and G. Governatori. "Compliance aware business process design". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4928 LNCS (2008), pp. 120–131. DOI: 10.1007/978-3-540-78238-4_14. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-40549095696&doi=10.1007%2f978-3-540-78238-4_14&partnerID=40&md5=836ca8369657b9dc4d3f1d878f647d22.

[100] T. Lukasiewicz and U. Straccia. "Managing uncertainty and vagueness in description logics for the Semantic Web". In: *Web Semantics* 6.4 (2008), pp. 291–308. DOI: 10.1016/j.websem.2008.04.001.

[101] A. Meštrović and A. Calì. "An ontology-based approach to information retrieval". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10151 LNCS (2017), pp. 150–156. DOI: 10.1007/978-3-319-53640-8_13.

[102] Sanjay Modgil. "Hierarchical Argumentation". In: *JELIA*. 2006, pp. 319–332.

[103] Martín O. Moguillansky et al. "Dynamics of knowledge in *DeLP* through Argument Theory Change". In: *TPLP* 13.6 (2013), pp. 893–957.

[104] M.S. Moore. "Legal reality: A naturalist approach to legal ontology". In: *Law and Philosophy* 21.6 (2002), pp. 619–705. DOI: 10.1023/A:1021967205087.

[105] L.S. Moss. "Intersecting adjectives in syllogistic logic". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6149 LNAI.M4D (2010), pp. 223–237. DOI: 10.1007/978-3-642-14322-9_17.

[106] J.C. Nardi et al. "A commitment-based reference ontology for services". In: *Information Systems* 54 (2015), pp. 263–288. DOI: 10.1016/j.is.2015.01.012.

[107] A. Nazarenko, F. Levy, and A. Wyner. "Towards a methodology for formalizing legal texts in LegalRuleML". In: *Frontiers in Artificial Intelligence and Applications* 294 (2016), pp. 149–154. DOI: 10.3233/978-1-61499-726-9-149.

[108] D. Nute. *Defeasible Deontic Logic*. 1st. Springer Publishing Company, Incorporated, 2010. ISBN: 904814874X.

[109] F. Olivieri, M. Cristani, and G. Governatori. "Compliant business processes with exclusive choices from agent specification". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9387 (2015), pp. 603–612. DOI: 10.1007/978-3-319-25524-8_43.

[110] F. Olivieri et al. "Compliant business process design by declarative specifications". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8291 LNAI (2013), pp. 213–228. DOI: 10.1007/978-3-642-44927-7_15. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84893105020&doi=10.1007%2f978-3-642-44927-7_15&partnerID=40&md5=8afb2304ac0933c6ff3bb67940c21dcf.

[111] Z. Pawlak. "VAGUENESS AND UNCERTAINTY: A ROUGH SET PERSPECTIVE". In: *Computational Intelligence* 11.2 (1995), pp. 227–232. DOI: 10.1111/j.1467-8640.1995.tb00029.x.

[112] Charles Petzold. *The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine*. Wiley Publishing, 2008.

[113] Henry Prakken and Giovanni Sartor. "Argument-Based Extended Logic Programming with Defeasible Priorities". In: *Journal of Applied Non-Classical Logics* 7.1 (1997).

[114] S. Ramakrishna and A. Paschke. "Semi-automated vocabulary building for structured legal English". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8620 LNCS (2014), pp. 201–215. DOI: 10.1007/978-3-319-09870-8_15.

[115] J.B. Ruhl, D.M. Katz, and II Bommarito M.J. "Harnessing legal complexity". In: *Science* 355.6332 (2017), pp. 1377–1378. DOI: 10.1126/science.aag3013. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85016610340&doi=10.1126%2fscience.aag3013&partnerID=40&md5=41da522404d902ab697ea94c06e4dc42.

[116] V. Santos Costa. "On just in time indexing of dynamic predicates in prolog". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5816 LNAI (2009), pp. 126–137.

[117] G.W. Sassoon. "Measurement theory in linguistics". In: *Synthese* 174.1 (2010), pp. 151–180. DOI: 10.1007/s11229-009-9687-5.

[118] G.W. Sassoon. "The degree functions of negative adjectives". In: *Natural Language Semantics* 18.2 (2010), pp. 141–181. DOI: 10.1007/s11050-009-9052-8.

[119] G.W. Sassoon. "Adjectival vs. nominal categorization processes: The rule vs. similarity hypothesis". In: *Belgian Journal of Linguistics* 25.1 (2011), pp. 104–147. DOI: 10.1075/bjl.25.06sas.

[120] G.W. Sassoon. "A typology of multidimensional adjectives". In: *Journal of Semantics* 30.3 (2013), pp. 335–380. DOI: 10.1093/jos/ffs012.

[121] G.W. Sassoon. “Vagueness, gradability and typicality”. In: *Current Research in the Semantics/Pragmatics Interface* 27 (2013), pp. 1–446.

[122] S. Scannapieco et al. “A methodology for plan revision under norm and outcome compliance”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8291 LNAI (2013), pp. 324–339. DOI: 10.1007/978-3-642-44927-7_22.

[123] L. Serafini and P. Bouquet. “Comparing formal theories of context in AI”. In: *Artificial Intelligence* 155.1-2 (2004), pp. 41–67. DOI: 10.1016/j.artint.2003.11.001.

[124] P. Thompson. “Automatic categorization of case law”. In: *Proceedings of the International Conference on Artificial Intelligence and Law* (2001), pp. 70–77. DOI: 10.1145/383535.383543. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0012051194&doi=10.1145%2f383535.383543&partnerID=40&md5=ee93350cb8394851347624d23ba8acc0.

[125] Alan M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society* 2.42 (1936), pp. 230–265.

[126] Peter D. Turney. “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”. In: 2002. URL: http://arxiv.org/abs/cs/0212032.

[127] S. Urchs, J. Mitrovic, and M. Granitzer. “Towards Classifying Parts of German Legal Writing Styles in German Legal Judgments”. In: *2020 10th International Conference on Advanced Computer Information Technologies, ACIT 2020 - Proceedings* (2020), pp. 451–454. DOI: 10.1109/ACIT49673.2020.9208956. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85094150073&doi=10.1109%2fACIT49673.2020.9208956&partnerID=40&md5=c2796eceb8164366968fb40ba7a54bcf.

[128] C. Uyttendaele, M.-F. Moens, and J. Dumortier. “SALOMON: Automatic Abstracting of Legal Cases for Effective Access to Court Decisions”. In: *Artificial Intelligence and Law* 6.1 (1998), pp. 59–79. DOI: 10.1023/A:1008256030548. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0032022293&doi=10.1023%2fA%3a1008256030548&partnerID=40&md5=0e2cbff367c22149bbb5cb102b3e837d.

[129] A. Valente. “Types and roles of legal ontologies”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3369 LNAI (2005), pp. 65–76. DOI: 10.1007/978-3-540-32253-5_5.

[130] J. Van Benthem. “Dynamic logic for belief revision”. In: *Journal of Applied Non-Classical Logics* 17.2 (2007), pp. 129–155. DOI: 10.3166/jancl.17.129-155.

[131] A. Wyner and G. Governatori. “A study on translating regulatory rules from natural language to defeasible logic”. In: vol. 1004. 2013.

[132] A. Wyner and W. Peters. “On rule extraction from regulations”. In: *Frontiers in Artificial Intelligence and Applications* 235 (2011), pp. 113–122. DOI: 10.3233/978-1-60750-981-3-113.