UNIVERSITÀ DEGLI STUDI DI BOLOGNA

DOTTORATO DI RICERCA IN INFORMATICA GIURIDICA E DIRITTO DELL'INFORMATICA

CICLO XX

TESI DI RÉGIS RIVERET

INTERACTIONS BETWEEN NORMATIVE SYSTEMS AND SOFTWARE COGNITIVE AGENTS.

A FORMALIZATION IN TEMPORAL MODAL DEFEASIBLE LOGIC.

Coordinatore

Chiar.mo Prof. Enrico Pattaro

Tutor

Chiar.ma Prof.ssa Monica Palmirani

Chiar.mo Prof. Antonino Rotolo

anno 2008

Preface

First of all, I would like to thank Enrico Pattaro for giving me the chance to work at CIRSIFD and for his help in the hardest moments of my health problems at the beginning of my thesis. I am also thankful to Giovanni Sartor for illuminating philosophical inputs and the opening of new horizons. I am grateful to my supervisor Monica Palmirani for giving me the opportunity to become a researcher in computer science and law, and for her constant belief in me.

I am deeply indebted to Antonino Rotolo for his guidance, for helping me and for having always been available. Without Nino, this thesis could have not been produced. I am also grateful to Guido Governatori for insightful comments on defeasible logic.

I thank all my room mates. Special thanks to Abraham Roth for all the discussions on (the philosophy of) sciences, and his friendly encouragements. I thank Andrea Resmini and Luca Cervone for their enthusiastic guidance on Web technologies and Alberto Artosi for well-disposed conversations. I am also thankful to the Norma group, the AI&Law group, and CIRSFID colleagues for setting up a stimulating environment. There are too many names to mention, but I must make an exception for Tazia Bianchi who provided me with administrative support throughout these years.

Naturally, I thank my parents, who have been a constant source of help in all aspects of my life, and my brother, Guillaume, who sowed the seeds of my education in computer science. Last but not the least, I owe Gloria for her love and for reminding me how relative things are.

Bologna,
30 April 2008 *Régis Riveret*

Summary

Sustainable computer systems require some flexibility to adapt to environmental unpredictable changes. A solution lies in autonomous software agents which can adapt autonomously to their environments. Though autonomy allows agents to decide which behavior to adopt, a disadvantage is a lack of control, and as a side effect even untrustworthiness: we want to keep some control over such autonomous agents. How to control autonomous agents while respecting their autonomy?

A solution is to regulate agents' behavior by norms. The normative paradigm makes it possible to control autonomous agents while respecting their autonomy, limiting untrustworthiness and augmenting system compliance. It can also facilitate the design of the system, for example, by regulating the coordination among agents. However, an autonomous agent will follow norms or violate them in some conditions. What are the conditions in which a norm is binding upon an agent?

While autonomy is regarded as the driving force behind the normative paradigm, cognitive agents provide a basis for modeling the bindingness of norms. In order to cope with the complexity of the modeling of cognitive agents and normative bindingness, we adopt an intentional stance.

Since agents are embedded into a dynamic environment, things may not pass at the same instant. Accordingly, our cognitive model is extended to account for some temporal aspects. Special attention is given to the temporal peculiarities of the legal domain such as, among others, the time in force and the time in efficacy of provisions. Some types of normative modifications are also discussed in the framework. It is noteworthy that our temporal account of legal reasoning is integrated to our commonsense temporal account of cognition.

As our intention is to build sustainable reasoning systems running unpredictable environment, we adopt a declarative representation of knowledge. A declarative representation of norms will make it easier to update their system representation, thus facilitating system maintenance; and to improve system transparency, thus easing system governance.

Since agents are bounded and are embedded into unpredictable environments, and since conflicts may appear amongst mental states and norms, agent reasoning has to be defeasible, i.e. new pieces of information can invalidate formerly derivable conclusions. In this dissertation, our model is formalized into a non-monotonic logic, namely into a temporal modal defeasible logic, in order to account for the interactions between normative systems and software cognitive agents.

Contents

Part I Systemization

1 Norm-governed computer systems . . . 3
1.1 Toward autonomy . . . 3
1.2 Toward norms . . . 4
1.3 Toward cognitive agents . . . 5
1.4 Toward declarative knowledge representation and defeasible reasoning . . . 6
1.5 An example of a norm-governed computer system: E-government . . 7
1.6 Plan of thesis . . . 8

2 Computer science and law . . . 11
2.1 Legal knowledge management systems . . . 11
2.2 Artificial intelligence and law . . . 15
2.2.1 Which AI paradigms for legal reasoning? . . . 15
2.2.2 Legal knowledge-based systems . . . 17
2.3 Prospects for computer science and law . . . 19

3 Agent technologies . . . 21
3.1 The concept of agent . . . 21
3.1.1 What is an agent? . . . 21
3.1.2 Agent architectures . . . 23
3.2 Multi-agent systems . . . 25
3.2.1 Communication . . . 26
3.2.2 Coordination . . . 27
3.2.3 Normative multi-agent systems . . . 29
3.3 Agents and other computer paradigms . . . 32
3.3.1 Relationship between agents and the client/server model . . . 32
3.3.2 Relationship between agents and objects . . . 33
3.3.3 Relationship between agents and expert systems . . . 34

Part II Informal

4 Reasoning framework . . . 37
4.1 Defeasible reasoning . . . 37
4.2 Arguments . . . 41
4.3 Defeaters and defeated . . . 44
4.3.1 Defeaters . . . 44
4.3.2 Conflicts handling . . . 46
4.4 Justified arguments . . . 47

5 Cognitive model . . . 51
5.1 Epistemic cognition . . . 51
5.1.1 (Dis)beliefs and (in)existences . . . 52
5.1.2 Detachment schema in epistemic reasoning . . . 53
5.1.3 Introspection and reflexion . . . 54
5.2 Practical cognition . . . 56
5.2.1 Desires . . . 57
5.2.2 Actions . . . 58
5.2.3 Plans and intentions . . . 63
5.3 Normative cognition . . . 64
5.3.1 Basic deontic concepts . . . 64
5.3.2 Normative conditionals . . . 66
5.3.3 Relations between deontic concepts . . . 66
5.3.4 Strong and weak permissions . . . 70
5.4 Reasoning schemata . . . 72
5.4.1 Combining modalities . . . 72
5.4.2 Detachment schema . . . 74
5.4.3 Rule conversion . . . 75
5.4.4 Introspection and reflexion . . . 76
5.5 Collisions of conclusions . . . 77
5.5.1 Conflicts . . . 77
5.5.2 Preferences . . . 79

6 Temporal model . . . 85
6.1 Temporal statements . . . 85
6.1.1 Temporal modal literals . . . 86
6.1.2 Temporal modal rules . . . 87
6.1.3 Legal temporal dimensions . . . 90
6.2 Temporal reasoning schemata . . . 93
6.2.1 Temporal detachment schemata . . . 93
6.2.2 General reasoning schema . . . 97
6.2.3 Introspection and reflexion with time . . . 98
6.3 Collisions of temporal conclusions . . . 100
6.3.1 Conflicts . . . 100

6.3.2 Preferences . . . 101
6.4 Normative modifications . . . 103
6.4.1 Categorization of modifications . . . 106
6.4.2 Temporal dimensions of modifications . . . 107
6.4.3 Temporal modifications . . . 108
6.4.4 Textual modifications . . . 110

Part III Formal

7 Logic formalization . . . 117
7.1 Logic and law . . . 117
7.1.1 Legal reasoning as deductive reasoning . . . 117
7.1.2 Logic as a tool to remove impreciseness . . . 120
7.1.3 Logic for norm-governed computer systems . . . 121
7.2 Classical logics . . . 122
7.2.1 Classical propositional logic . . . 122
7.2.2 First-order predicate logic . . . 126
7.3 Modal logics . . . 130
7.3.1 Possible world semantics . . . 130
7.3.2 Epistemic logic . . . 133
7.3.3 Deontic logic . . . 134
7.4 Non-monotonic logics . . . 135
7.4.1 Circumscription . . . 136
7.4.2 Default logic . . . 138
7.4.3 Logic programming . . . 140
7.4.4 Argumentation . . . 144
7.5 Time and non-monotonic logics . . . 146
7.5.1 Situation calculus . . . 146
7.5.2 Event calculus . . . 149

8 Defeasible logic . . . 153
8.1 Language . . . 154
8.2 Proof theory . . . 155
8.3 Argument semantics . . . 161
8.3.1 Arguments . . . 161
8.3.2 Acceptable arguments . . . 163
8.3.3 Justified arguments . . . 165

9 Modal defeasible logic . . . 169
9.1 Language . . . 169
9.2 Proof theory . . . 171
9.3 Argumentation semantics . . . 180
9.3.1 Arguments . . . 180
9.3.2 Acceptable arguments . . . 182

9.3.3 Justified arguments 184

10 Temporal modal defeasible logic 187
10.1 Language 187
10.2 Proof theory 190
10.3 Argumentation semantics 197
10.3.1 Arguments 197
10.3.2 Acceptable arguments 200
10.3.3 Justified arguments 202
10.4 A Prolog meta-program 203

11 Conclusions 211
11.1 Results 211
11.2 Prospects and limits 212

References 215

Part I

Systemization

1

Norm-governed computer systems

On the one hand, computer systems are driven by well-defined instructions for accomplishing some tasks. On the other hand, humans prescribe how the world ought to be by means of norms. While the use of norms to regulate computer behavior is a well-established idea , it is still a matter of research. This thesis investigates the use of norms to regulate computer behaviors.

In this Chapter, the motivation for this thesis is briefly provided. In Section 1.1, we root the research in sustainable computer systems via autonomous software agents, and in Section 1.2, we motive the use of norms to control autonomous agents. In Section 1.3, we assume a cognitive account of normative bindingness and commit to the intentional stance of such cognitive model. In Section 1.4, we argue for a declarative and defeasible representation of knowledge. In Section 1.5, an example of norm-governed computer system illustrates the vision. Finally, in Section 1.6, the outline of the dissertation is given.

1.1 Toward autonomy

Computers excel in automation, unfailingly executing instructions. However, in an ever-changing environment, executing automatically predefined instructions is not enough; computers are expected to derive their own tasks. To this regard, a step beyond automation is *autonomy*.

Autonomy for computer systems is not aimed at emulating biological agents but is rather a consequence of concrete environmental requirements. Indeed, as computer systems do not float in a vacuum, they are embedded into some environments whose properties influence features of such computer systems. For example, computer systems have evolved from centralized to distributed models in order to accompany the decentralization and cooperation of present-day organizations.

S. Russell and P. Norvig's investigation into the environment properties [185] provides us with some hints for autonomy requirements. Firstly, a system's environment may not be static but dynamic, that is, it may not be assumed to remain unchanged. Indeed, needs and preferences of human users may evolve. Some components may appear while others disappear. Other processes may operate on it and change it. Secondly, an environment may not be fully observable in space and in time, that is, the complete states may not be accessible. If an environment is not fully observable then it is partially observable and consequently the system usually runs on local information. From the point of view of an application embedded in an environment which is partially observable and dynamic, such environment may appear non-deterministic or unpredictable.

In such unpredictable environments, computer systems require some flexibility in order to adapt to changes and fulfill their initial assigned task. However, most aspects of the environments are often deeply buried in the code of the applications, and as a consequence most unpredictable changes require costly code maintenance or even the implementation from scratch of new applications. Sustainable computer systems should provide flexible means to adapt to environmental changes. How shall this vision of computing be achieved? Many people believe the answer lies in autonomous computer systems, or autonomous agents.

If autonomy is understood in terms of independence then different types of autonomy are identifiable: information autonomy shall refer to information independence, planning autonomy to planning independence, goal autonomy to goal independence and so on. If we consider control independence, then we refer to the principle that an agent can make decisions on its own, that is, roughly speaking, control is inside the agent and not outside. This relates well with the Greek etymology of term autonomous which literally refers to giving oneself his/her own law. In this view, autonomy enables computer systems to adapt to environmental changes “without the intervention of human beings or other agents” [216]. In the remainder, autonomous agents shall refer to control autonomous agents.

1.2 Toward norms

If we build autonomous software agents, we do not want to lose total control. In some scenarios, an uncontrolled system may involve untrustworthiness: if we build autonomous software agents, we would like to be able to trust them. A natural solution lies in controlling to some extent those agents. But, how can an autonomous artificial system be controlled while respecting its autonomy? Many people believe that an answer lies in the normative paradigm.

The term norm is used in many different disciplines as in the philosophy of law, sociology, psychology, linguistics etc., but there is no commonly accepted definition of the term norm. In this thesis, a norm is a directive binding upon some agents and serving to guide an ideal behavior. Following A. Ross in [181], a directive is expressed by an enunciate of the directive discourse.

On this basis, a norm is used as an external source of control and it is by means of norms that an autonomous agent shall be controlled from the outside, while respecting its autonomy. Being autonomous, an agent can decide to behave according to norms or violate them in some circumstances. Though norms may be violated, the normative paradigm shall augment compliance of autonomous agents, and thus limit untrustworthiness.

As norms are usually abstract they can also be addressed to multiple agents. Then, each agent shall instantiate abstract norms relatively to its own condition. Consequently, normative systems permit to control multiple (interacting) agents. As norms guide agents, they can be used to coordinate agent activities: in doing so, norms are used to control isolated agents to coordinate societies of multiple interacting agents.

In computer science, a system composed of multiple agents (usually in interaction) is called a multi-agent system. The research area defined as the interaction of normative systems and multi-agents systems aims at investigating *normative multi-agent systems*. A computer system which is based on a normative multi-agent system has the advantages of both agent paradigm and normative paradigm. In this view, an important perspective for norm-governed computer systems [106] is normative multi-agent systems.

If norms are used to control autonomous agents then an essential issue is the bindingness of norms. Why and how do agents adopt a norm? In which conditions is a norm binding upon an agent? In which conditions shall an agent violate it? Many people believe that an answer lies in cognitive agents.

1.3 Toward cognitive agents

In philosophy of law, a vast literature exists which attempts to understand why and how a norm shall be binding upon an agent. Among all the models accounting for normative bindingness, we are interested by cognitive models [191].

Cognitive agents are complex systems and consequently it is often difficult to predict and discuss their behavior using the laws of physics. One common alternative when discussing agents is to conceive them using mental concepts that are more usually applied to humans, such as beliefs, desires and intentions. The philosopher D. Dennett explains this vision in [62] by considering three stances that one can take when characterizing the behavior of a system. (i) In the physical stance, one starts with the original configuration of a system, and then uses the laws of physics to predict how this system will behave. (ii) In the design stance, one uses knowledge of the purpose a system is supposed to fulfill, and on this basis predicts how the system will behave; for example, one takes normally the design stance when discussing an alarm clock. (iii) In the intentional stance, predictions rely on mental explanations of behavior (e.g., beliefs, intentions, desires, and so on). This intentional stance is well explained by D. Dennett in [62]:

> Here is how it works: first you decide to treat the object whose behavior is to be predicted as a rational agent; then you figure out what beliefs that

> agent ought to have, given its place in the world and its purpose. Then you figure out what desires it ought to have, on the same considerations, and finally you predict that this rational agent will act to further its goals in the light of its beliefs. A little practical reasoning from the chosen set of beliefs and desires will in most instances yield a decision about what the agent ought to do; that is what you predict the agent will do.

Since the physical or design stances for the account of cognitive agents are beyond our scope, we will thus reduce our model of cognition to the intentional stance by making use of mental concepts.

While autonomy is regarded as the driven force to introduce the norm paradigm in computer systems, cognitive agents provide us with a basis on which to model the bindingness of norms. The field of normative multi-agent systems is an opportunity to investigate computable models of such bindingness. Central to such models is the account of violations for at least two reasons: firstly because some agents' (mental) attitudes may conflict with norms, secondly because the norms themselves may conflict in some circumstances. Depending on the behaviors of agents with respect to norms, the system may sanction violations or reward good behaviors.

1.4 Toward declarative knowledge representation and defeasible reasoning

Knowledge can be represented procedurally or declaratively. In the procedural view, the knowledge about a domain is intricate with the control of agents' reasoning process, and thus is implicitly represented. In the declarative view, the knowledge is explicitly represented and separated from the reasoning procedures.

If new unpredicted knowledge will have to be treated, then the reasoning procedures and the intricate knowledge may not be adapted and may not be easily adaptable: in unpredictable environment, the procedural view is not sustainable. On the contrary, in a declarative representation, as the knowledge to be treated is separated from the reasoning procedures, then, ideally, no modifications to the procedures have to be introduced. As our intention is to build sustainable reasoning systems running in an unpredictable environment, we shall favor the declarative view, and accordingly, information about states of affairs (including norms) shall be explicitly represented.

A clear and explicit representation of norms has nice implications: as norms may change over time, it shall facilitate the update of their system representation, and consequently system maintenance, improve system transparency, and thus ease system governance.

Beside procedural and declarative matters, the reasoning can be classically deductive or defeasible. A reasoning is defeasible when certain conclusions can be abandoned in light of further information, whereas a reasoning is classically deductive when the validity of conclusions are guaranteed even in light of further information. What is the most appropriate type of reasoning for our purposes?

To answer this question, it is important to remark that if the environment is unpredictable from the agent point of view, then the agent is not omniscient. As a consequence, new information can be communicated to the agents and conflict with formerly derived conclusions: in other words, certain conclusions may have to be abandoned in light of further information. Hence, it seems that defeasible reasoning is necessary for our purposes to build sustainable computer systems, and that deductive reasoning has to be discarded.

From a more technical point of view, the requirement for declarative representation of knowledge coupled with defeasible reasoning usually ends up in the choice of a non-monotonic logic to formalize ratiocinative process of software agents. We shall return to the choice and development of an appropriate non-monotonic logic more in depth in the next Parts. Before, in the next Chapters we shall connect our research agenda with respect to the larger field of computer sciences & law. First, in the following Section, we illustrate the idea of norm-governed computer system in electronic government.

1.5 An example of a norm-governed computer system: E-government

The so-called electronic government (e-government) is the use of information and communication technology in public administrations. Combined with organizational change, it has the potential to significantly improve public services and strengthen democracy by improving the two-way communication between the citizens and their government. The widespread deployment and use of e-government services can result in substantially reduced costs for both businesses and governments, thereby lowering taxes and boosting competitiveness.

Despite the emergence of best practices, many barriers remain before the take-up of e-government can be achieved. An important obstacle concerns costs maintenance. Indeed, in a fast changing political and economical environment, computer systems have to be continually updated. As the execution of complex services may involve multiple administrative agencies in coordination, software's adaptations often run into expensive costs. Efforts are thus oriented to reduce cost maintenance, for example by adopting standards.

A solution for sustainable e-government systems is to provide more flexibility to applications, namely by means of some autonomy. For example, such infrastructure shall ease the discovery of administrative electronic services by autonomous software agents to facilitate the provision of up-to-date applications.

However, autonomous e-government agents have to be kept under control. An elegant solution is to use the administrative normative framework (e.g. provisions included in the law corpora) to guide their behaviors since it is primary for the correct conduct of administrative information systems. As the complex coordination of multiple and distributed administrative agencies is usually prescribed by some provisions, the digital representation of formalized provisions can be used to guide the coordination of distributed software agents acting on behalf of administrations. Of course, some administrative tasks delegated to software agents may be incompatible with the normative framework, and some unpredictable conflicts between norms themselves shall occur: cognitive software agents will permit an account of normative bindingness. For example, intentions of cognitive agents acting one behalf of some local autonomous administrations may conflict with state legislations.

The environment as normative knowledge changes over time: a declarative representation of knowledge facilitates the update of the system, facilitates maintenance, improve system transparency, and thus system governance by human administrators.

Taking into account an enormous number of public services and dependencies between them, as well as the complexity of interpreting and implementing changes in government regulations, an efficient e-government solution must provide the possibility of adapting progressively, while ensuring its compliance with the regulation. A solution lies in the concept of normative multi-agent systems. While this is true for public organization where information systems have to comply with internal and external regulatory provisions, this is also true for private organizations where information systems have to implement business rules. Indeed, it is often the case that policies are related to how companies should carry out their business. For example, these policies are designed to ensure that they pay relevant taxes, to treat customers and suppliers fairly, to not have an adverse impact on the environment, or to provide reliable information for investors and creditors.

1.6 Plan of thesis

The thesis is divided in three Parts. The first Part, which includes Chapters 1, 2 and 3, aims at providing a systemization of the thesis with respect to computer science and law. Chapters 1 and 2 overview some tools to build norm-governed computer systems. Chapter 2 presents briefly possible relevant aspects of legal knowledge management and artificial intelligence & law. Chapter 3 introduces the notion of software agent and related concepts, in particular normative multi-agent systems.

The second Part of the thesis, that is, Chapters 4, 5 and 6, is dedicated to presenting informally the framework of interaction between normative systems and software cognitive agents. Chapter 4 provides a general and abstract model of defeasible reasoning. This model is further developed in Chapter 5 to account for the combinations of mental, agency and deontic concepts as believes, desires, actions, obligations etc. Chapter 6 extends the framework to capture some temporal aspects and normative modifications.

The third Part of the thesis, that is, Chapters 7, 8, 9, 10 and 11, concerns the formalization of the informal framework previously introduced. In Chapter 7, the appropriateness of logic to formalize legal reasoning as it can be often argued in the literature is briefly introduced, and a gentle tour through well-known logic formalisms is intended to indicate some requirements for our formalization. Chapter 8 presents the particular logic, namely defeasible logic, on which is based our formalization. In Chapter 9, defeasible logic is extended with modalities to account for mental, agency and deontic concepts. Temporal aspects are integrated in Chapter 10. Finally, the Chapter 11 concludes the work presented in this thesis.

2

Computer science and law

L. Loevinger is often cited in literature for having initiated in an article published in 1949 [124] the use of logics and formalization techniques to obtain concrete applications in law. Since then, the possible interactions between computer science and law have grown to constitute a dedicated field usually simply called computer science and law. The different ways of understanding the domains of computers science and law, plus their possible and diverse relations are the basis of numerous systemisations that one can find in the literature (see e.g. [144]). As the domain of computer science evolves, new paradigms can be applied to computer science and law, and new systemisations may be provided.

We focus on two active fields of research which are relevant for our purpose, namely legal knowledge management, and, artificial intelligence & law (AI & law). The field of legal knowledge management concerns itself with the management of legal knowledge including the acquisition, modeling, dissemination and maintenance of knowledge, while the field of AI & law is more focused on tools to emulate legal reasoning.

We overview legal knowledge management and AI & law in the Sections 2.1 and 2.2 respectively. We shall not attempt to provide an exhaustive and precise account of the rich and diverse works in both fields. Our aim is rather that of a brief introduction of possible connections between the topic of this thesis and some works in computer science and law. In Section 2.3, we conclude the Chapter by giving some prospects.

2.1 Legal knowledge management systems

Legal knowledge management refers to the management of legal knowledge by information technologies. The motivations behind the management of legal knowledge by information technologies are multiple. A major motivation lies in the observation that organizations are faced with mounting difficulties in working with the amount of normative information (legislations, regulations, precedents etc.) accumulated over the years. Many organizations are now undermined by a normative overproduction coupled with a complexity on the intricacy of normative systems structures (e.g. the hierarchy) and normative content. However, we expect normative systems to be effective and efficient: one of the possible solutions to make them effective and efficient is the use of legal knowledge management systems.

In the following, some key stages in the management of legal knowledge by information technologies are briefly presented.

Acquiring

Knowledge acquisition sets the challenge of getting the information that is around, and turning it into knowledge by making it usable. Key issues include the identification of sources of legal knowledge, how to make tacit knowledge explicit, how to acquire and integrate knowledge from multiple sources (e.g. different legal sources).

The sources of knowledge can be diverse and a selection can be made with regard to the context, the objectives and the users. For example, if the objective is to build a legislative information website for citizens, then the sources of knowledge can be limited to legislative laws published in official journals. More complete systems may be required to consider further sources of information as precedents of courts, legal doctrines, perhaps even legal administrative praxis, etc. In civil law countries, a priority may be given to laws, whereas in common law countries, precedents may be given more importance.

Modeling

If the acquired legal knowledge is intended to be processed by computers, then it has to be encoded into some computable formats. To this regard, there are many different types of legal knowledge, different processes, different kinds of users, etc. and this heterogeneity requires an interoperable encoding format (see e.g. [112]). Interoperability is usually ensured by meeting certain standards. Over the years, the open standard Extensible Mark-up Language (XML) recommended by the World Wide Web Consortium (W3C) has imposed itself as the most appropriate standard for the encoding of information in legal documents.

XML is a markup language providing the possibility of combining a text and extra information about it. The extra information, including structure, layout, or other substantial content, is expressed using markup elements (or tags), which are typically intermingled with the primary text. XML is extensible in the sense that its users can define their own markup elements. Elements marking up legal documents are typically classified into:

- structural elements to indicate, for example, titles, sections, articles, commas, etc.,
- metadata elements to indicate, for example, the type of norm, the date of publication, the period in force, etc.,
- semantic elements to indicate, for example, the substantial content of clauses as obligation, exception, etc.

Usually, some constraints on the possible structures binding XML elements are specified using schemata which are expressed by some related XML schema languages, for example, Document Type Definition (DTD) or XML Schema. XML schemata must be able to act both as straightforward placeholders for the acquired knowledge coming in, and to represent the knowledge so that it can be used for problem-solving. This can raise some issues if the acquired knowledge is not well structured or intelligible. For legislative documents, to overcome such issues, legistic techniques (see e.g. [154]) can help to provide more well-drafted documents.

XML also has the advantage that it does not carry information about how to display material: thus, the display of material can be adapted to the needs of human users. The display can be specified using Cascading Style Sheets (CSS) or the eXtensible Stylesheet Language (XSL).

XML standards have been progressively adopted by institutional authorities and private organizations. For example, in Italy, in the context of a national project called Norma In Rete[1] (NIR) launched in 1999, some specifications for the mark-up of normative documents in XML (NIR DTD's) and for the identification of normative resources (NIR URN's) have been adopted. Similar projects exist elsewhere, e.g. Lexdania in Denmark, CHLexML[2] in Switzerland, FORMEX for European Union publications, etc.

Recently, the Semantic Web [31] has opened new prospects in modeling legal knowledge by providing tools to give well-defined meaning to information. Pieces of information expressed in natural languages are 'semantically' marked up with diverse standard XML-based languages. For example, the Resource Description Framework (RDF) aims at making simple statements about resources in the form of subject-predicate-object expressions while the Web Ontology Language (OWL) is a language for defining Web ontologies. The standard for encoding rules is still not fixed and is expected to facilitate the interchange of rules in rule-based systems. Recently, the European project Estrella[3] has developed a Legal Knowledge Interchange Format (LIKF) [37] building upon emerging XML-based standards of the Semantic Web, including RDF and OWL, in order to facilitate the interchange of legal knowledge among computer systems.

Storing

If the acquired and modeled legal knowledge is intended to be reused, then it has to be stored in some repositories. The obvious issue is how to store legal knowledge. Usually, legal knowledge is stored using available techniques on the market, i.e. databases. The most diffused types of databases are relational databases and object-oriented databases, but due to the increasingly common use of XML for modeling legal knowledge, the use of XML databases is now investigated in many projects.

Retrieving and disseminating

In any large repository, retrieval of knowledge is an issue. This is particularly true for the legal domain in which legal knowledge is usually distributed into many reposi-

[1] http://www.normainrete.it

[2] http://www.chlexml.ch

[3] http://www.estrellaproject.org

tories, each containing thousands of documents. Retrieval is usually performed using available information retrieval techniques on the market, i.e. by indexing legal knowledge with additional knowledge structures to limit and direct search for relevant content. Hypermedia technologies as the World Wide Web have opened new prospects, for example, by facilitating the navigation through the complex network of legislative documents. In the context of the Semantic Web, ontologies can be used to enable a richer and more conceptual way of information retrieval.

The dissemination of legal knowledge has to take into consideration the validity of norms. Usually, official journals are the reference to certify the validity of published laws. How can the validity of legal documents disseminated by information and telecommunication technologies be insured? For example, in Italy, the ministry of justice launched in 1999 the national project Norma In Rete engaging national administrations (parliament, ministries, government, cabinet, etc.) as well as local governments (regional government, regional parliament, city administrations, etc.) to build a single portal for accessing on-line[4] the entire corpus of national and local regulations, making these available to anyone with Internet access.

Besides, the dissemination faces the usual issues of knowledge management, such as the deliverance of relevant knowledge to the right user, at the right time, in the right form. For example, as soon as a new regulation appears in a particular domain, then it can be delivered to whom it may concern, and can be accompanied more valuably with relevant doctrines and precedents. Many various techniques exist and can usually be classified as push or pull: a user can ‘pull’ information toward themselves, while a publisher may ‘push’ information toward subscribers (publish/subscribe).

Maintaining knowledge

As legal knowledge evolves, repositories have to be maintained regularly. Legal knowledge is not erased but maintained over time. For example, for legislative regulation, versioning techniques allow to trace the history of documents.

Techniques to maintain a normative systems in an acceptable ‘order’ preceded the use of computers. Solutions among others are simplication, codification, deregulation etc. (see e.g. [154]). The most common technique in legal informatics is the so-called consolidation.

According to R. Pagano in [154] p.78, the term consolidation refers to the operation of substitution of a plurality of dispositions that have been stratified through time within a unique text. A consolidation does not bring substantive change but coherence and clearness. Though, in theory, ‘full’ consolidation involves more than just the operation of updating texts by applying normative modifications, in practice, (semi-)automatic consolidation is reduced to such operation due to the complexity of ‘full’ consolidation. Nevertheless, as normative modifications can be of different types producing complex scenarios, automatic application of normative modifications is a matter of research in its own right (see e.g. [30]).

[4] http://www.normainrete.it

2.2 Artificial intelligence and law

The field of artificial intelligence and law (AI&Law) is born as an application of the larger discipline of artificial intelligence. Artificial intelligence, at its very beginning, has favored the legal domain for its applications. The research was particularly inspired by logic fitting in an old philosophical tradition (see e.g. G. Leibniz) to ground legal reasoning in logical foundations. The idea is that legal rules and reasoning are integrable in a system which would help human beings make decisions. This vision is not without problems as it requires models of law that are computable, that is, reduce legal activities to algorithms, to formalize law. Can legal activities be reduced to algorithms? Can legal knowledge be formalized? A full presentation of the field of AI&law is beyond the scope of this work, and the reader is referred to the general introduction provided in [187] by G. Sartor. A good summary by E. Rissland et al. can be found in [178]. In the following, only a brief account of some common discussion on paradigms of AI&Law which are relevant for our purposes will be related. Our focus will then turn to legal knowledge-based systems.

2.2.1 Which AI paradigms for legal reasoning?

Building intelligent entities does not seem a straightforward task, and AI is a domain of confronted views on how it should be conducted. In the following, these oppositions are overviewed and their influence on some paradigms in AI&Law is briefly discussed.

Symbolic AI vs. connectivist AI. In symbolic AI, solutions are derived through theories representing the problem domain, while in connectivist AI, solutions result from an adaptation of complex system after an adequate training. In practice, the symbolic approach has led to knowledge-based systems, whereas connectionism has led to neural networks.

The symbolic approach has dominated the AI research from its beginning until today: intelligence has been understood as consisting in the manipulation of symbols. If the symbolic representations are logic formulas, and the syntactical manipulations of these formulas are inferences, then the symbolic approach corresponds usually to a logic approach. The common motivation for a logic approach is a better understanding of the reasoning problem itself. The representations and reasoning that the connectivist approach would produce might be too complex to characterize or to understand at a conceptual level. For this reason, the large majority of works in AI&Law is located in symbolic AI though some tentatives have also been performed in connectivist AI (see e.g. [161]). In the remainder, we will only focus in the symbolic approach.

Neat AI vs. dirty AI. Neat AI is based ideally with few elegant principles whereas dirty (or scruffy) AI uses many approaches combined. The neat vs. dirty views are usually related to the procedural vs. declarative accounts of knowledge. In the procedural view, the knowledge about a domain is intricate with the control of the reasoning process, whereas in the declarative view, the knowledge about a domain is clearly separated from the control of the reasoning process. In practice, for today's computer scientists, it amounts to a choice between procedural and declarative language programming. Examples of procedural languages are Pascal, C, C++, Java, Visual Basic etc. and examples of well-known declarative languages are Lisp and Prolog. In general, if pragmatism replaces philosophical opinions, then a procedural language is used when the designer can predict the knowledge to be processed whereas a declarative language is preferred when the evolution of the knowledge is unpredictable. Also, since programs in declarative languages are usually less computationally efficient than procedural languages in terms of memory and speed, the seconds are favored to achieve specific tasks.

In AI&Law, the neat view has been preferred because the legal knowledge to process is generally unpredictable. Another reason is that the few elegant principles of the neat view helps the community to understand and discuss more easily proposed systems.

Strong AI vs. weak AI. Strong AI makes an analogy between the mind and computers and refers to machines that are capable not only of producing smart behaviors, but also capable of having consciousness or a comprehension of its own reasoning. Weak AI is a pragmatic approach and seeks to build systems capable of solving problems of limited scope. The controversy has deep philosophical roots since it relates to whether human minds can be mechanized. As a matter of fact, strong AI is still a matter of science fiction, and proposals in AI&Law so far cannot be classified as 'minded' systems. Nevertheless, many proposals use notions as mental states, notably for example in the field of agent: such use of such mental states, however, do not necessarily need strong AI principles.

Orthogonally to these general considerations, assuming that intelligent artifacts aim at performing some (intelligent) tasks, AI distinguishes common sense tasks and expert tasks. Roughly, common tasks do not presuppose a high level of special knowledge or skill (i.e. without expertise) whereas expert tasks do. Practicing the law requires both type of 'intelligence' because legal reasoning is often about scenarios involving common sense reasoning. This is apparent, for example, by observing that many laws regulate common sense tasks. Hence, arguably, AI&Law requires intelligent systems able to deal with both common sense reasoning and legal reasoning. However, as a matter of fact, AI techniques are not enough advanced to deal with common sense reasoning. Hence, today's implementable AI&Law systems must be circumscribed to some limited expertise tasks.

Finally, the circumscription AI&Law systems to limited expertise tasks is further constraint in practice by the utility of these applications, and in [40], K. Branting, among others, argues that:

> Only when our conclusions are rooted in actual legal practice are we [the AI and Law community] likely to make a significant contribution to the legal community and the citizen-consumers of legal services.

2.2.2 Legal knowledge-based systems

The neat view of AI&Law has led to a preference for developing legal application based on the model of so-called *knowledge-based systems*.

Knowledge-based systems are substantially different from traditional systems in the way the knowledge is processed. In the latter, knowledge is implicitly encoded into some hard-wired procedures, whereas in knowledge-based systems, knowledge is represented explicitly under some formal structures describing the problem domain, for example as a set of statements expressed into some logical languages. This set of formal structures constitutes the so-called knowledge base. On top of this knowledge base, a module process the formal structures and provide an output, for example, by proposing legal advice.

A strong motivation for the use of knowledge-based systems in the legal domain lies in the maintenance of knowledge. Indeed, changes of the knowledge buried into traditional applications can require costly modifications of spaghetti procedures while, in knowledge-based systems, only a modular modification of the knowledge base is often sufficient, for example by adding or eliminating specific statements. As legal knowledge is likely to change over time, knowledge-base systems are less costly to maintain up-to-date.

Knowledge-based systems have to accommodate to the two main models of legal reasoning: a first model (here, called rule-based reasoning) assumes a set of legal norms (e.g. legislation), and a second model (called case-based reasoning) assumes a set of decisions related to precedent cases (e.g. legal sentences). The first model has resulted in knowledge-based systems called rule-based systems in which knowledge is encoded under rules, while the second model into case-based systems in which knowledge is represented by cases. Both systems are briefly overviewed in the following.

A rule-based systems is roughly composed of a knowledge base and an inference engine to process the knowledge. The knowledge base is assumed to be a set of facts and set of norms which are represented by rules. In its simplest, a rule has the form of an 'If-Then' conditional statement:

If antecedents Then consequent.

If the antecedents hold then the rule is fired and the consequent is derived. For example, a rule expressing that unmarried males are bachelors could be encoded as:

If x is a male and unmarried Then x is a bachelor.

The inference engine processes the rules stored in the knowledge base (the input) to arrive at some conclusions (the output). For instance, if the knowledge base contains the statement ‘Mario is a male and unmarried’ along with the rule given above, then the system is able to derive that ‘Mario is a bachelor’. As a rule base can be very large, it is necessary to have efficient inferencing mechanisms that search through the knowledge and deduce results in an organized manner. Two types of inferencing are usually distinguished:

- Backward chaining: the inference engine works backward from a conclusion to be derived to determine if there are rules in the knowledge base to derive the conclusion.
- Forward chaining: the inference engine works forward from the content of the knowledge base to derive a conclusion.

The knowledge representation and reasoning cannot be addressed here as it would require us to discuss logic formalisms: the reader is referred to Section 7.

A well-known illustration of rule-based systems is M. Sergot et al. in [199] who formalized the British Nationality Act using logic programming techniques.

The model of law as a set of decisions related to cases has represented the main alternative to the model as a set of legal rules. It is rooted in the doctrine of *stare decisis* according to which courts should stand by and adhere to precedent decisions. In case-based systems, the knowledge base is a set of cases, i.e. problematic situations solved by legal decisions (e.g. by legal sentences), instead of a set of rules. Legal case-based systems are part of the family of case-based systems. Case-based systems solve new cases by adapting precedent solutions which were used to solve precedent cases. Case-based systems have typically a cyclical process as proposed by I. Watson and F. Marir [212]:

- retrieve the most similar case(s),
- reuse the case(s) to attempt to solve the problem,
- adapt the proposed solution if necessary, and
- retain the new solution as a part of a new case.

As pointed out by I. Watson and F. Marir, this cycle rarely occurs without human intervention. For example, many case-bases systems are primarily used as case retrieval systems, and case adaptation is often being undertaken by users. However, instead of assessing it as a weakness, it is sometimes advocated to encourage human collaboration in decision support.

Often-cited examples of legal case-based systems are, among others, HYPO by K. Ashley [20] and CATO by V. Aleven [4].

As legal explanations supporting conclusions are as important as the conclusions themselves, knowledge-based systems are advantageous because they ease the presentation of such explanations.

In this regard, it is often argued that legal knowledge systems should implement so-called *deep models* of knowledge and reasoning rather than *shallow models* (see e.g. [25]). Roughly, deep models assume an isomorphism between the formalized knowledge and the knowledge to formalize, whereas shallow models tolerate a formalization into empirical associations with, for example, have no explicit reference to some causal relation underlying the domain. Isomorphism has also the purpose to ease maintenance of legal knowledge.

Deep models can concern the reasoning processes too. In this regard, computational argumentation techniques intented to model more naturally legal reasoning has attracted most of the attention since the early 90s (see e.g. [86, 211]).

Many criticisms and doubts can be advanced for the development of legal knowledge-based systems. These criticisms can be roughly classified into ethical and technical.

A common criticism concerns whether legal reasoning can take the form of algorithms. For example, concerning rule-based systems, since the formalization of legal knowledge into rules is usually made in some logical languages, and the manipulation of it with logical inferences, the appropriateness of logic to formalize legal knowledge and reasoning can be fairly discussed. The reader can jump to Chapter 7 for a discussion on logic and law.

Another criticism is based on the difficulty in identifying and isolating well-circumscribed pieces of legal knowledge. As an illustration, if we want to implement a complete expert system on intellectual property rights (IPR), then such system shall not only require to capture IPR laws, but also contract laws, and others: laws are not disjoint but interconnected.

A last issue, which is more economical than technical, lies in the observation that, notwithstanding some commercial successes (e.g. business rules companies), and though the availability of open source systems have greatly reduced the efforts and costs involved in developing knowledge-based applications, the difficulty of formalizing legal knowledge may lower the return on investment to unacceptable levels for potential investors.

2.3 Prospects for computer science and law

In the previous Section, we saw that computer science has approached law in two main perspectives: one consists in the implementation of legal knowledge management systems while the other is an investigation of legal knowledge-based systems using artificial intelligence techniques.

Legal knowledge management refers to the management of legal knowledge by information technologies. In practice, such systems are usually composed of diverse technologies to acquire, model, store, retrieve, disseminate and maintain normative knowledge.

Legal knowledge-based systems refer to information technologies in which the legal knowledge is explicitly represented. Most common examples are legal expert systems.

Both aim to provide applications to help citizens and jurists in their legal activities, but usually from a different perspective. It is our prediction that in the future, legal knowledge management systems and legal knowledge-based systems will merge to provide more complete legal solutions. In this view, legal knowledge management techniques shall arrange the legal knowledge base of legal knowledge-based systems, and conversely, as current legal knowledge management systems has little reasoning capabilities, AI&Law techniques are foreseeable to augment their intelligent functionalities. This view is confirmed, for example, by the multiple research interests in the use of standard 'semantic' technologies (e.g. see [37]) intented to model normative knowledge embedded in legal knowledge management systems so that it can be processed by AI&Law techniques.

However, many obstacles remain before the realization of efficient legal knowledge-based systems can be achieved. Another prospect for legal informatics is norm-governed systems, and particularly normative multi-agent systems. As argued in [58] by R. Conte at al., though the fields of legal informatics and normative multi-agent systems differ traditionally in models of references, formalisms and implementing technologies, they could profit from each other by combining the results of each field.

In the next Chapter, we shall overview the field of agent technologies, with a special focus on normative multi-agent systems.

3

Agent technologies

The agent paradigm provides a powerful conceptual encapsulation to account for many fields in computer sciences as cognitive models, communication, coordination etc. It is not our intention here to give an exhaustive review of the agent paradigm and associated technologies, so we shall present only some aspects relevant for our field of research.

This Chapter on agents technologies is organized as follows. In Section 3.1, the concept of individual agent is introduced while in Section 3.2, we deal with the concept of multi-agent systems with a special focus on normative multi-agent systems. Finally, in Section 3.3, the agent paradigm is compared with other conventional computer technologies.

3.1 The concept of agent

This Section introduces the concept of agent and presents typical architectures by which individual agents are implemented.

3.1.1 What is an agent?

The richness and the many different uses of the concept of agent has caused a situation where there is no commonly accepted definition of the term agent (cf. [78] for a variety of definitions). In [185], S. Russell and P. Norvig give the following definition:

> An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.

This definition is rather broad: an agent is not related to a computer system. Anything, e.g. any artifact, that senses an environment and acts upon it will be considered an agent. There is no constraint on the kind of environment, which can be physical or not, and the kinds of actions that are performed by the agent. So, according to S.

Russell and P. Norvig, a thermostat is an agent. In [216], M. Wooldridge proposes a somewhat more restrictive definition:

> An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.

and, P. Maes in [128] gives a rather closed definition:

> Autonomous agents are computational systems that inhabit some complex, dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.

Crucial elements are added to the definition of an agent. Firstly, an agent is related to a computer system, and secondly, this computer system can act autonomously.

The notion of autonomy is a delicate issue that has its root in philosophy. As one is autonomous as for a given action or goal (and not for another), and from something or somebody, it is current to understand autonomy in terms of (in)dependence. Accordingly, an agent is autonomous if it is entitled to have resources independence, reasoning independence, etc. In [53], C. Castelfranchi proposes the following insight of autonomy:

> Given the abstract control loop of any control system which purposively relates information, knowledge, goals, situation, and action, or given the general sensing-acting-environment loop of any interacting agent, if another agent interferes in, is inserted in the control loop, if the flow of causal effects on the world is interrupted and pass through another agent and needs it, the system is no more completely independent or autonomous; it depends on X's intervention.

C. Castelfranchi notes that if the variable X is an (human) agent, then this characterization corresponds to the well-known M. Wooldridge's view of autonomy as acting "without the intervention of human beings or other agents" [216]. If autonomy is understood in terms of independence then different types of autonomy are identifiable : information autonomy shall refer to information independence, planning autonomy shall to planning independence, goal autonomy to goal independence and so on. If we consider control independence, then we refer to the principle that an agent can make decisions and take on its own, that is, roughly speaking, control is inside the agent and not outside. This relates well with the Greek etymology of autonomous (*αυτøνøμøs*), literally, who gives oneself his/her own law. If an agent is autonomous, that does not mean that its behavior cannot be influenced. For example, a goal autonomous agent may (or may not) accept some requests and may adopt other goals under some conditions.

Beside the notion of autonomy, the definition of agent by M. Wooldridge and P. Maes given above introduces both the notion of design objectives, that is, the hypothesis that an agent is designed for specific purposes. On this point, according to C. CastelFranchi [53],

> [. . .] because a goal autonomous agent is an agent endowed with its own goals. An agent is fully socially autonomous if it has its own goals, endogenous, not derived from other agent's will. It adopts goals from outside, from other agents; it is liable to influencing. It adopts other agent goals only if it sees that adoption as a way of enabling itself to achieve some of its won goal.

This relates to delegation practice when the requesting agent has limited knowledge, competence, time and capability. If an agent is autonomous and, all the more, has some goals to realize, then it can take initiatives and not only react to change when it happens. For this reason, many people attribute pro-activity along with reactivity to an agent.

Because an agent can perform actions, we may prefer an agent that acts 'intelligently'. Tough there is no commonly accepted definition of intelligence, rationality is commonly accepted as a central property of intelligence. A rational agent is defined as an agent that always chooses the action which maximizes its utility, given all of the knowledge it currently possesses. Also, if agents can react autonomously to changes, then the ability to learn and improve with experience, is deeply linked to autonomy. While adaptivity is often considered a fundamental aspect of autonomy and intelligence, how to implement it is still an issue which we cannot address here.

Finally and orthogonally to those properties, an agent can enjoy mobility; that is, not all agents are mobile. For a software agent, mobility makes it free to travel among the hosts of a network (such as the Internet). It is not bound to the host in which it begins execution; it can transport its code and state to another host in the network, where it resumes execution. As D. Lange and M. Oshima states in [121], "this ability allows it to move to a system containing an object with which it wants to interact and then to take advantage of being in the same host or network as the object." Doing so, mobile agents can a solution to reduce the network load. Drawbacks of mobile agents concerns the hosts and the agents themselves. Hosts have to secure themselves from malicious agents (as a program running in on host) that shall violate the norms of the host. Meanwhile, mobile agents shall also protect themselves from mischievous hosts.

3.1.2 Agent architectures

Depending on the required behavior, agents can be implemented according to different styles of architectures. The styles of architecture most widely referred to in the literature are reactive agents, model-based agents, deliberative agents, and layered architectures.

Reactive agents, also called reflex-based agents, operate in a simple stimulus/response fashion. This architecture does not support any explicit reasoning. Its main advantage is that agents can react in time in fast changing environment. An often cited work of reactive architecture is [47]. The main disadvantage is that reactive-based agents work only if the correct decision can be made on the basis of current perceptions exclusively. So, reactive-based agents may not be practicable if the decision-making is influenced by the history of perceptions or if the environment is not fully observable (i.e., a partially observable environment), that is, if more sophisticated behaviors are required.

Model-based agents remove the limitations of reactive agents by managing environments that are partially observable and by using the history of perceptions. To do so, such agents have a model that represents relevant aspects of the environment. An agent maintains internal states to keep track of (i) parts of the environment that the agent cannot perceive, and (ii) the history of its perceptions. Typically, the internal states correspond to symbolic representations of the environment, and the agent's behaviors are defined by syntactically manipulating this symbolic representation. If these symbolic representations are logical formulas, and the syntactical manipulation corresponds to inferences between these formulas, then we are dealing with the class of logic-based agents. So a logical theory is implemented, and the process of selecting an action reduces to a problem of proof: if the agent derives a logic formula corresponding to a state of the world, then the agent performs the associated actions. Examples of works on logic-based agents are MetateM [76] and Congolog [122]. However, the inherent computational complexity of theorem-proving makes it questionable whether agents as theorem provers can operate effectively in time-constrained environments. So the theorem-prover architecture is often not acceptable in environments changing faster than the decision-making agent, since goals may become obsolete.

Deliberative agents remove the limitations of previous agents by reasoning about their internal states: for example, a deliberative agent can figure out that a plan or a goal has become obsolete and thus can reconsider it. Deliberative agents are mostly represented by the class of belief-desire-intention (BDI) agents, which are increasingly prevalent. These architectures have their roots in the philosophical tradition of practical reasoning, i.e., the process of figuring out what actions to perform (see e.g. [42, 167]). Practical reasoning consists of two distinct activities. The first involves deciding *what* states of affairs we want to achieve based on the current beliefs: this activity is called the deliberation. The second activity involves *how* we want to achieve these states of affairs: this activity relates to planning and scheduling. A compelling reason to use BDI architectures is that since an agent can be described in terms of mental notions that are more usually applied to humans, it becomes easy for observers to understand and predict its behavior [62]. A well-known example of BDI architectures are the Procedural Reasoning System (PRS) [85] which has inspired the development of successive similar systems (see e.g. [67]).

Recently, with the acknowledgment that norms can be an appropriate mean to control autonomous agents, some extensions of BDI architectures accounting for normative agents have been proposed. Examples of works in this direction are [63] proposing a BDI interpreter loop that takes norms and obligations into account in an agent's deliberation, and the BOID architecture (see e.g. [44, 61]) as an abstract agent representation, that consists of the four components beliefs, obligations, intentions and desires.

Layered architectures, also called hybrid architectures, attempt to give agents both reactive and deliberative features. Indeed, in general, agents can be neither totally deliberative nor purely reactive. If they are just reactive they cannot reason. If they are only deliberative they may never be able to act efficiently in time. This leads typically to a combination of both reaction and deliberation architectures, in an effort to take the best from each approach. Generally, reaction and deliberation processes are associated with different layers that interact so that agents can follow their own plans while sometimes reacting to external events without deliberation. Most cited works on layered architectures are InteRRap [141] and TouringMachines [73].

3.2 Multi-agent systems

Computer systems have evolved from centralized to distributed models in order to accompany the decentralization and cooperation of present-day organizations. Applications are no longer monolithic programs functioning on one machine for a single user, or distributed applications managed by a single organization such as today's intranet, but rather societies of components that may not all have been designed by the same software development team and may be owned and managed by different organizations. E-government systems spanning multiple administrations is an example. Due to this inherent distribution of information and competencies, components may be unable to accomplish their own tasks alone and are consequently required to interact by providing services to one another. This is reflected, for example, in the emerging visions of Semantic Web [31], Grid [77], and Ubiquous Computing [213], among others.

Systems composed of multiple (interacting) agents are referred to as multi-agent systems (MAS). Applications of MAS already exist. They are appreciated for example in situations where agents interact with one another to solve problems that are beyond the individual competencies, and in which several agents must together make some joint decisions in order to accomplish some assigned tasks. Such applications refers typically to many agents that are spread widely over a geographically distributed environment (see e.g. [158]). Other applications are multi-agent simulation systems (MASS) are used to simulate complex systems as economies, societies and biological environments. Indeed, MASS allows to provide answers to complex problems that would otherwise be unobtainable via classic mathematical tools. For example, J. Epstein and R. Axfell in [69] investigate how the heterogeneous behavior of individual actors or agents generate the global macroscopic regularities of social phenomena. Doing so, MASS can permit to develop plausible explanations of observed phenomena and help in decision making. Also, using MASS, algorithms can be tested in a isolated environment, before going to field trials with actual users. For example, A. Artikis at al. in [19] simulate a formal framework for specifying, animating, reasoning about and verifying the properties of open computational societies.

Whereas the previous Section presented the characteristics of individual agents, in this Section we focus on multi-agent systems.

3.2.1 Communication

Interaction between agents is generally done by communication; that is, messages are transmitted from senders to receivers via a communication medium provided by an infrastructure. Messages are written in an agent communication language (ACL).

Most of the work done in ACL is based on the theory of speech acts (see e.g. [196, 197]), which conceives human natural language in terms of actions. Roughly speaking, speech acts theory states that a language is used not only for making a statement but also for performing actions. When an agent transmits information to another agent, this has an effect just as any other action would have. Thus, one can define preconditions and effects for communicative acts in terms of mental states as beliefs, goals, intentions, desires, etc. Each communicative act can be conceived as an attempt by the sender to influence the mental state of the receiver of a message. Defining ACLs in terms of mental states may ease its integration with BDI agent's architecture (see 3.1.2). Speech act theory is relevant to agent communication in that it serves as a formal basis for deciding on primitive concepts in ACL. Furthermore, a significant contribution of this work is that the semantic of the composite communicative act derives from the act's constituents. Popular ACLs of the mental agency approach are KQML [74], and FIPA ACL [75].

Typically, a message includes an indication of the type of communicative act (for example, an act by which to inform, query, propose, etc.), the name of the sender and receiver agents, the content of the message, and the ontology to be used in interpreting that content. The content of a message is written in a Content Interchange Language (CIL). CILs typically use the logic of predicates to make statements concerning, for example, simple concrete facts, definitions, abstractions, inference rules, constraints, and even meta-knowledge (knowledge about knowledge). A popular CIL is the Knowledge Interchange Format (KIF) [84].

Ontologies allow agents to agree about the meaning of the terms from which the contents of messages are made. Typically, ontologies are concretized under files that contain sets of terms (a vocabulary) organized under taxonomic hierarchies: these hierarchies contain different types of relations between terms, and these relations are formalized. Usually, such files are written using specific languages, such as the Web Ontology Language (OWL), which is a specification of the World Wide Web Consortium (W3C) [210].

It has been argued that the ACLs of the mental agency approach are unrealistic because they suppose that agents can access each other's mental states. For this reason, M. Singh in [202], for example, proposed an ACLs semantics base in social agency instead of mental agency. This social agency recognizes that communication is inherently public, emphasizes conventional meaning, avoids pragmatics and considers context. The social agency approach is more focused on the notion of protocols and for this reason we will return to it in the next Section.

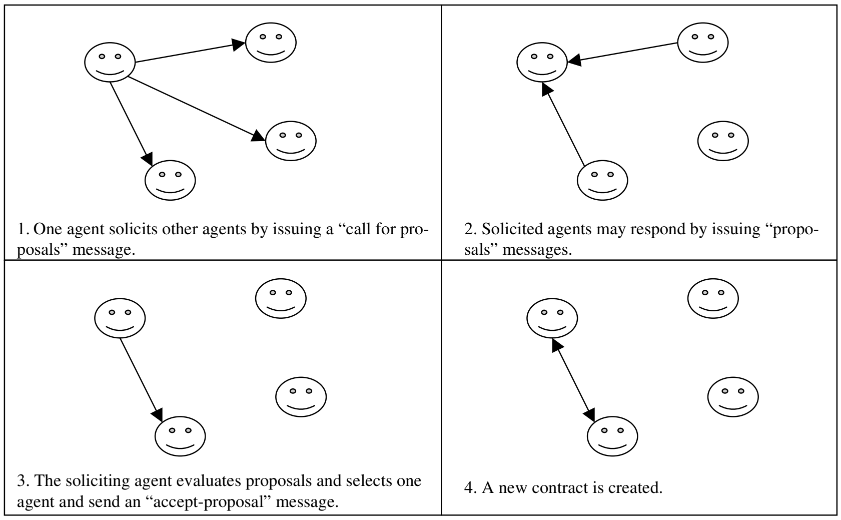
3.2.2 Coordination

Agents are often required to deal with coordination to govern interaction, for example to manage dependencies between activities performed to achieve a goal, or to avoid conflicts. Coordination can take many forms, for example, negotiations among competitive or self-interested agents, cooperations among non-antagonistic agents, etc. Aspects of coordination include the synchronization of agents, the avoidance of live-locks and dead-locks, optimization of resources etc.

In a MAS, the coordination must integrate the specificities of agents. Components in typical concurrent systems are built by the same organization to achieve a common goal so that coordination is typically hardwired in at design time. Instead, in MAS, each agent can be designed by different organizations and may have different (conflicting) goals so that coordination among agents is assumed to happen at run-time.

Typically, agents coordinate by communicating. Ideally, an ACL of the mental agency approach, such as KQML or FIPA ACL, allows agents to coordinate by reasoning on mental attitudes as beliefs, goals, etc. of other agents. As a matter fact, this approach of coordination based on mental attitudes is hardly explored because the associated issues. These issues include the attribution of mental states to programs in general (see e.g. [215]), the access to others private mental states, or some assumptions such as sincerity and cooperation (see e.g. [202]).

An alternative is the use of communication *protocols* (or conversation specifications) which define sets of rules to guide the interactions taking place between several agents: these rules, in other words, govern the exchange of a series of messages among agents. Several generic communication protocols have been devised for systems of agents. For example, a popular protocol is the Contract Net Protocol [203] which aims to control the bidding and contracting mechanisms of agents, in a marketplace setting for instance. The Contract Net Protocol is shown in Figure 3.2.2 and works as follows: one agent solicits other agents by issuing a ‘call for proposals’ message that specifies the task and any conditions placed on the execution of the solicited task. Solicited agents may respond by issuing some messages that indicate their proposals. Proposals may include the preconditions for the task, which may be the price, the time when the task will have to be done, etc. Once a deadline passes, the soliciting agent evaluates any received proposals and selects one or more agents to perform the task. The agents of the selected proposal(s) will be sent an ‘accept-proposal’ message, and the others will receive a ‘reject-proposal’ message. Ideally, the proposals are binding, so that once the soliciting agent accepts the proposal, the selected agents will commit themselves to performing the task and, for example, a contract is formed.



1. One agent solicits other agents by issuing a "call for proposals" message.

2. Solicited agents may respond by issuing "proposals" messages.

3. The soliciting agent evaluates proposals and selects one agent and send an "accept-proposal" message.

4. A new contract is created.

Fig. 3.1. A graphical representation of the contract Net Protocol.

Typically, messages received are interpreted according to the protocol they are inserted in. For example, a message proposal received in the context of the Contract Net Protocol will be interpreted differently in the context of the Dutch Auction Protocol, which is another sophisticated protocol to rule auction mechanisms between agents.

On the pursuit of coordination by ACLs, M. Singh in [202] proposes to characterize protocols in terms of commitments:

> "[. . .] agents play different roles within a society. The roles define the associated social commitments or obligations to other roles. When agents join a group, they join in one or more roles, thereby acquiring the commitments that go with those roles. The commitments of a role are restrictions on how agents playing that role must act and, in particular, communicate."

Accordingly, protocols are defined in terms of commitments rather than as finite state machines. M. Singh argues that if protocol requirements would be expressed solely in terms of commitments, then agents could be tested for compliance on the basis of their communications.

In the same line, conversation policies which define constraints over different aspects of conversations have been proposed along with conversation specifications in order to provide more flexible control over agent communication, and thus coordination (see e.g. [162, 109]). In this view, modifications of declarative representations of conversation policies can ease the adaptation of the communication modules of agents to circumstances.

The protocols discussed above can be classified as communication protocols which consist in rules to guide the communication internal to some specific tasks. However, coordination can be understood as the governance of interaction among these specific tasks. At this level, many techniques exist and, unfortunately, no clear classification emerged in the literature (see [153]).

Whatever the coordinating technique, the autonomy of agent has to be respected. In this regard, argumentation techniques have been advocated (see e.g. [146]) to structure to structure a wide range of coordination aspects occurring in persuasion, negotiation, information-seeking, or inquiry settings etc. For example, as coordination aims at managing dependencies between activities performed to achieve a goal, an agent may request another to act in a certain way. The request may more successful if the agent builds and communicates arguments to persuade the other to act in the requested way. In this view, works on argumentative dialogue games can be used to specify protocols to govern the exchange of arguments.

For our purposes, it is interesting to note that the use of norms has been acknowledged as an appropriate mean for coordination. For example, Y. Shoham and M. Tennenholtz in [201] suggested to use social laws to coordinate agents: the agents agree on certain rules in order to decrease conflicts among them and promote cooperative behavior. In [57], R. Conte and C. Castelfranchi proposes also norms to achieve coordination between them, but conisder a more cognitive ground for norms studies. A. Omicini and S. Ossowski inspired by M. Shumacher [194] make in [152] a distinction between subjective and objective coordination, depending on whether we adopt a psychological or normative account of coordination.

3.2.3 Normative multi-agent systems

A system composed of multiple (interacting) agents is called a multi-agent system. The research area defined as the interaction of normative systems and multi-agents systems aims at investigating *normative multi-agent systems*. In [52], A. Jones and J. Carmo provide the following definition of normative systems:

> Sets of agents whose interactions are norm-governed; the norms prescribe how the agents ideally should and should not behave. [...] Importantly, the norms allow for the possibility that actual behavior may at times deviate from the ideal, i.e., that violations of obligations, or of agents' rights, may occur.

Another definition is provided by M. Luck and F. López y. López in [125]:

> A normative agent is an autonomous agent whose behavior is shaped by the norms it must comply with. A normative agent must be able to decide, based on its own goals and motivations, whether a norm must be either adopted or complied with aware of the consequences of dismissing norms. [...] A

> normative multi-agent system is a collection of normative agents which are controlled by a set of common norms varying from obligations and social commitments, to social codes.

In [35], G. Boella and L. van der Torre proposes to extend these definitions with agents' control over the set of norms:

> A normative multi-agent system is a multi-agent system together with normative systems in which agents on the one hand can decide whether to follow the explicitly represented norms, and on the other the normative systems specify how and in which extent the agents can modify the norms.

As an organization of agents bound with norms can be defined as an institution, some authors have proposed to model (open) agent organization as electronic institutions (see e.g. [71]). However, the distinction between electronic institutions and normative multi-agent systems is not clear since institutions are constituted by norms.

A normative system can be composed by diverse types of norms, and many classifications, often rooted in social sciences, exist.

In philosophy of law for example, H. Hart makes in [102] a distinction between primary and secondary legal rules: primary rules govern conduct of agents while secondary rule manage primary rules. Secondary rules are further classified into rules of recognition (to determine the 'validity' of legal rules), rules of change (to change legal rules) and rules of adjudication (to empower some agents with adjudication power, for example to adjudicate whether a law has been violated.)

In computer sciences, G. Boella and L. van der Torre (see e.g. [36]) consider regulative norms to specify the ideal behavior of agents, constitutive norms used to constitute institutions, procedural norms address to the mechanisms to support primary substantive norms.

We cannot address here the multiple existing classifications, but we hope that the two examples given above illustrate well that such classifications allow systematic indications for the use of norms in multi-agent systems.

For example, an observer of a multi-agent system can be invited to reason on an organizational rather than an individual level in order to discuss the overall behavior of the society. Thus, a multi-agent system can be understood as having some sort of collective intention rather than a sum of individual intentions. Accordingly, norms can be collective, i.e. they are directed to a group of agents so that the group ought to behave accordingly. What is particularly interesting with collective norms is to understand their impact on the individual behaviors of the agents in the group, i.e. to understand when and how the collective directives are translated into individual directives. If a collective directive is violated, then the application of the sanction is at issue. Legal reasoning associated with collective directives can take account for many parameters. For example, L. Cholvy and C. Garion proposes in [81] a model in which if a group has neither a particular hierarchical structure nor an institutionalized representative agent, then the derivation of individual directives depends on the ability of the agents (i.e. what they can do) and their own personal commitments (i.e.

what they are determined to do). As for checking if these obligations are fulfilled or not, we need to know what are the actual actions performed by the agents.

An important issue in normative multi-agent systems is the sources of norms. In the field of multi-agent systems, there are two approaches: norms can be designed off-line (by some administrators for example), or they can emerge from the agents themselves.

The former is the most prominent approach while the latter is usually reducted to scenarios of self-contracting agents. In this regard, the notion of so-called *institutional power* is a possibility to account for the characteristic feature of an institution whereby some agents are empowered to create or modify significant normative aspects of that institutions, usually by performing a special act, for example by signing a contract. The notion of institutional power is sometimes captured by considering that an action or a state of affair A *counts as* an action or state of affair B in that particular institution (see e.g. [107, 82]).

Moreover, as norm emergence can be related to norm propagation, and though a substantive amount of work on the propagation of norms exists, few formal frameworks have been developed in this regard. In [193] for example, some mechanisms for the propagation of norms from a leader of a society to the followers are proposed.

Though in common multi-agent systems, there is a single normative system, there can be several of them. This is particularly important since artificial societies are often characterized according their properties of openness. Openness of a society relates to the degree on which agents can join the society. So, artificial societies are traditionally divided up in two types: open societies, where it is possible for an external agent to join the society, and closed societies, where an external agent cannot join it.

In open multi-agent systems, an agent belonging to an institution with a normative system X may join another institution with a normative system Y. The relationship between the institutions can be hierarchical for example between a global and local authority. This raises the question of conflicts among norms of different institutions and how agents deal with them. In this regard, F. López y López et al. [126] argues that the normative behavior of agents not only relates to the decision of whether to comply with a norm or not, but also to the decisions of whether to join, to stay or to leave a society regulated by norms.

A closed society refers typically to a multi-agent system in which agents are explicitly designed to achieve common goals. As a closed society is by definition not opened, and usually designed and managed by an unique organization, the issues concerning conflicts among norms of different institutions hardly exist unless the closed society itself is composed of diverse societies.

The systematization of all the different notions to design and implement normative multi-agent systems is an issue in its own right. On this aspect, norms can be useful to design a system by specifying declaratively aspects of such systems.

In this view, V. Dignum et al., for example, proposed in [64] a framework for modeling agent organizations by three dimensions: normative, organizational and ontological that describe different characterizations of the environment. Orthogonally, in order to make design of the multi-agent system manageable, the methodology distinguish three levels of abstraction with increasing implementation detail. At the most abstract level, the statutes of the organization to be modeled are defined (statues indicate the main objective, values and context of the organization) and the ontology of the model itself. At the concrete level, starting from the abstract values defined in the previous level, the analysis and design of the system is refined in terms of norms and rules, roles, landmarks and concrete ontological concepts. Finally, at the implementation level, the design specified in the concrete level is implemented in a given multi-agent architecture.

The implementation of multi-agent systems can be facilitated using middlewares, that is, softwares which make the link between the hardware, the operating system, the network at the bottom, and the agent-software application on top. Middlewares hide the complexity of networking for the application developer and offer an infrastructure for general distributed applications. The aim of middleware is to facilitate the implementation, deployment, and management of distributed agent applications by providing different basic services, i.e. promoting the reuse of code. A example of a popular open source middleware is JADE [24]. A comprehensible review of multi-agent system middlewares can be found in [127]. Some investigations with regards to the implementation of normative multi-agent systems exist (see e.g. [113, 205, 3, 72, 70]). Such infrastructures are concerned with norm-oriented services as, for examples, the detection of when a norm is active, the detection of a violation on a norm, and the handling of violations with sanctions.

3.3 Agents and other computer paradigms

A comparison of software agents with other conventional information technologies makes it possible to highlight similarities and differences, and perhaps to better discern the concept of agent. I discuss below the relationship that agents have with the client/server model, with objects, and with expert systems.

3.3.1 Relationship between agents and the client/server model

The most current model for distributed applications is the client/server model. The client/server model makes a distinction of roles between its distributed elements, namely, servers and clients. Servers are reactive, in that they cannot communicate with a client on their own initiative; they handle most of the capabilities and must provide stability (they cannot appear and disappear, for example). Clients, on the other hand, do take initiatives, in the sense that they can initiate a communication with a server but cannot communicate directly with one another; they have few capabilities, and they are allowed to appear and disappear. Most today's Internet applications are based on the client/server model.

On this basis, one would be tempted to assimilate client and server as two special kinds of agents and consequently consider the client/server model as a special case of the agent-based model. However, the two models are traditionally keep distinct because, as noted by [24], agent-based models do not necessary make a distinction of roles between client and server. That is, all the applications based on the agent-based model cannot fit well with the client/server model. In this view, the agent paradigm is appreciated to facilitate the design of applications that manage the complexity of a domain. Indeed, as the complexity of a system arises from the interactions between elements of the system, agents ease their representation and the implementation of their interactions. Furthermore, scalability and flexibility of applications is facilitated by the integration of agents using step-by-step procedures whereas traditional servers tend to limit them.

In the increasingly popular peer-to-peer model (see [139] for a review), peers are nodes of a network that can function both as servers and clients at any time, depending on the role the peer acts in. A peer may thus be likened to an agent, and for that reason, some people have argued that the multi-agent paradigm can be superimposed on the peer-to-peer architecture.

3.3.2 Relationship between agents and objects

Objects are commonly defined as computational entities capable of receiving messages, processing data, and sending messages to other objects. A computer program, then, may be conceived of as composed of a collection of objects that act on one another, as opposed to a traditional view of a program as a collection of functions or procedures, or simply as a list of instructions to the computer. Objects usually possess so-called “methods” that consists of a sequence of instructions to perform an action, a set of input parameters to parametrize those actions, and possibly an output value of some kind. One can invoke a method for an object that must then perform the sequence of instructions contained in the method. A method invocation is thus considered to be a request for an object to perform some task, and it is often conceive as a means of passing a message to an object.

Some distinctions between agents and expert systems can be indicated (see e.g. [151]). Firstly, agents are usually argued to embody a stronger notion of autonomy than objects; in particular, agents decide for themselves whether or not to perform an action on request. When invoking a object’s method to make it to perform an action, the object has no choice whether to execute its method or not: once an object’s method is invoked, its execution will follow necessarily. In an agent-oriented setting, there is no concept of ’invoking a method’ on a agent. It cannot be taken for granted that an agent will perform an action only because one requests the agent to do so. With objects, control comes from the outside, whereas in an agent setting, it comes from the inside: this why agents are argued to embody a stronger notion of autonomy than objects. Secondly, agents are capable of flexible (reactive, proactive, social) behavior, while the standard object model is not primarily concerned with such types of behavior. Thirdly, a multi-agent system is by nature multi-threaded, that is, each agent is assumed to have at least one thread of control, whereas the standard object model usually runs with a single thread of control in the system.

Furthermore, because object-oriented applications have specific methodologies but agents are not objects, it has been argued that agent-oriented applications requires also specific methodologies (see e.g. [217]). For example, object models hardly capture notions as an agent proactively generating actions or dynamically reacting to changes in their environment, still less how to effectively cooperate and negotiate with other self-interested agents. Many agent-oriented methodologies have been proposed over the years, and their presentations goes beyond the scope of this thesis. The reader can find a survey in [105].

Finally, note that agents are commonly implemented using object technologies, that is, the two paradigms are not competitive but complementary.

3.3.3 Relationship between agents and expert systems

An expert system is a system capable of solving problems or giving advice in a specific knowledge domain and has been an important artificial intelligence technology. The problem-solving activity is performed by an inferential engine, that is, a program capable of drawing inferences from pieces of knowledge contained in a knowledge base. The knowledge is represented explicitly, typically as a set of logic statements.

While there are obvious similarities between agents and expert systems, there are also significant differences. These differences are evoked in the following by [216]. First, expert systems do not usually interact directly with any environment: they get the information not through sensors but through a user acting as an intermediary, while the interaction between agents and their environment takes place directly and autonomously. Secondly, expert systems are not normally capable of proactive behavior. Thirdly, while some applications make use of several expert systems that communicate with one another, making these systems look in some sense like multi-agent systems, expert systems are not usually equipped with any evolved communicating ability.

In regard to the legal domain, laws may apply to agents whereas legal expert systems are intended to provide legal advices.

In light of these distinctions, any expert system may be likened to a sort of very ‘poor’ agent, whereas not all agent can be likened to an expert system.

Part II

Informal

4

Reasoning framework

Our intention is to give a formal account for some aspects of the interaction between normative systems and cognitive agents. The term cognition is used to refer to a faculty for the human-like processing of information and we are interested here in the reasoning processes involved in cognition. In this Chapter, we first present informally a general abstract reasoning framework to express our model of cognition. Then, in Chapters 5 and 6, this abstract framework of reasoning will be specified to account for some aspects of reasoning involving mental concepts and temporal aspects.

In Section 4.1 of this Chapter, we root our framework on defeasible reasoning. Then, we present the notion of argument (Section 4.2), defeat between arguments (Section 4.3), and finally justified arguments (Section 4.4).

4.1 Defeasible reasoning

A reasoning is *defeasible* when certain conclusions are abandoned in light of further information whereas a reasoning is *deductive* when validity of conclusions are guaranteed even in light of further information. Though defeasible reasoning belongs to the domain of common sense, it has been studied intensively in philosophy and computer science, especially in artificial intelligence.

In antiquity, Aristotle, in his work entitled the Topics, emphasized deductive reasoning, especially in the form of the syllogism: If we have the premises that all men are mortal and that Socrates is a man, then we can conclude that Socrates is mortal. Having these premises, nothing can be added to our knowledge will change that conclusion. However, Aristotle also pointed out the important place of *dialectic* reasoning using rational inference based on probable premises. For Plato, dialectic becomes the process to provide the good.

After antiquity, and in the western world, deductive reasoning seemed to be the major object of concerns, and it was only in the mid twentieth century that a new special attention of defeasible reasoning was given. In this regard, R. Chisholm is often cited for his thesis that sensory appearances provide righteous but defeasible reasons for believing in some facts about the physical world [54, 55]. If something looks red, then we can presume that it is red. But if the light is red then this presumption can be defeated.

Another often cited figure is the philopher J. Pollock who proposed a formal theory to account for defeasible reasoning in terms of defeasible arguments which can be defeated by other (defeasible) arguments [163, 164, 165, 166, 167]. In the view of J. Pollock, the motivation for defeasible reasoning is strongly related to the limitations of agent capacities. Indeed, a bounded agent must be able to adopt beliefs on the basis of incomplete information whereas an omniscient agent would not require necessarily defeasible reasoning because no further information can be provided to its knowledge. However, even if the agent is omniscient, it may not fast enough to process all the information: an agent with defeasible reasoning has the advantage that it can provide quickly provisional conclusions while continuing to inquiry in order to adjust these conclusions. This may be also a disadvantage because a provisional conclusions may be merely incorrect or uncertain, this issues when an agent should stop inquiring by balancing the costs and benefits of the inquiry.

In the legal domain, many doctrinal studies observed the notion of defeasible reasoning. In this regard, some quotations from famous philosophers can be found in [191] by G. Sartor:

> All law is universal, and there are some things about which it is not possible to pronounce rightly in general terms; therefore in cases where it is necessary to make a general pronouncement, but impossible to do so rightly, the law takes account of the majority of cases, though not unaware that in this way errors are made. And the law is none the less right; because the error lies not in the law nor in the legislator, but in the nature of the case, for the raw material of human behavior is essentially of this kind. So, when the law states a general rule, and a case arises under this that is exceptional, then it is right, where the legislator, owing to the generality of his language, has erred in not covering that case, to correct the omission by a ruling such as the legislator himself would have given if he had been present there, and as he would have enacted if he had been aware of the circumstances. (Aristotle, Nicomachean Ehics)

> [It] is right and true for all to act according to reason: And from this principle it follows as a proper conclusion, that goods entrusted to another should be restored to their owner. Now this is true for the majority of cases: But it may happen in a particular case that it would be injurious, and therefore unreasonable, to restore goods held in trust; for instance, if they are claimed for the purpose of fighting against ones country. And this principle will be found to fail the more, according as we descend further into detail, e.g., if one were to say that goods held in trust should be restored with such and such a guarantee, or in such and such a way; because the greater the number of conditions added, the greater the number of ways in which the

> principle may fail, so that it be not right to restore or not to restore. (T. Aquinas, Summa Theologiae)

In the first half of the twentieth century, defeasibility has been studied by D. Ross who developed a theory of prima-facie obligations (see e.g. [182, 183]) in which the balance of conflicting prima-facie obligations allows us to establish our real obligations. In [183], D. Ross writes:

> Moral intuitions are not principles by the immediate application of which our duty in particular circumstances can be deduced. They state [...] prima facie obligations. [...] [We] are not obliged to do that which is only prima facie obligatory. We are only bound to do that act whose prima facie obligatoriness in those respects in which it is prima facie obligatory most outweighs its prima facie disobligatoriness in those aspects in which it is prima facie disobligatory.

Later, H. Hart [102] reintroduced the notion of defeasibility:

> When the student has learnt that in English law there are positive conditions required for the existence of a valid contract, [...] he has still to learn what can defeat a claim that there is a valid contract, even though all these conditions are satisfied. The student has still to learn what can follow on the word unless, which should accompany the statement of these conditions. This characteristic of legal concepts is one for which no word exists in ordinary English. [. . .] The law has a word which with some hesitation I borrow and extend: This is the word 'defeasible,' used of a legal interest in property which is subject to termination of 'defeat' in a number of different contingencies but remains intact if no such contingencies mature.

A. Peczenik investigates the idea of non-deductive 'jumps' in legal reasoning and the opposition of prima-facie legal obligation and all-considered legal obligations [159]. Those considerations involves a notion defeasibility. In [159], the need for defeasibility is motivated by A. Peczenik in a J. Pollock's flavour by arguing that:

> No human being has the resources sufficient to formulate all-things-considered statement *sensu stricto*.

More recently, H. Prakken and G. Sartor in [175] argue that defeasibility in the law involves three different aspects, which they call inference-based defeasibility, process-based defeasibility, and theory-based defeasibility. Regarding inference-based defeasibility and process-based defeasibility, H. Prakken and G. Sartor write:

> Inference-based defeasibility covers the fact that legal conclusions, though correctly supported by certain pieces of information, cannot be inferred when the theory including those information is expanded with further pieces of information (we use the term "theory" to mean in general any set of premises intended to provide an account of a legal domain). This idea is also frequently expressed by saying that common sense reasoning (as opposed to logical deduction) is nonmonotonic: the conclusions of the reasoning process do not grow inevitably as further input information is provided.

In legal texts, inference-based defeasibility is allowed by many diverse expressions such as “unless proved otherwise”, “unless otherwise agreed”, “except in those cases …”, “subject to …”, etc. The distinction between inference-based defeasibility and process-based defeasibility is explained in the following:

> Theories of inference-based defeasibility (i.e., non-monotonic logics) essentially take a static view on reasoning, since they just look at a given body of information, and say which beliefs are warranted on the basis of that information. Even the definition of non-monotonicity is stated in essentially static terms: it just says that the beliefs warranted on the basis of an information set S need not be included in the beliefs warranted on the basis of a larger information set S0. This definition abstracts from whether S precedes S0 in the reasoning process or not; and if S did precede S0, it is is silent on how the reasoning progressed from S to S0. Most importantly, ‘standard’ non-monotonic logics say nothing about in which circumstances S can be taken as a basis for decision, and in which circumstances it is better to search for more information. Process-based defeasibility addresses such ‘dynamic’ aspects of defeasible reasoning. As for legal reasoning, a crucial observation here is that it often proceeds according to the rules of legal procedures.

Process-based defeasibility is well illustrated in dialectic debates of judicial proceedings where each party tries to defeat the arguments of the other parties. Theory-based defeasibility is explained in the following:

> The third form of defeasibility [...] is theory-based defeasibility. This results from the evaluation and the choice of theories which explain and systematize the available input information: when a better theory becomes available, inferior theories are to be abandoned.

On this aspect, L. McCarty in [137] observes:

> A judge rendering a decision in a case is constructing a theory of that case. It follows, then, that a lawyer’s job (in an appellate argument) is to construct a theory of the case, too, and one that just happens to coincide with his client’s interest. Since the opposing lawyer is also constructing a theory of the case, which coincides with her client’s interest, the argument boils down to this: which theory should the judge accept? There are several constraints here, but they are weak ones. Legal theories should be consistent with past cases, and they should give acceptable results on future cases. They should not be too hard to understand, or too hard to administer, which means that they should be free of major anomalies. One term that is sometimes used to describe these requirements is “coherence”.

T. Bench-Capon and G. Sartor in [190, 26, 27, 28] provide a model of legal reasoning accounting for theory construction and theory comparison: alternative legal theories are built by parties in case, and the party that have built the ‘best’ theory wins. Comparison among theories are based on combination diverse aspects of coherence such as:

- factor-coverage, a better theory takes into account more features of precedents,
- value-coverage, a better theory takes into account a larger set of values,
- analogical connectivity, a better theory includes more analogical connections between its components,
- non-arbitrariness, a better theory contains fewer ad-hoc statements, required neither for explaining the past evidence nor for implementing the shared assessment of value-priorities.

Finally, defeasibility that leads naturally to the comparison of the weight of sets of reason is well illustrated by a traditional symbol of justice, namely the balance.

4.2 Arguments

In line with the foregoing, our model of reasoning must account for defeasible reasoning. The following provides informally such abstract model of defeasible reasoning in which the process of reasoning is captured by deriving defeasible conclusions from premises.

A ratiocination proceeds through discrete reasoning steps by instancing some reasoning schemata. A reasoning schema has the form as below:

$$\frac{A_1, A_2, \dots, A_n}{B_1, B_2, \dots, B_n} \tag{4.1}$$

It is read as A_1, A_2, ..., A_n is a reason for B_1, B_2, ..., B_n. The A_i's are called the preconditions of the schema while the B_i's are its post-conditions. The set of preconditions in a schema is termed its reason while the set of post-conditions is termed its conclusion. The A_i's and the B_i's are statements symbolizing declarative sentences. Statements can be conditional (i.e. statement of the form if ... then ...) or non-conditional. For our purposes, non-conditional statements shall be expressed by literals. A literal is an atomic formula (or an atom) or the negation of an atomic formula. Atomic formulas are expressed in the language of predicate logic (see Section 7.2.2). For example, the statement "Mario is married" shall be formalized as the positive literal $married(\text{mario})$, and the "Mario is not married" as the negative literal $\neg married(\text{mario})$ where the symbol ¬ represents the negation. Conditional statements are expressed by rules relating statements to another statement and have the following form:

$$\alpha_1,\, \alpha_2,\, \dots\, \alpha_n \rightarrow \beta. \tag{4.2}$$

The α_i' s constitute the antecedents of the rule while the β is its consequent and they related are by the symbol →. For example, the rule expressing that unmarried males are bachelors can be formulated as follows:

$$male(x), \neg married(x) \rightarrow bachelor(x). \tag{4.3}$$

A very useful reasoning schema is *detachment* with which from the preconditions $\alpha_1, \alpha_2, \dots, \alpha_n$ and the rule $(\alpha_1,\, \alpha_2,\, \dots\, \alpha_n \rightarrow \beta)$, we conclude β:

$$\frac{\begin{array}{l}\alpha_1, \alpha_2, \ldots, \alpha_n,\\ \alpha_1, \alpha_2, \ldots, \alpha_n \rightarrow \beta\end{array}}{\beta} \tag{4.4}$$

For example, from the fact that Mario is a male and unmarried, and the rule expressing that unmarried males are bachelors, then we can derive that Mario is a bachelor:

$$\frac{\begin{array}{l}\mathit{male}(\text{mario}), \neg \mathit{married}(\text{mario}),\\ \mathit{male}(\text{mario}), \neg \mathit{married}(\text{mario}) \rightarrow \mathit{bachelor}(\text{mario})\end{array}}{\mathit{bachelor}(\text{mario})} \tag{4.5}$$

Conclusions of instantiated schemata may be used as premises for other schemata. Doing so, instantiated schemata are ‘chained’, and such chains are called *arguments*. We represent graphically an argument as an inference tree in which each vertice (or node) is labeled by a rule or a literal, and each compound arrow linking nodes corresponds to the application of a reasoning schema. The root is the conclusion and leafs are facts.

For example, if we consider the fact that Mario is a man and the rule that a man is a male, then we can conclude that Mario is a male. Given the fact that Mario is unmarried, and from the previous rule expressing that unmarried males are bachelors, then we can conclude that Mario is a bachelor. The tree associated to this argument is given in Figure 4.1.

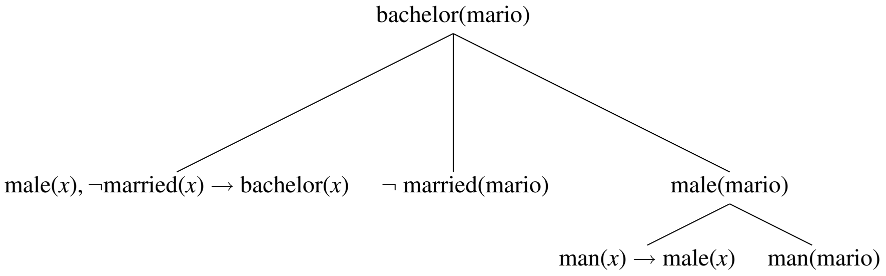


Fig. 4.1. An argument represented graphically as a tree.

We distinguish strict and defeasible rules. Strict rules use the symbol → while defeasible rules use the symbol ⇒. Hence a strict rule has the form:

$$\alpha_1, \alpha_2, \ldots, \alpha_n \rightarrow \beta, \tag{4.6}$$

while a defeasible rule a has the form below:

$$\alpha_1, \alpha_2, \ldots, \alpha_n \Rightarrow \beta. \tag{4.7}$$

For example, the policy rules expressing that students have discount is expressed as:

$$student(x) \Rightarrow \neg discount(x). \tag{4.8}$$

Both strict and defeasible rules can be used in the detachment schema. A schema whose preconditions contain a strict rule is strict whereas a schema whose preconditions contain a defeasible rule is defeasible. The distinction holds in that a conclusion of a strict schema can*not* be rejected on light of further information whereas a conclusion of a defeasible schema can be rejected on light of further information. For example, from the fact that Mario is student and the policy that students have discount we can conclude defeasibly that Mario has a discount. This is expressed by the following instantiated detachment schema:

$$\frac{\begin{array}{l} student(\text{mario}), \\ student(x) \Rightarrow discount(x) \end{array}}{discount(\text{mario})} \tag{4.9}$$

Strict or defeasible conclusions of instantiated schema may be used as premises for other schemata to build chains of schemata, i.e. arguments. An argument that does not use a defeasible detachment schema is strict while an argument that use a defeasible detachment schema is defeasible. For example, the argument given in Figure 4.1 is strict while the argument in (4.9) is defeasible. Some authors refer to provisional conclusions of defeasible schemata as *prima facie* conclusions to suggest that such conclusions are derived on the basis of information available to the agents. Some other authors as G. Sartor and A. Peczenik prefer the terminology of *pro-tanto* conclusions to suggest conclusions that can be withdraw on light of further information.

A reasoning based on strict schemata is a deductive reasoning while a reasoning based on strict and defeasible schemata is a defeasible reasoning.

Before moving to the next Section, we may argue that a negative literal $\neg\gamma$ may result from the non-derivation of the corresponding positive literal γ from a set of premises. The non-derivation of a certain literal can be related to an argumentative adaptation of the closed world assumption according which something is false if it is not currently proved to be true. Given a set of premises A, this can be captured by the following:

$$\frac{A \nvdash \gamma}{A \vdash \neg\gamma} \tag{4.10}$$

Notice this involves that from a finite set of premises, we may derive an infinity of negative literals, and this is problematic if we want to build a reasoner which reasons forward. Importantly, the negation $\neg$ in the schema given above is not a ‘sound’ or ‘strong’ negation as we intended negation initially but rather a ‘weak’ negation which, for example, in logic programming is traditionally called negation as failure. If we denote weak negation by ${}^{weak}\neg$, then the schema (4.10) should be more adequately replaced by the following:

$$\frac{A \not\vdash \gamma}{A \vdash^{weak} \neg\gamma} \tag{4.11}$$

The relation between strong negation $\neg$ and weak negation ${}^{weak}\neg$ goes beyond our scope, and we leave it as a matter of future research.

4.3 Defeaters and defeated

The previous Section isolated the concept of argument supporting some (defeasible) conclusions. In some circumstances, the detachment of the consequent of rule(s) may be defeated by other rule(s). In this Section, we present the notion of defeaters and the treatment of defeat.

4.3.1 Defeaters

As proposed by J. Pollock [166], and in most accounts of defeasible reasoning, two types of defeaters are distinguished: (i) *rebutting* defeaters which provide a reason for contradicting a conclusion, and (ii) *undercutting* defeaters which undermine the support of a conclusion without contradicting it.

In our setting, we have we are limited to account for undercutting and rebutting defeaters as the defeat of the detachment of the consequent (of a defeasible rule) by the detachment of a conflicting consequent. In this view, as proposed by [149], “the conflicting rule my either rebut the first by supporting a conflicting consequent, or it may be undercut the first rule by identifying a situation in which the rule does not apply.” Before moving to the account of undercutting and rebutting defeaters in our setting, it is important to make some remarks on conflicts detection.

Conflicting rules are rules whose consequents conflict, and conflicting consequents are defined as conflicting literals. In the following, the set of literals conflicting with a literal γ is denoted $\mathscr{C}_{\text{onflict}}(\gamma)$ or more shortly $\sim\gamma$. If γ is a positive literal q then $\sim\gamma$ is a negative literal $\neg q$, and if γ is a negative literal $\neg q$ then $\sim\gamma$ is a positive literal q. Among the possibility of conflicting literals, it is natural and standard to assume that a literal γ and its complement $\sim\gamma$ conflict. Hence in the following we assume that $\mathscr{C}_{\text{onflict}}(\gamma) = \neg\gamma$ and $\mathscr{C}_{\text{onflict}}(\neg\gamma) = \gamma$. For example, the literals *discount* and $\neg$*discount* are complement of each other, hence they conflict and we can write $\mathscr{C}_{\text{onflict}}(discount) = \neg discount$ and $\mathscr{C}_{\text{onflict}}(\neg discount) = discount$. We assume that if γ_1 conflicts with γ_2 then γ_2 conflicts with γ_1.

For example, consider that some inquiry indicates the policy that persons above 25 years are not entitled to discount. This policy is expressed by the following defeasible rule:

$$over\_25(x) \Rightarrow \neg discount(x). \tag{4.12}$$

Consider that Mario is over 25 years hold, then we can draw the following argument concluding defeasibly that Mario is not entitled of any discount.

$$\frac{\begin{array}{l} \mathit{over\_25}(\text{mario}), \\ \mathit{over\_25}(x) \Rightarrow \neg \mathit{discount}(x) \end{array}}{\neg \mathit{discount}(\text{mario})} \tag{4.13}$$

Since Mario is a student and is above 25, then Mario is pushed toward the incompatible conclusions of being entitled of a discount and not being entitled of a discount. Indeed, according the argument in (4.9), Mario as a student has no discount, while according the argument in (4.13) Mario has no discount since is above 25. On the assumption that the argument in (4.9) is a stronger reason than the argument in (4.13), then the former rebuts the latter. We shall return latter to the notion of strength of arguments.

At this stage, since we have only strict and defeasible rules in our hands, we are limited to account for rebutting defeaters without having the possibility to capture undercutting defeaters. In order to capture undercutting defeaters, following D. Nute [149], we introduce another type of rule called defeater. A defeater rule has the following form:

$$\alpha_1, \alpha_2, \ldots \alpha_n \rightsquigarrow \beta. \tag{4.14}$$

Defeater rules can be used in detachment schemata but are not intented to support the consequent: their purpose is just to undercut other rules. For example, if we replace the defeasible rule in (4.13), representing the policy that persons above 25 years are not entitled to discount, by a defeater rule:

$$\mathit{over\_25}(x) \rightsquigarrow \neg \mathit{discount}(x), \tag{4.15}$$

and given that Mario is over 25 years hold, then we can draw the following argument:

$$\frac{\begin{array}{l} \mathit{over\_25}(\text{mario}), \\ \mathit{over\_25}(x) \rightsquigarrow \neg \mathit{discount}(x) \end{array}}{\neg \mathit{discount}(\text{mario})} \tag{4.16}$$

If the argument in (4.16) is assumed to be stronger than the argument in (4.9), then the former undercut the latter. As the argument in (4.16) is a detachment of the consequent of a defeater rule, it cannot be used to support the conclusion $\neg \mathit{discount}(\text{mario})$.

In this setting, the distinction between rebutting and undercutting is captured thanks to the introduction of defeater rules along with strict and defeasible rules. Strict rules and defeasible rule are used as rebutting defeaters, while defeater rules are used as undercutting defeaters. To detect when an argument defeats another one, the notion of conflict is central. Accordingly, we say that an argument *S conflicts with* an argument *R* if and only if the arguments *S* and *R* contain two conclusions γ_1 and γ_2 such that γ_1 conflicts with γ_2 and vice versa. For example, the argument in (4.9) conflicts with the argument in (4.13) and vice versa.

4.3.2 Conflicts handling

Conflicts between arguments can be distinguished on the basis of the strict or defeasible nature of conclusions. Accordingly, conflicts may exist (i) between strict and defeasible conclusions, (ii) between defeasible conclusions, and (iii) between strict conclusions.

All models of defeasible reasoning handle the first type of conflicts by considering that strict conclusions override defeasible conclusions. In other words, given a strict and defeasible conclusion which are conflicting, the strict conclusion prevails while the defeasible conclusion is rejected. Accordingly, strict arguments cannot be defeated.

The second type of conflict between defeasible conclusions is resolved on the basis of a tier information, namely preferences. Many criteria are commonly used to settle preference between conclusions. For example, D. Poole [168, 169] proposed that if two arguments support conflicting conclusions, then the argument, which is based upon the most specific set of premises, defeats the other argument. We abstract to these criteria, and root preferences by an explicit strength order between rules. To do so, rules are identified by rule labels and a strength order is stabilized by the relation $\succ$ between two rules labels in order to indicate the relative strength of each rule. So, the formula $r_2 \succ r_1$ indicates that the rule labeled r_2 is stronger than the rule labeled by r_1. In this abstract account of preference, provided two applicable rules and a strength order between these two rules, one is allowed to conclude only for the consequent of the stronger rule. For example, consider the two following arguments:

$$\frac{\begin{array}{l} student(\text{mario}), \\ r_1: \quad student(x) \Rightarrow discount(x) \end{array}}{discount(\text{mario})} \tag{4.17}$$

$$\frac{\begin{array}{l} over\_25(\text{mario}), \\ r_2: \quad over\_25(x) \Rightarrow \neg discount(x) \end{array}}{\neg discount(\text{mario})} \tag{4.18}$$

These arguments support two conflicting defeasible conclusions, namely $\neg discount(\text{mario})$ and $discount(\text{mario})$. If we assume that the rule r_2 is stronger than rule r_1, i.e. $r_2 \succ r_1$, then we discard the conclusion $discount(\text{mario})$ to adopt the conclusion $\neg discount(\text{mario})$.

Accordingly, we say that an argument W *defeats* a defeasible argument S if and only if W and S contain two conclusions γ_2 consequent of a rule r_2 and γ_1 consequent of a rule r_1 respectively, such that γ_1 conflicts with γ_2 and the rule r_2 is stronger than the rule r_1 (i.e. $r_2 \succ r_1$). For example, the argument in (4.13) defeats the argument in (4.9) since the conclusion $discount(\text{mario})$ conflicts with the conclusion $\neg discount(\text{mario})$ and r_2 is stronger than r_1.

Conflicts between strict conclusions indicate an inconsistency in the theory. Since strict conclusions are not defeasible by nature, no preferences can applied to solve the conflict and the conflicting strict conclusions are derived.

4.4 Justified arguments

Comparing arguments by pairs is not enough since a defeating argument can in turn be defeated by other arguments. Indeed, an argument defeating a first argument can at its turn be defeated by a third argument, and in this case, the first argument is reinstated. We intend to determine justified arguments.

For example, given the fact that Mario is a member, and a provision indicating that members are entitled of a discount, we can conclude defeasibly that Mario is entitled of a discount:

$$\frac{\begin{array}{l} \mathit{member}(\text{mario}), \\ r_3\colon \quad \mathit{member}(x) \Rightarrow \mathit{discount}(x) \end{array}}{\mathit{discount}(\text{mario})} \tag{4.19}$$

Suppose that the rule r_3 is stronger than the rule r_2 (i.e. $r_3 \succ r_2$), then the argument in (4.19) defeats the argument (4.13). So, we have a first argument in (4.9) defeated by a second argument (4.13) which is at its turn defeated by a third argument (4.19): the first argument in (4.9) is reinstated and thus we can conclude that Mario is entitled of a discount.

Of course defeating arguments can be defeated by other arguments which can be at their turn defeated, and this pseudo-cycle can be repeated as much as needed. When the set of arguments defeating each others is unraveled, we end up with *justified arguments*. As a rule of thumb, an argument is justified if it is an undefeated argument, but the general criteria to determine them can turn out to be rather tricky in some cases. As a matter of fact, the set of justified arguments can be determined in many different ways (see Section 7.4.4) and no agreement has been reached so far how to compile such set.

As an illustration of the difficulty in deriving the right set of justified arguments, consider that no preferences exist between two conflicting defeasible arguments. In this case, conflicts may be resolved in two modes, namely the *skeptical* and the *credulous* one. These two modes yield different results as to what defeasible conclusions are warranted. Roughly, in the skeptical mode no conflicting conclusion is derived whereas in the credulous mode the maximum of possible defeasible conclusions (subject to a consistency requirement) are derived.

A well-known example from the literature (the so-called 'Nixon diamond') makes the distinction clear. Suppose that Nixon is both a Quaker and a Republican and that Quakers are pacifists whereas Republicans are not. Accordingly, we can build the arguments in (4.20) and (4.21). Figure 4.2 illustrates graphically the Nixon diamond.

$$\frac{\begin{array}{l} quaker(\text{nixon}), \\ quaker(x) \Rightarrow pacifist(x) \end{array}}{pacifist(\text{nixon})} \tag{4.20}$$

$$\frac{\begin{array}{l} republican(\text{nixon}), \\ republican(x) \Rightarrow \neg pacifist(x) \end{array}}{\neg pacifist(\text{nixon})} \tag{4.21}$$

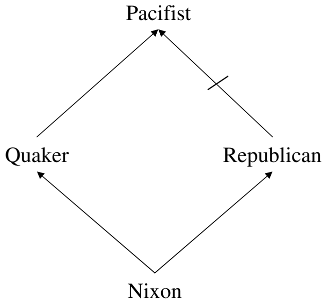


Fig. 4.2. The Nixon diamond.

In the credulous mode, we have no reason to prefer either conclusion to the other one, but will nevertheless commit to a conclusion, namely, either Nixon is a pacifist or Nixon is not a pacifist. In the skeptical mode instead, the reasoning agent recognizes that there is an incompatibility inferences and refrains from drawing either one.

Whether reasoning should be credulous or skeptical is largely debated, and no agreements has been reached so far: the commitment for one or the other is sometimes considered as a matter of taste. On this aspect, J. Pollock in [167] suggests to make a distinction on the basis of epistemic and practical reasoning:

> [...] the controversy over skeptical and credulous reasoning stems from a confusion of epistemic reasoning with practical reasoning. In practical reasoning, if one has no basis for choosing between two alternative plans, one should choose at random. The classical illustration is the medieval tale of Buridan's ass who starved to death standing midway between two equally succulent bales of hay because he could not decide from which to eat. [...] If

> the evidence favoring two alternative hypothesis is equally good, the agent should record that fact and withhold belief. Subsequent practical reasoning can then decide what to do given that epistemic conclusion. In some cases it may be reasonable to choose one of the hypotheses at random and act *as if* it is known to be true, and in other cases more caution will be prescribed. [...] Epistemic reasoning should acknowledge ignorance when it is encountered rather than drawing conclusions at random.

Such view is related to the observation that in some cases, the agent's epistemic ignorance makes the expected values of the different plans equal, and either plan is preferable to no plan at all. In some other cases, instead, the agent ignorance makes the expected value of doing nothing higher than the expected value of either plan. To sum up, J. Pollock in [167] argues:

> [...] a rational agent should acknowledge its ignorance and take that into account in computing the expected value of plans.

For example, in a medical diagnosis scenario, on the evidence supporting two diseases, it could be disastrous to treat the patient for one disease rather than the other. In our setting, we shall commit to skeptical reasoning, leaving the integration of credulous reasoning for future investigations.

As another complication, consider the case in which for any argument in support of a defeasible conclusion γ there is also an equally good argument against it.

$$\frac{\begin{array}{l} p_1, \\ r_1:\quad p_1 \Rightarrow q \end{array}}{q} \tag{4.22}$$

$$\frac{\begin{array}{l} p_2, \\ r_2:\quad p_2 \Rightarrow q \end{array}}{q} \tag{4.23}$$

$$\frac{\begin{array}{l} p_3, \\ r_3:\quad p_3 \Rightarrow \neg q \end{array}}{\neg q} \tag{4.24}$$

$$\frac{\begin{array}{l} p_4, \\ r_4:\quad p_4 \Rightarrow \neg q \end{array}}{\neg q} \tag{4.25}$$

Suppose that $r_1 \succ r_3$ and $r_2 \succ r_4$. On the one hand, for any argument supporting the defeasible conclusion q, there is an argument of equal strength supporting $\neg q$. In this view, we are tempted to admit that the conclusion q is not justified. On the other hand, for any argument supporting $\neg q$, there is a stronger argument supporting q, and in this view, we are tempted to admit that the conclusion q is justified.

If we assume this last view, then we assume so-called *team defeat*. So far, there is no collective agreement on whether team defeat should be assumed in argumentation: in the formal part of this dissertation, the formalization in defeasible logic will account for both views in the sense that defeasible logic can be easily tuned to cater or not for team defeat, leaving the ultimate choice to the users.

Notice that team defeat should not be confused with J. Pollocks collective defeat [166], in which, given a set of arguments, any argument of the set is defeated by another argument, and no argument of the set is defeated by an undefeated argument outside the set.

A final remark concerns competing defeasible arguments in which a defeasible part is in upstream with respect to a strict one. For example, consider the following rules [51]:

$$\begin{aligned} &wr \Rightarrow m,\\ &m \rightarrow hw,\\ &go \Rightarrow b,\\ &b \rightarrow \neg hw, \end{aligned} \tag{4.26}$$

where wr stands for "John wears something that looks like a wedding ring", m for "John is married", hw for "John has a wife", go for "John often goes out until late with his friends" and b for "John is a bachelor". Given the facts wr and go, we can build an argument supporting hw and another argument supporting $\neg hw$: we have a conflict to solve and there is no strength order among rules.

Since the incompatible conclusions hw and $\neg hw$ are supported by defeasible arguments, we refute hw and $\neg hw$. Moreover, though m and b are intuitively incompatible, they do not formally conflict and thus we can conclude m and b, that is, an incoherency. On this basis, we can argue that we derive an incoherency because the theory itself is not coherent. To recover coherency, we can change the initial theory by specifying explicitly that m and b conflicts ($m \in \mathscr{C}_{\text{onflict}}(b)$) or by adding the rule indicating that bachelors are not married: $b \rightarrow \neg m$.

Another possibility to recover coherency lies in the assumption that incompatibility of some defeasible statements supported by strict rules implies the incompatibility of some defeasible statements supporting the formers. For example, since the incompatible conclusions hw and $\neg hw$ are supported by two strict rules, we may consider that the conclusions b and m are also indirectly incompatible. In this view and in the setting of ambiguity blocking, we refute the conclusions hw and $\neg hw$ as the conclusions b and m. We shall call such mode of resolving conflicts 'coherency recovering' because starting from a incoherent theory, we recover coherent results. To do so, a simple solution consists in considering partial contraposition of strict rules [51]. In the example, by considering the contraposition $\neg hw \rightarrow \neg m$, we can build an argument attacking m and thus recover coherency.

For our purposes, we shall assume no mechanism to recover coherency. In particular, we shall not use contraposition, leaving it for future investigations. Hence, in our setting, the consequent of a conditional can be derived if its antecedent holds, but the negation of its antecedent cannot be derived if the negation of its consequent holds.

5

Cognitive model

The Chapter 4 set up a defeasible reasoning framework. In this Chapter, this framework is extended progressively to account for some cognitive aspects of agents. As autonomous agents may follow or violate norms, this cognitive account allows us to model normative bindingness.

A cognition of agent is traditionally analyzed by means of two types of cognition: *epistemic cognition*, which concerns what to believe and *practical cognition*, which concerns what to do. The distinction between epistemic and practical cognition is disputable since both are usually interdependent: epistemic cognition provides some beliefs which are required by practical reasoning, and practical reasoning is necessary to guide inquiries. We keep here the distinction. Both types of cognition require ratiocinative processes and non-ratiocinative processes (e.g. perception), but in the remainder, we shall focus only on the ratiocinative processes, and leave non-ratiocinative processes for possible future investigations.

In Sections 5.1, 5.2 and 5.3, we account for some aspects of epistemic, practical and normative cognitions respectively. Reasoning schemata for these three types cognition are then provided into Section 5.5. Finally, in Section 5.4, we address conflicts among conclusions which allow us a cognitive account of normative bindingness.

5.1 Epistemic cognition

Epistemic cognition allows an agent to represent its environment and reason on it. G. Sartor writes in [191]:

> An agent endowed with the faculty of epistemic cognition processes external inputs and obtain epistemic states. Then the agent reasons, producing new mental states on the basis of the epistemic states the agent already has. Epistemic reasoning is indeed the process through with one builds new epistemic states moving from the epistemic states one already possesses.

In our setting, the term ‘epistemic’ may involve some confusion because it is usually referred to the notion of knowledge whereas we are interested by (dis)beliefs and (in)existences of states of affairs. The distinction between the notions of knowledge and belief has a long philosophical tradition since antiquity. In epistemology, the dedicated branch of philosophy concerned with knowledge and beliefs, knowledge is traditionally defined as justified true belief. Here, we shall not enter in this still on-going debate among philosophers, and do not address the distinction between the two notions.

5.1.1 (Dis)beliefs and (in)existences

An agent *ag* endowed with epistemic cognition processes external inputs to form internal states to represent and reason on its environment. Such internal states are called *epistemic states* and consists of sets of *beliefs*. In the following, we interpret beliefs of the type "an agent *ag* believes in γ" as "It holds, from the viewpoint of an agent *ag*, that γ", and we shall formalize it as:

$$\mathrm{Hold}_{ag}\gamma. \tag{5.1}$$

For example, the statements "It holds, from the viewpoint of Mario, that he is student" and "It holds, from the viewpoint of Guido, that Mario is not a student" are written as:

$$\mathrm{Hold}_{\mathrm{mario}} student(\mathrm{mario}), \tag{5.2}$$

$$\mathrm{Hold}_{\mathrm{guido}} \neg student(\mathrm{mario}). \tag{5.3}$$

In the remainder, beliefs of the type "an agent *ag* believes in γ" is interpreted as 'It holds, from the viewpoint of an agent *ag*, that γ', but the inverse is not true as we shall see in the temporal setting of the next Chapter. However, in order to ease the discussion, both forms shall be referred to as beliefs.

A disbelief as the refusal to believe something is captured by the negation of a belief. For example, that "Mario disbelieves (does not believe) that he is not a student" and "Guido disbelieves (does not believe) to be a student" can be written as:

$$\neg\mathrm{Hold}_{\mathrm{mario}} \neg student(\mathrm{mario}), \tag{5.4}$$

$$\neg\mathrm{Hold}_{\mathrm{guido}} student(\mathrm{guido}). \tag{5.5}$$

The notation $\mathrm{Hold}_{ag}\gamma$ indicates that γ holds from the viewpoint of agent *ag*, accordingly γ holds subjectively. To indicate that γ holds objectively irrespectively to any agent, we use the notation $\mathrm{Hold}_{\mathrm{obj}}\gamma$. For instance, the statement "It holds objectively that Mario is Italian" is formulated as:

$$\mathrm{Hold}_{\mathrm{obj}} italian(\mathrm{mario}). \tag{5.6}$$

Formulas of the form $\mathrm{Hold}_{\mathrm{obj}}\gamma$ are pieces of information holding objectively which may not be accessible by any agent while subjectives (dis)believes are pieces of information holding subjectively by an agent who may have an 'access' to it for example by introspection as we shall see soon. Formulas of the form $\mathrm{Hold}_{\mathrm{obj}}\gamma$ refer to objective existence, whereas formulas of the form $\mathrm{Hold}_{ag}\gamma$ (with ag different to obj) are called beliefs.

Notice that we assume that a plain literal denotes the objective existence of the state of affair expressed by the plain literal. For example, the formula $italian(\mathrm{mario})$ is also another representation of the statement "Objectively, Mario is Italian". However, we believe that the use of the notation $\mathrm{Hold}_{\mathrm{obj}}$ helps to remove ambiguities on the epistemic status of plain literals, and, as we will see in the next Chapter, it prepares an adequate notation for our temporal setting.

A combination of (dis)beliefs and (in)existences of state of affairs is represented by a sequence of the operator Hold_{ag}. For example, the statement "Guido believes that Mario disbelieves that Guido is a student" is written as:

$$\mathrm{Hold}_{\mathrm{guido}}\neg\mathrm{Hold}_{\mathrm{mario}}student(\mathrm{guido}). \tag{5.7}$$

In the remainder, a sequence of epistemic operators shall be noted $(X_i)_{1..n}$ where X_i stands for Hold_{ag} or $\neg\mathrm{Hold}_{ag}$. Accordingly, an epistemic statement shall have the form $(X_i)_{1..n}\gamma$ where γ is a literal.

In the previous Chapter, rules were introduced to relate some literals to another literal. With the introduction of (dis)beliefs, rules can relate (dis)beliefs $(\neg)\mathrm{Hold}_{ag}\alpha$ and objective (in)existence of state of affairs $(\neg)\mathrm{Hold}_{\mathrm{obj}}\alpha$. Accordingly, a rule shall have the following form:

$$r\colon \quad \alpha_1,\ldots,\alpha_n \hookrightarrow \beta. \tag{5.8}$$

where $\alpha_1,\ldots,\alpha_n,\beta$ are epistemic statements, and $\hookrightarrow$ stands for either $\rightarrow$, $\Rightarrow$ or $\rightsquigarrow$. For example, the conditional statement "If Mario is Italian and Mario believes that the wine x is Italian, then Mario believes that the wine x is good" can be formulated as:

$$r_2\colon \quad \mathrm{Hold}_{\mathrm{obj}}italian(\mathrm{mario}), \mathrm{Hold}_{\mathrm{mario}}italian\_wine(x) \Rightarrow \mathrm{Hold}_{\mathrm{mario}}good(x). \tag{5.9}$$

Rules too can hold objectively or with respect to an agent beliefs. For example, the statement "Mario believes that if one is 28 years old, then one is above 25 years old" can be expressed by:

$$\mathrm{Hold}_{\mathrm{mario}}(r_1\colon \quad \mathrm{Hold}_{ag}has\_28(x) \rightarrow \mathrm{Hold}_{ag}over\_25(x)). \tag{5.10}$$

Next, we cater for the detachment of the consequent of such epistemic rules.

5.1.2 Detachment schema in epistemic reasoning

The process through which an agent derives new beliefs from other beliefs is called *epistemic reasoning*. An important schema in epistemic cognition is epistemic detachment:

$$\frac{\begin{array}{l}(X_i)_{1..n}\alpha_1, (X_i)_{1..n}\ldots, (X_i)_{1..n}\alpha_n,\\ (X_i)_{1..n}(r\colon \quad \alpha_1, \ldots, \alpha_n \hookrightarrow \beta)\end{array}}{(X_i)_{1..n}\beta} \tag{5.11}$$

where $\hookrightarrow$ is either $\rightarrow$, $\Rightarrow$ or $\leadsto$, and $(X_i)_{1..n}$ is a (non-empty) sequence of epistemic operators. In line with the previous Chapter 4, we can have strict epistemic detachments and defeasible ones. A strict epistemic detachment schema has a strict rule in its pre-conditions:

$$\frac{\begin{array}{l}(X_i)_{1..n}\alpha_1, \ldots, (X_i)_{1..n}\alpha_n,\\ (X_i)_{1..n}(r\colon \quad \alpha_1, \ldots, \alpha_n \rightarrow \beta)\end{array}}{(X_i)_{1..n}\beta} \tag{5.12}$$

For example, given the definite conditional "It holds objectively that if Mario believes that one is 28 years old, then Mario believes that one is above 25 years old" and the statement "Objectively, Mario believes that he is 28 years old", we can form the following instance of the epistemic detachment:

$$\frac{\begin{array}{l}\mathrm{Hold_{obj}Hold_{mario}}\mathit{has\_28}(\mathrm{mario}),\\ \mathrm{Hold_{obj}}(r_1\colon \quad \mathrm{Hold_{mario}}\mathit{has\_28}(x) \rightarrow \mathrm{Hold_{mario}}\mathit{over\_25}(x))\end{array}}{\mathrm{Hold_{obj}Hold_{mario}}\mathit{over\_25}(\mathrm{mario})} \tag{5.13}$$

A defeasible epistemic detachment schema has a defeasible rule in its pre-conditions:

$$\frac{\begin{array}{l}(X_i)_{1..n}\alpha_1, \ldots, (X_i)_{1..n}\alpha_n,\\ (X_i)_{1..n}(r\colon \quad \alpha_1, \ldots, \alpha_n \Rightarrow \beta)\end{array}}{(X_i)_{1..n}\beta} \tag{5.14}$$

For example, given the statements "If Mario is Italian and Mario believes that the wine x is Italian, then Mario believes that the wine x is good" and "Mario is Italian" and "Mario believes that Chianti is an Italian wine", then we conclude defeasibly "Mario believes that Chianti is a good wine".

$$\frac{\begin{array}{l}\mathrm{Hold_{obj}}\mathit{italian}(\mathrm{mario}),\\ \mathrm{Hold_{mario}}\mathit{italian\_wine}(\mathrm{chianti}),\\ r_1\colon \quad \mathrm{Hold_{obj}}\mathit{italian}(\mathrm{mario}), \mathrm{Hold_{mario}}\mathit{italian\_wine}(x) \Rightarrow \mathrm{Hold_{mario}}\mathit{good}(x)\end{array}}{\mathrm{Hold_{mario}}\mathit{good}(\mathrm{chianti})} \tag{5.15}$$

Next, we investigate the phenomena of introspection and its dual, reflexion.

5.1.3 Introspection and reflexion

If an agent can observe and sense its environment, he may also be able to do so with its mental states. Such agents are called *reflexive* agents whereas agents that cannot sense or observe their mental states are said *planar*. A reflexive agent may hence form some beliefs about his beliefs and disbeliefs. The formation of beliefs about beliefs is captured by the schema of *positive introspection*:

$$\frac{\mathrm{Hold}_{ag}\gamma}{\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma} \tag{5.16}$$

while the formation of beliefs about disbeliefs is captured by the schema of *negative introspection*:

$$\frac{\neg\mathrm{Hold}_{ag}\gamma}{\mathrm{Hold}_{ag}\neg\mathrm{Hold}_{ag}\gamma} \tag{5.17}$$

For example, given "Mario believes to be a student", we can derive "Mario believes that Mario believes to be a student":

$$\frac{\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})}{\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})} \tag{5.18}$$

Believed rules also can be object of introspection. For example, given that Mario believes that students are entitled to discount, then we can conclude that Mario believes that Mario believes that students are entitled to discount:

$$\frac{\mathrm{Hold}_{\mathrm{mario}}(r\colon\quad \mathit{student}(x) \Rightarrow \mathit{discount}(x))}{\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}(r\colon\quad \mathit{student}(x) \Rightarrow \mathit{discount}(x))} \tag{5.19}$$

The inverse of introspection, called here reflexion, is captured by the following schemata:

$$\frac{\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma}{\mathrm{Hold}_{ag}\gamma} \tag{5.20}$$

$$\frac{\mathrm{Hold}_{ag}\neg\mathrm{Hold}_{ag}\gamma}{\neg\mathrm{Hold}_{ag}\gamma} \tag{5.21}$$

For example, if Mario believes that Mario believes to be a student then Mario believes to be a student:

$$\frac{\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})}{\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})} \tag{5.22}$$

In the foregoing, both phenomena of introspection and reflexion are captured at the level of reasoning schemata. However, it is arguable that both phenomena are defeasible with respect to the substantial contents of beliefs and some doxastic preferences. Since preferences stand between rules, both phenomena of introspection and reflexion are more natural expressed at the rule level. For this reason, the schemata of positive introspection (5.16) and negative introspection (5.17) shall be replaced by the following defeasible schema rules:

$$r_{\mathrm{Hold}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma}\colon\quad \mathrm{Hold}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma, \tag{5.23}$$

$$r_{\neg\mathrm{Hold}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Hold}_{ag}\gamma}\colon\quad \neg\mathrm{Hold}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Hold}_{ag}\gamma. \tag{5.24}$$

Notice that the schema rules are labeled by their own content to insure their unique labeling, and so that a preference can be set among rule labels. For example, the schema (5.18) shall be replaced by the following detachment schema:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario}),\\ r_{\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})\Rightarrow\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})}\colon\\ \qquad\quad \mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})\Rightarrow\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})\end{array}}{\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathit{student}(\mathrm{mario})} \tag{5.25}$$

Similarly, we replace the schemata (5.20) and (5.21) by the defeasible schema rules:

$$r_{\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}\colon\quad \mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma, \tag{5.26}$$

$$r_{\mathrm{Hold}_{ag}\neg\mathrm{Hold}_{ag}\gamma\Rightarrow\neg\mathrm{Hold}_{ag}\gamma}\colon\quad \mathrm{Hold}_{ag}\neg\mathrm{Hold}_{ag}\gamma\Rightarrow\neg\mathrm{Hold}_{ag}\gamma. \tag{5.27}$$

Beside introspection and reflexion, we assume that given any state of affairs, then we can entail defeasibly that this state holds objectively. Accordingly, we have the following rule:

$$r_{\gamma\Rightarrow\mathrm{Hold}_{\mathrm{obj}}\gamma}\colon\quad \gamma\Rightarrow\mathrm{Hold}_{\mathrm{obj}}\gamma, \tag{5.28}$$

and the reverse:

$$r_{\mathrm{Hold}_{\mathrm{obj}}\gamma\Rightarrow\gamma}\colon\quad \mathrm{Hold}_{\mathrm{obj}}\gamma\Rightarrow\gamma. \tag{5.29}$$

Finally, in epistemic setting, new conflicts may arise from the collisions of incompatible beliefs. For example, the conclusions “Mario believes that Chianti is a good wine” and “Mario believes that Chianti is not a good wine” are incompatible. Thus, a (dis)belief, which is supported by chaining some reasoning schemata (i.e. an argument), may not be endorsed by an agent because the argument supporting it may be defeated by some other arguments. The discussion on conflicts between epistemic statements is postponed to Section 5.5.

5.2 Practical cognition

Though epistemic cognition allows an agent to represent and reason on its environment, it does not provide any means to guide the behavior of the agent. To guide its behavior, an agent requires the so-called practical cognition.

An agent endowed with practical cognition forms internal states called *conative states* composed of mental attitudes as desires, goals, intentions, etc. [191]. A rational agent directs its behavior to render its situation more likable. To do so, an agent selects some goals and enters in a phase of planning by building and selecting some plans to reach these goals. A plan is the endorsement of a set of actions to be performed and shall then guide the agent's conduct.

5.2.1 Desires

Desires (or goals) indicate the states of affairs which an agent aims to reach. In the following, a desire is indicated by the operator Des_{ag} where *ag* is a variable denoting the agent bearing the desire. Accordingly, the agent *ag* desires γ is formalized as follows:

$$\mathrm{Des}_{ag}\gamma, \tag{5.30}$$

where γ is the desired state of affairs. For example, the statement "Guido desires to be entitled of a discount" and "Mario desires to not be member" can be formalized as:

$$\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}), \tag{5.31}$$

$$\mathrm{Des}_{\mathrm{mario}}\neg \mathit{member}(\mathrm{mario}). \tag{5.32}$$

Notice that we do not permit the subscript of the operator Des with obj because we do not have an (intuitive) interpretation of formulas as $\mathrm{Des}_{\mathrm{obj}}\gamma$.

A desire can be negated, to indicate for example agents' aversion. For instance, the statement "Guido does not desire to not be entitled of a discount", and "Mario does not desires to be member" can be represented by:

$$\neg\mathrm{Des}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}), \tag{5.33}$$

$$\neg\mathrm{Des}_{\mathrm{mario}}\mathit{member}(\mathrm{mario}). \tag{5.34}$$

As for epistemic reasoning in which we have rules relating literals and (dis)beliefs, in pratical reasoning we can have rules relating epistemic statements and (negated) desires. For example, given the statements "Mario believes that if Guido believes to not have any discount, then Guido desires to get a discount" and "Mario believes that Guido believes to not have a discount", we can derive that "Mario believes that Guido desires a discount". This can be expressed by the following instantiated detachment reasoning schema:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}),\\ \mathrm{Hold}_{\mathrm{mario}}(r\text{:}\quad \mathrm{Hold}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}) \Rightarrow \mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}))\end{array}}{\mathrm{Hold}_{\mathrm{mario}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido})} \tag{5.35}$$

In Section 5.4.2, we will present the detachment reasoning schema for the general case in practical reasoning.

As many desires can be (defeasibly) derived, bounded agents may be unable to satisfy all the undefeated desires. Hence, an agent shall select some desires. This selection process can be made via certain kinds of utilities associated with the desired states. On this approach, a utility can be a numerical value representing how ‘good’ the desired state is: the higher the utility is, the better. The task of the agent, then, is to bring about states that maximize utility. The main disadvantage is that it is often difficult to derive an appropriate utility function. An alternative is to encode the utility into the logic theory by specifying that in some conditions, some desires discard some others.

5.2.2 Actions

Since norms and practical reasoning are about the behaviors of agents, the concept of action is crucial for our purpose. A huge literature analyzing the concept of action exists. It is common to root the concept of action in the distinction between the static and dynamic aspects of a system. The static aspects are described by *properties* while dynamic aspects are captured by *occurrences*. Further analysis of the class occurrence in terms of subclasses is largely debated in the literature. As an example of ontology, J. F. Allen in [5] proposes that the class occurrence is divided in two subclasses, *processes* and *events*:

> Processes refer to some activity not involving a culmination or anticipated result, such as the process denoted by the sentence, “I am running”. Events describe an activity that involves a product or outcome, such as the event denoted by the sentence “I walked to the store”.

An action is eventually another subclass of the class occurrence involving an agent. Actions can be processes (e.g. “I am running”) or events (e.g. “I walked to the store”). In this view, an agent and an occurrence constitute the action of the agent causing the occurrence. An action of which the occurrence is an event, is termed a *performance*, while an action of which the occurrence is a process is an *activity*.

Such ontologies are undoubtedly invaluable but are not catered for here because not essential for our purpose. Closer to our approach is the work of G. Sartor in [191] who proposes two characterizations of actions:

- a behavioral characterization, describing the type of action, and
- a productive characterization, describing the state produced by the action.

For example, “Guido joined” is a behavioral characterization, while “Guido is member” is a productive characterization. Consequently, two types of action can be considered:

- behavioral actions, which are described by the type of behavior, and
- productive actions, which are described by the type of produced state.

These two types of action are represented by the two following action operators:

- Do_{ag} for behavioral actions, and
- Bring_{ag} for productive actions.

where *ag* indicates the acting agent. For example, the behavioral action "Guido joined" and the productive action "Guido brings about his membership" can be expressed by:

$$\text{Do}_{\text{guido}} join(\text{guido}), \tag{5.36}$$

$$\text{Bring}_{\text{guido}} member(\text{guido}). \tag{5.37}$$

Productive actions can be interpreted in terms of causality: that an agent *ag* brings a state of affair means that *ag* causes this state of affair. As for desires, notice that we do not permit the subscript of the operator Bring or Do with obj because we do not have an (intuitive) interpretation for formulas as $\text{Bring}_{\text{obj}}\gamma$ or $\text{Do}_{\text{obj}}\gamma$.

Actions can be negated. On this aspect, the negation of an action and the omission of the same action are distinguished:

- $\text{Bring}_{ag}\neg\gamma$ for the negation, and
- $\neg\text{Bring}_{ag}\gamma$ for the omission.

For instance, the productive negated action "Mario brings about his non membership" can be expressed by:

$$\text{Bring}_{\text{mario}}\neg member(\text{mario}), \tag{5.38}$$

while the omission of the productive action "Mario does not bring that is member" is expressed by:

$$\neg\text{Bring}_{\text{mario}} member(\text{mario}). \tag{5.39}$$

For combinatorial reason, the negation of a behavioral action and its omission are considered.

- $\text{Do}_{ag}\neg\gamma$ for the negation, and
- $\neg\text{Do}_{ag}\gamma$ for the omission.

A confusion may appear about a behavioral account of a negated action. For instance, what is the negation of the action corresponding to the verb "to join"? If it is about leaving then we deal with the 'positive' action of leaving, and this may be expressed as follows:

$$\text{Do}_{\text{mario}} leave(\text{mario}). \tag{5.40}$$

However, the following notation which intends to be equivalent is somewhat suspect.

$$\text{Do}_{\text{mario}}\neg join(\text{mario}). \tag{5.41}$$

More intuitive is the omission of a behavioral action. For example, the omission of the behavioral action "Mario does not join" is expressed by:

$$\neg \mathrm{Do}_{\mathrm{mario}}\, \mathit{join}(\mathrm{mario}). \tag{5.42}$$

If it is acknowledged that a behavioral characterization for the negation of an action may be awkward in some cases, then it is perhaps appropriate to notice one characterization is often reducible to the other. As remarked by [191], productive actions can be expressed using behavioral actions by observing that an agent *ag* who brings it about that γ is equivalent to *ag* who does the action of bringing about that γ:

$$\mathrm{Bring}_{ag}\gamma \equiv \mathrm{Do}_{ag}(\mathit{bring\ about}\ \gamma). \tag{5.43}$$

Inversely, a behavioral action can be expressed using productive actions by observing that an agent *ag* who does γ is equivalent to *ag* who brings it about that *ag* does γ:

$$\mathrm{Do}_{ag}\gamma \equiv \mathrm{Bring}_{ag}(\mathit{ag\ does}\ \gamma). \tag{5.44}$$

On this setting, rules can be enriched with the concept of action. For example, the conditional statement "If Guido believes to be not entitled of any discount, Guido desires to be entitled of any discount, then Guido brings about being a member" can be expressed by the following defeasible rule:

$$\begin{aligned} r\!: \quad & \mathrm{Hold}_{\mathrm{guido}} \neg \mathit{discount}(\mathrm{guido}), \mathrm{Des}_{\mathrm{guido}} \mathit{member}(\mathrm{guido}) \\ & \Rightarrow \mathrm{Bring}_{\mathrm{guido}} \mathit{member}(\mathrm{guido}). \end{aligned} \tag{5.45}$$

By applying the detachment schema, and under the appropriate conditions, we derive $\mathrm{Bring}_{\mathrm{guido}} \mathit{member}(\mathrm{guido})$.

Modeling dynamics of a system via actions is confronted to at least three well-known problems, namely the *qualification problem*, the *frame problem* and the *ramification problem*. Each of these problems are briefly investigated below for our framework.

The qualification problem arose with the difficulties in specifying all the conditions which must be fulfilled to get an effective action. This problem was early a central point of research in artificial intelligence, and, J. McCarty in [135] explains:

> The 'qualification problem', immediately arose in representing general common sense knowledge. It seemed that in order to fully represent the conditions for the successful performance of an action, an impractical and implausible number of qualifications would have to be included in the sentences expressing them. For example, the successful use of a boat to cross a river requires, if the boat is a rowboat, that the oars and row locks be present and unbroken, and that they fit each other. Many other qualifications can be added, making the rules for using a rowboat almost impossible to apply, and yet anyone will still be able to think of additional requirements not yet stated.

The idea for overcoming the qualification problem is to assume the defeasible successful performance of an action, and then to consider eventual reasons against its success. If no reason against the defeasible success of an action can be advanced, then the effect of the action is derived. Accordingly, an important schema for actions is about the defeasible success of an action: performing a productive action entails the corresponding state of affair. This is expressed by the following schema:

$$\frac{\mathrm{Bring}_{ag}\gamma}{\mathrm{Hold}_{\mathrm{obj}}\gamma} \tag{5.46}$$

For example, if Guido performs the action of bringing his membership, then we conclude defeasibly that it objectively holds that Guido is a member:

$$\frac{\mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido})}{\mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{guido})} \tag{5.47}$$

In the foregoing, in order to tackle the qualification problem, the defeasible successful performance of actions is treated at the level of reasoning schemata. However, it is arguable that both the successful performance of an action is defeasible with respect to the substantial type of the action, and some strength order between its default successfulness and its unsuccessfulness. Since preferences stand between rules, the successful performance is more naturally expressed at the rule level. For this reason, the schemata of defeasible successful performance (5.46) shall be replaced by the following defeasible schema rule:

$$r_{\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{\mathrm{obj}}\gamma}: \quad \mathrm{Bring}_{ag}\gamma \Rightarrow \mathrm{Hold}_{\mathrm{obj}}\gamma. \tag{5.48}$$

The schema rules are labeled by their own contents to ensure their unique labeling, and so that a preference can be set among rule labels. For example, the schema in (5.47) shall be replaced by the following rule:

$$\begin{aligned} &r_{\mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido})\Rightarrow\mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{guido})}: \\ &\qquad \mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido}) \Rightarrow \mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{guido}). \end{aligned} \tag{5.49}$$

Given the above rule and the premise $\mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido})$, we build the following detachment schema:

$$\frac{\begin{aligned} &\mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido}), \\ &r_{\mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido})\Rightarrow\mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{guido})}: \\ &\qquad \mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido}) \Rightarrow \mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{guido}) \end{aligned}}{\mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{guido})} \tag{5.50}$$

The successful performance of the attempt $\mathrm{Bring}_{\mathrm{guido}}\mathit{member}(\mathrm{guido})$ shall then depend on the other applicable rules supporting the contrary state $\mathrm{Hold}_{\mathrm{obj}}\neg\mathit{member}(\mathrm{guido})$.

Notice that the schema rule in (5.48) assumes that performing a productive action entails the objective existence of the corresponding state of affair (indicated by the operator Hold_{obj}). Arguably, we could also assume that performing a productive action involves the subjective belief of the corresponding state of affair (indicated by the operator Hold_{ag}), in this case we would have:

$$r_{\text{Bring}_{ag}\gamma \Rightarrow \text{Hold}_{ag}\gamma}: \quad \text{Bring}_{ag}\gamma \Rightarrow \text{Hold}_{ag}\gamma. \tag{5.51}$$

However, it is also arguable that, if a productive action is performed, first, the corresponding state of affair holds objectively, then the agent observes this state of affairs, and eventually the agent believes in this state of affair. In this view, we would have to discard the reasoning schema given above, and eventually, we can simulate the perception of the state of affairs by some kinds of rules of the following form:

$$\text{Hold}_{\text{obj}}\gamma, \text{Do}_{ag}\mathit{observe}(\gamma) \Rightarrow \text{Hold}_{ag}\gamma. \tag{5.52}$$

As a matter of fact, it seems that in some circumstances, performing a productive action involves the subjective belief of the corresponding state of affair (indicated by the operator Hold_{ag}), whereas in other circumstances, performing a productive action does not involve the subjective belief of the corresponding state of affair. In other words, we retrieve basically the qualification problem. To overcome it, we shall assume that agent believes by default and defeasibly in the successful performance of her action, and then consider eventual reasons against her beliefs. On this basis, we assume both the schema rules in (5.48) and in (5.51).

Beside the qualification problem, the frame problem [136] is the difficulty of formalizing all the things that remain unchanged when an action is performed whereas the ramification problem is the difficulty of formalizing all the things that do change as the result of an action. In our setting, the ramification problem is handled by specifying some rules aiming at relating the action (or the result of the action) to the changing aspects of the system. Concerning the frame problem, as the issue is in fact a temporal one, we shall discuss it in our temporal model in Section 6.

Orthogonally to the qualification, frame and ramification problem, it is sometimes assumed the principle according which, for both behavioral and productive actions, doing separate actions entails doing their combination:

$$\frac{\text{Bring}_{ag}\gamma_1 \quad \mathit{and} \quad \text{Bring}_{ag}\gamma_2}{\text{Bring}_{ag}(\gamma_1 \quad \mathit{and} \quad \gamma_2)} \tag{5.53}$$

$$\frac{\text{Do}_{ag}\gamma_1 \quad \mathit{and} \quad \text{Do}_{ag}\gamma_2}{\text{Do}_{ag}(\gamma_1 \quad \mathit{and} \quad \gamma_2)} \tag{5.54}$$

Finally, in legal reasoning and common-sense reasoning, actions can have an intentional character or not. If an action $\text{Bring}_{ag}\gamma$ (or $\text{Do}_{ag}\gamma$) is performed by an agent ag then it is sometimes inferred that the agent ag has the desire to bring about the state γ:

$$\frac{\text{Do}_{ag}\gamma}{\text{Des}_{ag}\gamma} \tag{5.55}$$

$$\frac{\text{Bring}_{ag}\gamma}{\text{Des}_{ag}\gamma} \tag{5.56}$$

Though the above inferences (5.53) - (5.56) could be added to the framework, we do not account for such inferences in the remainder for the sake of simplicity.

5.2.3 Plans and intentions

When an agent has selected a desire, then it has to satisfy it. To satisfy this desire, it enters in a phase of planning, to build plans.

A plan is the endorsement of a set of operations to be performed. These plans shall then guide the latter conduct of the agent. Note that plans may not be reduced to a mere linear sequences of operations and may have more complex structure. For example, a plan may contain conditioned operations to specify the conditions on which an operation shall be performed.

Planning is rooted into the general fact that, for bounded agents, deliberation requires time: if our actions were influenced on the fly short deliberation then performed actions would may not be optimal. Hence, bounded agents need ways to influence action beyond the present.

Plans built by bounded agents cannot be specified in details once and for all. Rather, planning agents tend to create, in a first phase, *partial plans* that may be further refined with respect to the present circumstances of an agent. Such partial plans are constituted of *instrumental desires* (see e.g. [167]) which are at the initiation of further planning in a second phase. On this view, planning consists of means-end reasoning which is aimed to build plan for *how* to satisfy (instrumental) desires, and such planning is called *teleological planning*. It is common to consider another level in planning, in which an agent choses *when* to perform actions adopted in teleological planning. Planning for when to perform actions is *scheduling*. Since teleological planning may be dependent of scheduling, then teleological planning and scheduling are usually interdependent.

As many plan candidates can satisfy a desire, they have to be compared to make a selection and adopt a candidate plan. A common view to compare candidates plans is based on the *expected utility* of plans. The candidate plan having the highest expected utility is then adopted by the agent. In a dynamic world, since plans have to cater for naturally occurring events and actions of other agents, then it is perhaps more relevant to consider strategies instead of plans, as suggested by game theory.

An *intention* is often interpreted as a partial plan that the agent has committed to apply to satisfy a desire. In this view, a common account of intention is the so called desire-belief model. In such model, an action is said intentional or performed with a certain intention if this action is part of a plan that has been built on the basis of some beliefs to reach contents of goals. In the desire-belief model, intention can be hence explained in terms of desires and beliefs.

For example, suppose that Guido intentionally join. This action is said intentional because it is part of the plan built by Guido to get a discount. This plan has been built by Guido because he desires some discounts, and his plan has been built on his belief that joining will provide him some discounts.

This simple belief-desire approach has been criticized by M. Bratman (see e.g. [42]) that argues that intentions should not be characterized by some reduction to beliefs and desires, but instead, accounted for as distinct mental attitudes. His argumentation is based on his consideration that intentions are conduct-controlling (instead of conduct-influencing as desires), proactive attitudes resisting reconsideration, and are subject to inertia. On this basis, M. Bratman considers that the simple belief-desire model does not account for such forms of intentions, and proposes that:

> Prior intentions and plans, then, provide a *background framework* against which the weighing of desire-belief reasons for and against various [deliberative] options takes place. This framework helps focus deliberation: it helps determine which options are relevant and admissible.

Hence, instead of having a linear belief-desire model, we have to consider interactions between deliberative processes and planning-scheduling processes.

It is not our intention here to present and formalize how planning can be performed, and the reader is referred nevertheless for a philosophical discussion to M. Bratman [42] and for more technical solutions to S. Russell and P. Norvig [185].

5.3 Normative cognition

In this Section, we shall focus on normative knowledge, how it can be captured, and how an agent can reason on it. We shall first introduce the three usual building concepts of normative knowledge, i.e. obligation, prohibitions and permissions.

5.3.1 Basic deontic concepts

An obligation indicates something that one has to do. In the following, an obligation is indicated by the deontic operator Obl_{ag} where the subscript ag is a variable over the subject of the obligation. Accordingly, the expression "It is obligatory for ag that γ" is represented by the following formula:

$$\mathrm{Obl}_{ag}\gamma.$$

For example, that "Mario is obliged to cooperate" can be formulated as:

$$\mathrm{Obl}_{\mathrm{mario}}cooperate(\mathrm{mario}).$$

An obligation can be negated. On this aspect, the negation of an obligation and the obligation of the omission of something are distinguished:

- $\neg\text{Obl}_{ag}\gamma$ for the negation of obligation, and
- $\text{Obl}_{ag}\neg\gamma$ for the obligation to omit.

For instance, the negation obligation "Mario has no obligation to attend meetings" can be expressed by:

$$\neg\text{Obl}_{\text{mario}}attend\_meetings(\text{mario}),$$

while the obligation to omit to attend meeting may be formulated by:

$$\text{Obl}_{\text{mario}}\neg attend\_meetings(\text{mario}).$$

The obligation of omission is paralleled by the idea of a prohibition. Being forbidden or prohibited is the status of an action that should not be performed or omitted. In natural language, prohibitions are expressed in various ways. For example we may express the same idea by saying: "It is forbidden that Mario does not cooperate," "There is a prohibition that Mario does not cooperate," etc. In the following, a prohibition is indicated by the deontic operator Forb_{ag}. Accordingly, the expression "it is forbidden for *ag* that γ" can be expressed by the following formula:

$$\text{Forb}_{ag}\gamma.$$

For instance, the expression "Mario is prohibited to attend meeting" can be formulated as:

$$\text{Forb}_{\text{mario}}attend\_meetings(\text{mario}).$$

The third basic deontic concept, besides obligation and prohibition, is permission. In natural language, permissions are expressed in various ways. For example, we may express the same idea by saying: "Mario is permitted to elect the president," "Mario is allowed to elect the president", "Mario can elect the president", etc. A permission shall be indicated by the deontic operator Perm_{ag} and a statement of the type "it is permitted for *ag* that γ" shall be expressed by the following formula:

$$\text{Perm}_{ag}\gamma.$$

For instance, the statement "Mario is permitted to elect the president" can be formulated as:

$$\text{Perm}_{\text{mario}}elect\_president(\text{mario}).$$

The last basic deontic concept is facultativeness which shall be indicated by the deontic operator Fac_{ag}. Accordingly, the expression "it is facultative for *ag* that γ" can be expressed by the following formula:

$$\text{Fac}_{ag}\gamma.$$

For example, the statement "it is facultative for Mario to be a president candidate" can be expressed by:

$$\text{Fac}_{\text{mario}}president\_candidate(\text{mario}).$$

More normative notions can be built by extending the basic deontic concepts as obligations and permissions. For example, G. Sartor in [191, 192] proposes a formalization of rights including obligative rights, permissive rights, ergo-omnes rights and exclusionary rights. Other normative constructs include the notions of legal power (which is distinct from the idea of permission in that legal power provides the ability to determine certain results through one action), enabling power, potestive rights, declarative powers etc.

Though such legal notions are fundamental, we do not address them here, leaving their integration in our framework for future investigations.

5.3.2 Normative conditionals

Following G. Sartor's works [191, 192], a normative conditional expresses that under certain conditions a certain normative conclusion holds. In this view, a normative conditional indicates that the antecedent normatively determines the dependent realization of the consequent, and can have the form:

$$\text{If } antecedents \text{ then}^n \text{ } consequent$$

where the supscript n expresses the idea of normative determination. An ontology of different kinds of normative conditionals is beyond the scope of this work. We assume that interesting norms for our purposes can be reducted to normative conditionals, in the sense that they can express by varying the nature of the antecedents and consequent. In our setting, a normative conditional is represented as a rule of the following form:

$$r: \quad \alpha_1, \ldots, \alpha_n \hookrightarrow \beta \tag{5.57}$$

where $\hookrightarrow$ stands for either $\rightarrow$, $\Rightarrow$ or $\rightsquigarrow$. For example, a provision asserting that members must cooperate can be formulated as a deontic conditional:

$$r: \quad \mathrm{Hold}_{\mathrm{obj}} member(x) \Rightarrow \mathrm{Obl}_x cooperate(x). \tag{5.58}$$

Constitutive rules which express that a thing (state of affairs, event, et.) counts as another thing in a certain context may be viewed as non-deontic conditionals. For example, the constitutive rule, expressing that the action of rising one's hand counts as making a bid in the context of an auction room, can be formulated as:

$$r_{5.59}: \quad \mathrm{Hold}_{\mathrm{obj}} is\_in\_auction\_room(x), \mathrm{Do}_x rise\_hand(x) \Rightarrow \mathrm{Do}_x bid(x). \tag{5.59}$$

As any rule, rules representing normative conditionals can be used in the detachment schema (see Section 5.4.2).

5.3.3 Relations between deontic concepts

Relations between basic deontic concepts can be classified into relations equivalence, compatibility or, at the opposite, incompatibility (or conflict). Each of the four deontic operator X can appear in formulas such as $X\gamma$, $X{\sim}\gamma$, $\neg X\gamma$ or $\neg X{\sim}\gamma$ and instead of explicitly dealing with each of the possible binary relations between formulas, we shall focus on some primitive relations with which the remaining ones can be derived.

We begin with the deontic operator Obl_{ag} and the trivial observations on the incompatibility between the obligation of γ (i.e. $\mathrm{Obl}_{ag}\gamma$) and the obligation of $\sim\gamma$ (i.e. $\mathrm{Obl}_{ag}\sim\gamma$) and between the obligation of γ (i.e. $\mathrm{Obl}_{ag}\gamma$) and the non obligation γ (i.e. $\neg\mathrm{Obl}_{ag}\gamma$). We formalize these incompabilities by writing:

$$\mathrm{Obl}_{ag}\gamma \in \mathscr{C}_{\mathrm{onflict}}(\mathrm{Obl}_{ag}\sim\gamma), \tag{5.60}$$

$$\mathrm{Obl}_{ag}\gamma \in \mathscr{C}_{\mathrm{onflict}}(\neg\mathrm{Obl}_{ag}\gamma). \tag{5.61}$$

As most of the literature, we assume that the obligation of γ is equivalent to the prohibition to $\neg\gamma$. This is expressed by the following where the symbol $\equiv$ represents the relation of equivalence.

$$\mathrm{Obl}_{ag}\gamma \equiv \mathrm{Forb}_{ag}\sim\gamma \tag{5.62}$$

For instance, the statement “Mario must cooperate” is equivalent to “Mario is forbidden to not cooperate”:

$$\mathrm{Obl}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}) \equiv \mathrm{Forb}_{\mathrm{mario}}\neg\mathit{cooperate}(\mathrm{mario}). \tag{5.63}$$

We integrate the equivalence in our framework by two strict rules:

$$r_{\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Forb}_{ag}\sim\gamma}:\quad \mathrm{Obl}_{ag}\gamma \rightarrow \mathrm{Forb}_{ag}\sim\gamma, \tag{5.64}$$

$$r_{\mathrm{Forb}_{ag}\sim\gamma\rightarrow\mathrm{Obl}_{ag}\gamma}:\quad \mathrm{Forb}_{ag}\sim\gamma \rightarrow \mathrm{Obl}_{ag}\gamma. \tag{5.65}$$

A consequence of formulas (5.60) and (5.65) is the intuitive incompatibly of the obligation of γ and the prohibition of γ. For example, the statements “Mario is obliged to cooperate” and “Mario is prohibited to cooperate” are incompatible. We express the conflict between prohibitions and obligations by writing:

$$\mathrm{Obl}_{ag}\gamma \in \mathscr{C}_{\mathrm{onflict}}(\mathrm{Forb}_{ag}\gamma). \tag{5.66}$$

We move to the relation between obligations and permissions. We assume that if it is obligatory that γ then it is permitted that γ. This is expressed by the following schema rule:

$$r_{\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Perm}_{ag}\gamma}:\quad \mathrm{Obl}_{ag}\gamma \Rightarrow \mathrm{Perm}_{ag}\gamma. \tag{5.67}$$

For example, that Mario is obligated to cooperate involves that he is permitted to cooperate:

$$\begin{gathered} r_{\mathrm{Obl}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})\Rightarrow\mathrm{Perm}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})}: \\ \mathrm{Obl}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}) \Rightarrow \mathrm{Perm}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}). \end{gathered} \tag{5.68}$$

Remark that we cannot derive the obligation of γ from the permission to γ. At this stage, we allow us to move to the consideration that the prohibition of γ is equivalent to the non permission of γ:

$$\mathrm{Forb}_{ag}\gamma \equiv \neg\mathrm{Perm}_{ag}\gamma \tag{5.69}$$

Since the obligation of γ ($\mathrm{Obl}_{ag}\gamma$) is equivalent to the prohibition of $\sim\gamma$ (i.e. $\mathrm{Forb}_{ag}{\sim}\gamma$), and the prohibition to γ (i.e. $\mathrm{Forb}_{ag}\gamma$) is equivalent to the non permission of γ (i.e. $\neg\mathrm{Perm}_{ag}\gamma$), then we obtain by transitivity of the relation of equivalence, that the obligation of γ (i.e. $\mathrm{Obl}_{ag}\gamma$) is equivalent to the non permission of $\sim\gamma$ (i.e. $\neg\mathrm{Perm}_{ag}{\sim}\gamma$):

$$\mathrm{Obl}_{ag}\gamma \equiv \neg\mathrm{Perm}_{ag}{\sim}\gamma, \tag{5.70}$$

or equivalently:

$$\neg\mathrm{Obl}_{ag}\gamma \equiv \mathrm{Perm}_{ag}{\sim}\gamma. \tag{5.71}$$

In other words, if no obligation holds, then we have the permission of the contrary. In line with the treatment of equivalence between $\mathrm{Obl}_{ag}\gamma$ and $\mathrm{Forb}_{ag}\gamma$ as strict rules, we integrate in our framework the above equivalences as strict rules. For example, the equivalence is translated as the following schema rules:

$$r_{\mathrm{Forb}_{ag}\gamma\to\neg\mathrm{Perm}_{ag}\gamma}\colon\quad \mathrm{Forb}_{ag}\gamma \to \neg\mathrm{Perm}_{ag}\gamma, \tag{5.72}$$

$$r_{\neg\mathrm{Perm}_{ag}\gamma\to\mathrm{Forb}_{ag}\gamma}\colon\quad \neg\mathrm{Perm}_{ag}\gamma \to \mathrm{Forb}_{ag}\gamma. \tag{5.73}$$

In line with the literature, ‘it is facultative that γ’ is equivalent to say that both γ and $\neg\gamma$’s omission are permitted, and in vice versa. We have thus the following equivalence:

$$\mathrm{Fac}_{ag}\gamma \equiv \mathrm{Perm}_{ag}\gamma\ \mathit{and}\ \mathrm{Perm}_{ag}{\sim}\gamma. \tag{5.74}$$

Since we do not authorize conjunctions in the consequent of rules, we translate the equivalence given above by the three following strict rules:

$$r_{\mathrm{Fac}_{ag}\gamma\to\mathrm{Perm}_{ag}\gamma}\colon\quad \mathrm{Fac}_{ag}\gamma \to \mathrm{Perm}_{ag}\gamma, \tag{5.75}$$

$$r_{\mathrm{Fac}_{ag}\gamma\to\mathrm{Perm}_{ag}{\sim}\gamma}\colon\quad \mathrm{Fac}_{ag}\gamma \to \mathrm{Perm}_{ag}{\sim}\gamma, \tag{5.76}$$

$$r_{\mathrm{Perm}_{ag}\gamma,\,\mathrm{Perm}_{ag}{\sim}\gamma\to\mathrm{Fac}_{ag}\gamma}\colon\quad \mathrm{Perm}_{ag}\gamma,\ \mathrm{Perm}_{ag}{\sim}\gamma \to \mathrm{Fac}_{ag}\gamma. \tag{5.77}$$

For instance, it is facultative for Mario to cooperate involves that Mario is permitted to cooperate and that Mario is not permitted to not cooperate:

$$\begin{gathered} r_{\mathrm{Fac}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})\to\mathrm{Perm}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})}\colon \\ \mathrm{Fac}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}) \to \mathrm{Perm}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}), \end{gathered} \tag{5.78}$$

$$\begin{gathered} r_{\mathrm{Fac}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})\to\mathrm{Perm}_{\mathrm{mario}}\neg\mathit{cooperate}(\mathrm{mario})}\colon \\ \mathrm{Fac}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}) \to \mathrm{Perm}_{\mathrm{mario}}\neg\mathit{cooperate}(\mathrm{mario}). \end{gathered} \tag{5.79}$$

We resume graphically some of the relations between deontic operators given above and derivable relations in Figure 5.1.

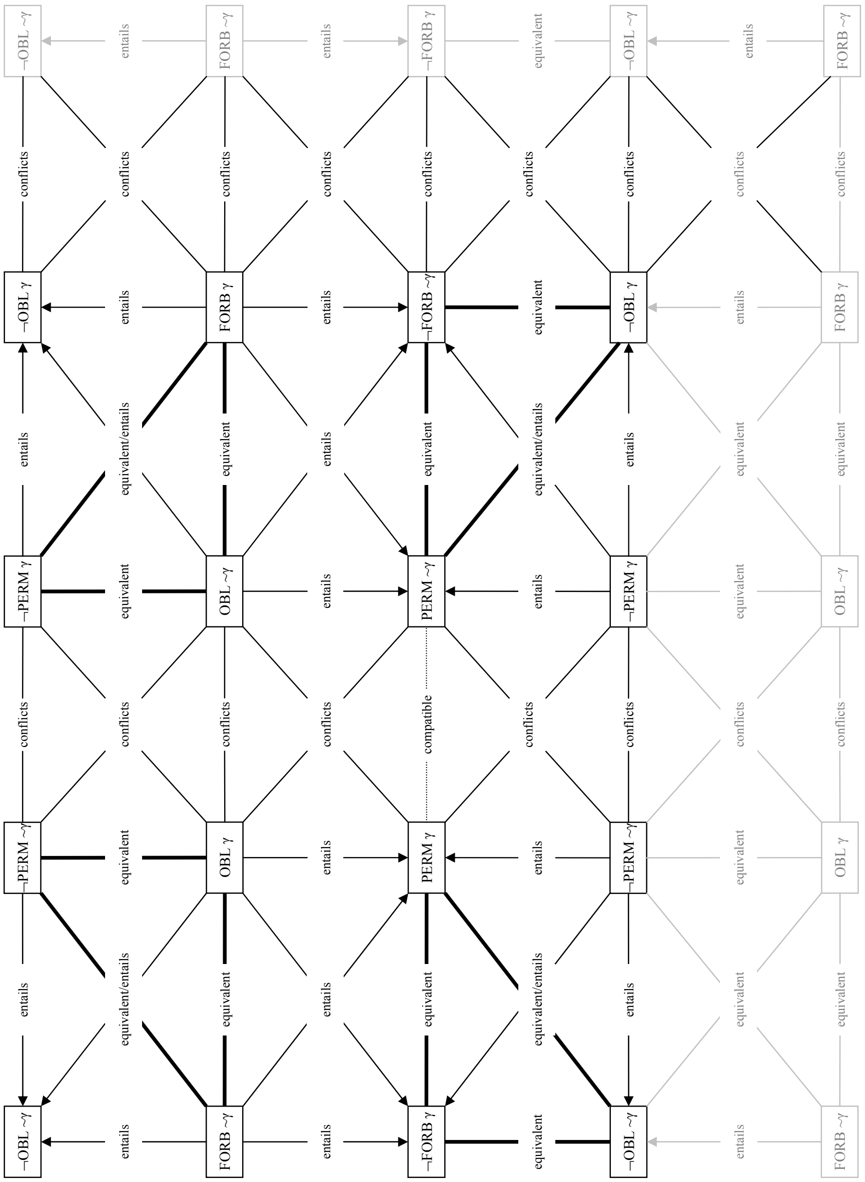


Fig. 5.1. Some relations between deontic operators.

5.3.4 Strong and weak permissions

In [208], p.86, G. von Wright discussed the distinction between *weak permission* (denoted hereafter ${}^{weak}\mathrm{Perm}$) and *strong permission* (denoted Perm)[1]:

> As new kinds of act originate, the authorities of norms may feel a need for considering whether to order or to permit or to prohibit them to subjects. The authority or law-giver may, for example, consider whether the use of alcohol or tobacco should be permitted. In the case of every authority, personal or impersonal, there will always be a great many acts about the normative status of which he never cares.
>
> It is therefore reasonable, given an authority of norms, to divide human acts into two main groups, *viz.* acts which are and acts which are not (not yet) subject to norm, some are permitted, some prohibited, some commanded. Those acts which are not subject to norm are *ipso facto* not forbidden. If an agent does such an act the law-giver cannot accuse him of trespassing against the law. *In that sense* suh an act can be said to be 'permitted'.
>
> If we accept this division of acts into two main groups-relative to a given authority of norms-and if we decide to call acts permitted simply by virtue of the fact that they are not forbidden, then it becomes sensible to distinguish between two kinds of permission. These I shall call *strong* and *weak* permission respectivelly. An act will be said to be permitted in the weak sense if it is not forbidden; and it will be said to be permitted in the strong sense if it is not forbidden but subject to norm. [...] Weak permission is not an independent norm-character. Weak permissions are not prescriptions or norms at all. Strong permission only is a norm-character.

Accordingly, in our setting, given a set of premises A, if the set contains an applicable norm indicating explicitly a permission, we may derive $\mathrm{Perm}_{ag}\gamma$ (i.e. $A \vdash \mathrm{Perm}_{ag}\gamma$). If A does not entail $\mathrm{Forb}_{ag}\gamma$ (i.e. $A \nvdash \mathrm{Forb}\gamma$) then it entails ${}^{weak}\mathrm{Perm}_{ag}\gamma$ (i.e. $A \vdash^{weak}$ $\mathrm{Perm}_{ag}\gamma$).

In this view, weak permission is related to the notion of weak negation ${}^{weak}\neg$ as introduced in Chapter 4. Given a set of premises A, the weak negation of a positive literal γ (i.e. ${}^{weak}\neg\gamma$) is derived if γ is not entailed from A:

$$\frac{A \nvdash \gamma}{A \vdash {}^{weak}\neg\gamma} \tag{5.80}$$

Hence, if we have $A \nvdash \mathrm{Forb}_{ag}\gamma$ then we obtain $A \vdash {}^{weak}\neg\mathrm{Forb}_{ag}\gamma$. Accordingly, as we assumed that $\neg\mathrm{Forb}_{ag}\gamma$ is equivalent to $\mathrm{Perm}_{ag}\gamma$, we have that ${}^{weak}\neg\mathrm{Forb}_{ag}\gamma$ is equivalent to ${}^{weak}\mathrm{Perm}_{ag}\gamma$.

Note that some authors reserve also the expression 'weak permission' to indicate the derivation of a permission from an obligation. This choice seems to me awkward and confusing:

[1] Sometimes, weak and strong permissions are called negative and positive permissions respectively.

- awkward because intuitively, from an obligation we obtain a very strong permission (instead of a weak permission),
- confusing because it tends to make a confusion between permissions obtained from an explicit obligation, and proper weak permissions as intended by G. von Wright: "Weak permissions are not prescriptions or norms at all".

In order to avoid an awkward and confusing terminology, we shall reserve hereafter the expression 'weak permission' w.r.t. γ (i.e. ${}^{weak}\text{Perm}_{ag}\gamma$) to indicate that a set A of premises does not entail a prohibition w.r.t. γ (i.e. $A \nvdash \text{Forb}_{ag}\gamma$). In this view, given an obligation of γ we can derive the strong permission of γ:

$$r_{\text{Obl}_{ag}\gamma \Rightarrow \text{Perm}_{ag}\gamma}\colon \quad \text{Obl}_{ag}\gamma \Rightarrow \text{Perm}_{ag}\gamma. \tag{5.81}$$

Given the obligation of γ we have the strong negation of the permission of $\sim\gamma$ and vice versa:

$$r_{\text{Obl}_{ag}\gamma \Rightarrow \neg\text{Perm}_{ag}\sim\gamma}\colon \quad \text{Obl}_{ag}\gamma \rightarrow \neg\text{Perm}_{ag}\sim\gamma. \tag{5.82}$$

$$r_{\neg\text{Perm}_{ag}\sim\gamma \Rightarrow \text{Obl}_{ag}\gamma}\colon \quad \neg\text{Perm}_{ag}\sim\gamma \rightarrow \text{Obl}_{ag}\gamma. \tag{5.83}$$

Given a set of premises A such that $A \nvdash \text{Forb}_{ag}\gamma$ we have:

$$\frac{A \nvdash \text{Forb}_{ag}\gamma}{A \vdash {}^{weak}\text{Perm}_{ag}\gamma} \tag{5.84}$$

or

$$\frac{A \vdash {}^{weak}\neg\text{Forb}_{ag}\gamma}{A \vdash {}^{weak}\text{Perm}_{ag}\gamma} \tag{5.85}$$

$$\frac{A \nvdash \text{Obl}_{ag}\sim\gamma}{A \vdash {}^{weak}\text{Perm}_{ag}\gamma} \tag{5.86}$$

$$\frac{A \vdash {}^{weak}\neg\text{Obl}_{ag}\sim\gamma}{A \vdash {}^{weak}\text{Perm}_{ag}\gamma} \tag{5.87}$$

A question is whether strong permissions have a proper ontological status. Indeed, a provision indicating a permission seems somewhat useless to guide the behavior of an agent. In this regards, A. Ross argues that a permission is useful only within the context of a contrary obligation, and write in [181], p. 120-122:

> Telling me what I am permitted to do provides no guide to conduct unless the permission is taken as an exception to a norm of obligation (which may be the general maxim that what is not permitted is prohibited). Norms of permission have the normative function only of indicating, within some system, what are the exceptions from the norms of the obligation of the system

> [. . .] I know of no permissive legal rule which is not logically an exemption modifying some prohibition, and interpretable as the negation of an obligation.

In the same line, N. Bobbio identifies the function of permissive norms as exceptions to obligations. In [33], p. 891-892, N. Bobbio writes:

> Permissive norms are subsidiary norms: subsidiary in that their existence presupposes the existence of imperative norms [...] a permissive norms is necessary when we have to repeal a preceding imperative norm or to derogate to it. That is to abolish a part of it (that in this case it is not necessary preexisting because a law itself may prescribe a limit to its own extension.)

A strong permission $\text{Perm}_{ag}\gamma$ may occur as an explicit exception to a 'background' prohibition $\text{Forb}_{ag}{\sim}\gamma$. In that regard, some authors have argued that strong permissions which are non exceptions of prohibitions exist in the setting of hierarchical normative systems (see e.g. E. Bulygin in [48], p.213, and G. Boella and L. van der Torre in [34]) In this view, permissions can be used to allow some authorities to issue norms on some fields, and thus blocking the same authorities to block norms on different fields. These permissions aims at guiding legislators by indicating the fields which can be object of regulation. Here, the difference between such permissions and declarative powers is not clear: do the legislators have the permission to issue norms on some fields? or do the legislators have the declarative power to issue norms on some fields?

Furthermore, G. Boella and L. van der Torre in [34] remark that situations exist in which we have a strong permission for γ, and a weak permission for ${\sim}\gamma$: for example, if it is forbidden to have guns, but it is permitted for policemen to have guns, then a policeman has the strong permission to have guns and the weak permission to not have gun. This brings us to the relation among the concept of facultativeness, strong permission and weak permission. More generally, in which conditions a weak permission can be assimilated to a strong permission?

Such discussion goes beyond our scope, and we shall consider in the remainder only strong permission, leaving the integration of weak permission in our framework for future investigations.

5.4 Reasoning schemata

The process of reasoning proceeds through discrete reasoning steps by instantiating reasoning schemata which are presented in this Section. Before moving to the proper reasoning schemata, we first investigate combinations of modalities.

5.4.1 Combining modalities

In the foregoing, epistemic, practical and normative modalities have been introduced in isolation, but commonsense reasoning present many situations that combine such modalities. For example, Guido desires to believe to be member, Mario believes that Guido does not desire to be a member, Mario desires that Guido brings about being a member, and so on. An infinite number of combination between modalities exists and such combination shall be formulated as a sequence of modalities formulated as follows:

$$(\Phi_i)_{1..n}\gamma \tag{5.88}$$

where γ is a literal, and where Φ_i is modality or the negation of such modality, or in other words, Φ_i is an element of the set $\{\text{Hold}_{ag}$, $\neg\text{Hold}_{ag}$, Des_{ag}, $\neg\text{Des}_{ag}$, Bring_{ag}, $\neg\text{Bring}_{ag}$, Obl_{ag}, $\neg\text{Obl}_{ag}$, Forb_{ag}, $\neg\text{Forb}_{ag}$, Perm_{ag}, $\neg\text{Perm}_{ag}$, Fac_{ag}, $\neg\text{Fac}_{ag}$ $\}$. The variable in subscript *ag* ranges over the set of agents. For example, Guido desires to believe to be member shall be formulated:

$$\text{Des}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{member}(\text{guido}). \tag{5.89}$$

Mario believes that Guido does not desire to be a member as:

$$\text{Hold}_{\text{mario}}\neg\text{Des}_{\text{guido}}\mathit{member}(\text{guido}). \tag{5.90}$$

Mario desires that Guido brings about being a member as:

$$\text{Des}_{\text{mario}}\text{Bring}_{\text{guido}}\mathit{member}(\text{guido}). \tag{5.91}$$

We enrich rules with combination of modalities. A rule can relate modal statements to other modal statements. For example, the conditional statement “If Guido believes to be not entitled of any discount, Guido desires to believe to be entitled of any discount, then Guido desires to bring about being a member” can be expressed by:

$$\begin{aligned} r\colon\quad & \text{Hold}_{\text{guido}}\neg\mathit{discount}(\text{guido}), \text{Des}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{member}(\text{guido}) \\ & \Rightarrow \text{Des}_{\text{guido}}\text{Bring}_{\text{guido}}\mathit{member}(\text{guido}). \end{aligned} \tag{5.92}$$

Rules themselves can be object of modalities, but we limit them to epistemic modalities. Let $(\Phi_i)_{1..n}$ denote a sequence of (negated) epistemic modalities, where Φ_i is stands for $(\neg)\text{Hold}_{ag}$. We admit the possibility for formulas of the following form:

$$(\Phi_i)_{1..n}(r\colon\quad \alpha_1,\ldots,\alpha_n \hookrightarrow \beta), \tag{5.93}$$

where Γ_i and Γ stands for any statement. For example, that the statement “Mario believes that Guido believes that if Guido believes to be not entitled of any discount, Guido desires to believe to be entitled of any discount, then Guido desires to brings about being a member” can be expressed by:

$$\begin{aligned} &\text{Hold}_{\text{mario}}\text{Hold}_{\text{guido}}(r\colon\quad \text{Hold}_{\text{guido}}\neg\mathit{discount}(\text{guido}), \text{Des}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{member}(\text{guido}) \\ &\qquad \Rightarrow \text{Des}_{\text{guido}}\text{Bring}_{\text{guido}}\mathit{member}(\text{guido})). \end{aligned} \tag{5.94}$$

The detachment of the consequent of such rules is investigated next.

5.4.2 Detachment schema

An important schema is the detachment schema which extends the simple detachment presented in Section 4.2 to account for modalities. Suppose a rule believed by an agent *ag*:

$$\mathrm{Hold}_{ag}(r\colon \quad \alpha_1,\ldots,\alpha_n \hookrightarrow \beta).$$

In order to apply such a rule, we assume that the agent *ag* has to believe in $\alpha_1,\ldots,\alpha_n$, that is, we assume that we have $\mathrm{Hold}_{ag}\alpha_1,\ldots,\mathrm{Hold}_{ag}\alpha_n$. For example, given the statements "Guido believes that if he believes to not have any discount, then he desires to get a discount" and "Guido believes that he believes to not have a discount", we derive defeasibly that "Guido believes that he desires a discount". This can be expressed by the following instantiated reasoning schema:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{guido}}\mathrm{Hold}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}),\\ \mathrm{Hold}_{\mathrm{guido}}(r\colon \quad \mathrm{Hold}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}) \Rightarrow \mathrm{Des}_{\mathrm{mario}}\mathit{discount}(\mathrm{guido}))\end{array}}{\mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido})} \tag{5.95}$$

We generalize the schema given above. Let $(\Phi_i)_{1..n}$ denote a sequence of modalities, where Φ_i stands for Hold_{ag}. The sequence can be empty. The variable in subscript *ag* ranges over the set of agents. Let α_i and β any statement. We have the following detachment schema:

$$\frac{\begin{array}{l}(\Phi_i)_{1..n}\alpha_1,\ldots,(\Phi_i)_{1..n}\alpha_n,\\ (\Phi_i)_{1..n}(r\colon \quad \alpha_1,\ldots,\alpha_n \hookrightarrow \beta)\end{array}}{(\Phi_i)_{1..n}\beta} \tag{5.96}$$

where $\hookrightarrow$ stands either for $\rightarrow$, $\Rightarrow$ or $\leadsto$. Accordingly to our previous setting in strict and defeasible reasoning, we can have strict detachments and defeasible ones. A strict detachment schema has a strict rule in its pre-conditions while a defeasible detachment uses a defeasible rule or a defeater rule.

For instance, given that Mario is a member, we can derive defeasibly that Mario is obliged to cooperate:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{obj}}\mathit{member}(\mathrm{mario}),\\ r\colon \quad \mathrm{Hold}_{\mathrm{obj}}\mathit{member}(x) \Rightarrow \mathrm{Obl}_x\mathit{cooperate}(x)\end{array}}{\mathrm{Obl}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})} \tag{5.97}$$

Notice that the detachment schema given in (5.96) allows agents to emulate the reasoning schema of other agents. For example, given the statements "Luigi believes that Mario believes that if Guido believes to not have any discount, then Guido desire to get a discount", and that "Luigi believes that Mario believes that Guido believes to not have a discount", we conclude defeasibly that "Luigi believes that Mario believes that Guido desires a discount". This can be expressed by the following instantiated reasoning schema:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{luigi}}\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}),\\ \mathrm{Hold}_{\mathrm{luigi}}\mathrm{Hold}_{\mathrm{mario}}(r\colon\quad \mathrm{Hold}_{\mathrm{guido}}\neg \mathit{discount}(\mathrm{guido}) \Rightarrow \mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}))\end{array}}{\mathrm{Hold}_{\mathrm{luigi}}\mathrm{Hold}_{\mathrm{mario}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}).} \tag{5.98}$$

Conclusions of instantiated schemata may be used for other schemata in order to build arguments. These arguments may support conflicting conclusions: the analysis of conflicting conclusions is postponed to Section 5.5.

5.4.3 Rule conversion

So-called rule conversion (see e.g. [93]) refers to the inference in which given a rule with modality X, and given that all the literals in the antecedent of the rule are derived in one and the same modality Y, then the conclusion of the rule inherits the modality Y of the antecedent.

For example, given the statements "Mario believes that if one kills another then the latter is died" and "Mario desires to kill Guido", we conclude defeasibly that "Mario desires Guido died":

$$\frac{\begin{array}{l}\mathrm{Des}_{\mathrm{mario}}\mathit{kill}(\mathrm{guido}),\\ \mathrm{Hold}_{\mathrm{mario}}(r\colon\quad \mathit{kill}(x) \Rightarrow \mathit{died}(x))\end{array}}{\mathrm{Des}_{\mathrm{mario}}\mathit{died}(\mathrm{guido})} \tag{5.99}$$

If the notion of rule conversion is appealing to capture many situations, it may be also counter-intuitive in many others. For example, suppose the statements "Mario believes that if one is rich then one pays taxes" and "Mario desires to be rich", then following the inference of rule conversion then we conclude defeasibly that "Mario desires to pay taxes":

$$\frac{\begin{array}{l}\mathrm{Des}_{\mathrm{mario}}\mathit{rich}(\mathrm{mario}),\\ \mathrm{Hold}_{\mathrm{mario}}(r\colon\quad \mathit{rich}(x) \Rightarrow \mathit{tax}(x))\end{array}}{\mathrm{Des}_{\mathrm{mario}}\mathit{tax}(\mathrm{mario})} \tag{5.100}$$

Hence in some cases, rule conversion makes sense whereas in some others it does not. At this stage, two possibilities: either the inference rule conversion is accepted, or it is discarded.

If rule conversion is accepted as an inference then counter-intuitive conclusions can be blocked on light of further grounds. For example, the agent theory can be accompanied with the conditional that one desiring to be rich do not desire to pay taxes.

If the rule of conversion is discarded then, eventually, conditionals which are relevantly object of conversion, can be associated to the corresponding 'converted' conditional. For example, we can add to the first conditional "If one kills another then the latter is died" a second conditional "If one desire to kill another then one desire the latter died", so that given "Mario desires to kill Guido", then we conclude defeasibly by simple detachment that "Mario desires Guido died". Conditionals which are not relevantly object of conversion, are not associated to any 'converted' conditional.

In both cases (i.e. whether rule conversion is accepted or not), some extra conditionals have to be added to the knowledge base to restore intuitions. However, if rule conversion is accepted then a rule of inference has to be added. Hence, it seems that the inference of rule of conversion brings little to the model while complicating it. For this reason, we discard from our model the inference rule conversion.

5.4.4 Introspection and reflexion

In Section 5.1.3 on introspection and reflexion, we proposed some schema rules so that agents are able to reason on their epistemic states. We enlarge here the introspection and reflexion of an agent *ag* to statements of the form $\Phi_{ag}\gamma$ where γ is any statement, and Φ is an element of the set $\{\mathrm{Hold}_{ag}, \neg\mathrm{Hold}_{ag}, \mathrm{Des}_{ag}, \neg\mathrm{Des}_{ag}, \mathrm{Bring}_{ag}, \neg\mathrm{Bring}_{ag}, \mathrm{Obl}_{ag}, \neg\mathrm{Obl}_{ag}, \mathrm{Forb}_{ag}, \neg\mathrm{Forb}_{ag}, \mathrm{Perm}_{ag}, \neg\mathrm{Perm}_{ag}, \mathrm{Fac}_{ag}, \neg\mathrm{Fac}_{ag}\ \}$. Accordingly, we have the following schema rules:

$$r_{\Phi_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\Phi_{ag}\gamma}: \quad \Phi_{ag}\gamma \Rightarrow \mathrm{Hold}_{ag}\Phi_{ag}\gamma, \tag{5.101}$$

$$r_{\mathrm{Hold}_{ag}\Phi_{ag}\gamma\Rightarrow\Phi_{ag}\gamma}: \quad \mathrm{Hold}_{ag}\Phi_{ag}\gamma \Rightarrow \mathrm{Hold}_{ag}\Phi_{ag}\gamma. \tag{5.102}$$

For instance, if Guido desire to get some discount then Guido believes that he desires to get some discount:

$$\begin{split} &r_{\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido})\Rightarrow\mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido})}:\\ &\qquad \mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}) \Rightarrow \mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}). \end{split} \tag{5.103}$$

If Guido believes that he desires to get some discount then Guido desires to get some discount:

$$\begin{split} &r_{\mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido})\Rightarrow\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido})}:\\ &\qquad \mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}) \Rightarrow \mathrm{Des}_{\mathrm{guido}}\mathit{discount}(\mathrm{guido}). \end{split} \tag{5.104}$$

Beside introspection and reflexion, we assume that given any statement, then we can entail defeasibly that this statement holds objectively, and inversely. Accordingly, we have the following rule:

$$r_{\gamma\Rightarrow\mathrm{Hold}_{\mathrm{obj}}\gamma}: \quad \gamma \Rightarrow \mathrm{Hold}_{\mathrm{obj}}\gamma, \tag{5.105}$$

$$r_{\mathrm{Hold}_{\mathrm{obj}}\gamma\Rightarrow\gamma}: \quad \mathrm{Hold}_{\mathrm{obj}}\gamma \Rightarrow \gamma. \tag{5.106}$$

Next, we investigate the collisions of conclusions.

5.5 Collisions of conclusions

In Section 4.3, we saw that an argument S conflicts with an argument R if and only if the arguments S and R contain two conclusions γ_1 and γ_2 such that γ_1 conflicts with γ_2 and vice versa. The set of literals conflicting with a literal γ was denoted $\mathscr{C}_{\text{onflict}}(\gamma)$ and we assumed that a statement γ conflicts with its negation $\neg\gamma$:

- $\mathscr{C}_{\text{onflict}}(\gamma) = \{\neg\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\neg\gamma) = \{\gamma\}$.

In the setting of epistemic, practical and normative reasoning, new conflicts may arise from the collisions of incompatible conclusions. For example, the conclusions “Guido desires a discount” and “Guido does not desire a discount” conflict. In the remainder of this Section, we investigate sets of conflicting statements, and how to handle such conflicts.

5.5.1 Conflicts

In epistemic reasoning, new conflicts may arise from the collisions of incompatible epistemic statements. For example, there is a conflict between the statements “Mario believes to be a member” and “Mario disbelieves to be a member”. The statements “Mario believes to be a member’ and “Mario believes to not be a member” conflict too. More generally, if γ holds from a viewpoint and if γ does not hold from the same viewpoint, then there is a conflict. Likewise, if γ holds from a viewpoint and if $\sim\gamma$[2] holds from the same viewpoint, then there is also a conflict.

- $\mathscr{C}_{\text{onflict}}(\neg\text{Hold}_{ag}\gamma) \supseteq \{\text{Hold}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\text{Hold}_{ag}\gamma) \supseteq \{\text{Hold}_{ag}\sim\gamma\}$.

In terms of belief, a belief of γ and the disbelief of γ conflict, and the belief of γ conflicts with the belief of $\sim\gamma$. For example, we can write $\mathscr{C}_{\text{onflict}}(\neg\text{Hold}_{\text{mario}}\mathit{member}) \supseteq \{\text{Hold}_{\text{mario}}\mathit{member}\}$ and $\mathscr{C}_{\text{onflict}}(\text{Hold}_{\text{mario}}\neg\mathit{member}) \supseteq \{\text{Hold}_{\text{mario}}\mathit{member}\}$. However, we cannot write $\mathscr{C}_{\text{onflict}}(\text{Hold}_{\text{mario}}\neg\mathit{member}) \supseteq \{\neg\text{Hold}_{\text{mario}}\mathit{member}\}$. Notice that conflicts are restricted to believes beared by the same agents. For example, there is no conflicts between the conclusions “Mario believes to be a member” and “Guido does not believe to be member”. Furthermore, at this stage, there is no conflict between $\text{Hold}_{\text{mario}}\mathit{member}$ and $\text{Hold}_{\text{mario}}\text{Hold}_{\text{mario}}\neg\mathit{member}$.

In our practical reasoning setting, we have to cater for potential conflicts between desires, and conflicts between actions. Both types of conflict follow the same schema of conflict between epistemic statements, so a desire (an action) and its negation conflict, and the desire (action) of γ conflicts with the desire (action) of $\sim\gamma$:

- $\mathscr{C}_{\text{onflict}}(\neg\text{Des}_{ag}\gamma) \supseteq \{\text{Des}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\text{Des}_{ag}\gamma) \supseteq \{\text{Des}_{ag}\sim\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\neg\text{Bring}_{ag}\gamma) \supseteq \{\text{Bring}_{ag}\gamma\}$,

[2] As for notation, $\sim\gamma$ denotes any element of the set $\text{Conflict}(\gamma)$.

- $\mathscr{C}_{\text{onflict}}(\text{Bring}_{ag}\gamma) \supseteq \{\text{Bring}_{ag}{\sim}\gamma\}$.

As for epistemic statements, conflicts are restricted to modalities held by the same agents. For example, there is no conflicts between the conclusions “Mario desires to not be a member” and “Guido desires to be a member”.

Concerning normative reasoning, we have already identified primitive conflicts between deontic modalities in Section (5.3.3), we resume all possible conflicts in the following:

- $\mathscr{C}_{\text{onflict}}(\text{Obl}_{ag}\gamma) \supseteq \{\neg\text{Obl}_{ag}\gamma, \text{Obl}_{ag}{\sim}\gamma, \neg\text{Perm}_{ag}\gamma, \text{Perm}_{ag}{\sim}\gamma, \text{Forb}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\neg\text{Obl}_{ag}\gamma) \supseteq \{\text{Obl}_{ag}\gamma, \neg\text{Perm}_{ag}\gamma, \text{Forb}_{ag}{\sim}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\text{Perm}_{ag}\gamma) \supseteq \{\text{Obl}_{ag}{\sim}\gamma, \neg\text{Perm}_{ag}\gamma, \text{Forb}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\neg\text{Perm}_{ag}\gamma) \supseteq \{\text{Obl}_{ag}\gamma, \neg\text{Obl}_{ag}{\sim}\gamma, \text{Perm}_{ag}\gamma, \neg\text{Perm}_{ag}{\sim}\gamma, \text{Forb}_{ag}{\sim}\gamma, \neg\text{Forb}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\text{Forb}_{ag}\gamma) \supseteq \{\text{Obl}_{ag}\gamma, \text{Perm}_{ag}\gamma, \text{Forb}_{ag}\neg\gamma, \neg\text{Forb}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\neg\text{Forb}_{ag}\gamma) \supseteq \{\neg\text{Obl}_{ag}\gamma, \neg\text{Perm}_{ag}\gamma, \text{Forb}_{ag}\gamma, \neg\text{Forb}_{ag}\neg\gamma\}$,
- $\mathscr{C}_{\text{onflict}}(\text{Fac}_{ag}\gamma) \supseteq \{\text{Obl}_{ag}{\sim}\gamma, \neg\text{Perm}_{ag}\gamma, \text{Forb}_{ag}\gamma, \text{Obl}_{ag}\gamma, \neg\text{Perm}_{ag}{\sim}\gamma, \text{Forb}_{ag}{\sim}\gamma\}$.

Remark that our model allows us to cater for combinations of modalities. For example, since we have $\mathscr{C}_{\text{onflict}}(\neg\text{Des}_{\text{guido}}\mathit{member}) \supseteq \{\text{Des}_{\text{guido}}\mathit{member}\}$ then we obtain $\mathscr{C}_{\text{onflict}}(\text{Hold}_{\text{mario}}\neg\text{Des}_{\text{guido}}\mathit{member}) \supseteq \{\text{Hold}_{\text{mario}}\text{Des}_{\text{guido}}\mathit{member}\}$. However, $\text{Hold}_{\text{mario}}\neg\text{Des}_{\text{guido}}\mathit{member}$ and $\neg\text{Des}_{\text{guido}}\mathit{member}$ do not conflict.

Whilst the foregoing can be considered as a minimum set of conflicts that any agent shall adopt, other potential conflicts may be specific for some particular agents. For example, some ‘social’ agents may consider that the desire of γ and the obligation of ${\sim}\gamma$ conflict, whereas other ‘independent’ agents may not. Possible specific conflicts are indicated in the Table 5.1 where each type of conflict is associated with a type of agent (see [204, 46, 45, 59, 93] for related proposals).

Conflict type	Agent type
$\text{Forb}_{ag}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Des}_{ag}\gamma)$	Desire-compliant
$\text{Forb}_{ag}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Bring}_{ag}\gamma)$	Action-compliant
$\text{Des}_{ag}{\sim}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Bring}_{ag}\gamma)$	Slothful
$\text{Hold}_{ag}\text{Hold}_{ag}{\sim}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{ag}\gamma)$	1-introspective consistent
$\text{Hold}_{ag}\neg\text{Hold}_{ag}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{ag}\gamma)$	2-introspective consistent
$\text{Forb}_{ag}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Des}_{ag}\gamma)$	Desire-deviant
$\text{Forb}_{ag}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Bring}_{ag}\gamma)$	Action-deviant
$\text{Des}_{ag}{\sim}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Bring}_{ag}\gamma)$	Unstable
$\text{Hold}_{ag}\text{Hold}_{ag}{\sim}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{ag}\gamma)$	1-introspective inconsistent
$\text{Hold}_{ag}\neg\text{Hold}_{ag}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{ag}\gamma)$	2-introspective inconsistent

Table 5.1. Types of conflict.

In order to distinguish conflicts valid for any agent and conflicts specific to some particular agents, we indicate the latter with a star *. For example, $\text{Forb}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Des}_{\text{mario}}\gamma)$ indicates that, from Mario's viewpoint, any prohibition of γ conflicts with the desire of γ. Though we discard dynamic conflicts for the sake of simplicity, we allow for some simple epistemic reasoning on them:

- Given $Y_{ag}\beta \in \mathscr{C}^*_{\text{onflict}}(X_{ag}\gamma)$, we have $\text{Hold}_{\text{obj}}Y_{ag}\beta \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{\text{obj}}X_{ag}\gamma)$, and vice versa,
- Given $Y_{ag}\beta \in \mathscr{C}^*_{\text{onflict}}(X_{ag}\gamma)$, we have $\text{Hold}_{ag}Y_{ag}\beta \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{ag}X_{ag}\gamma)$, and vice versa,
- Given $\beta' \in \mathscr{C}^*_{\text{onflict}}(\gamma)$ and $\beta \equiv \beta'$, we have $\beta \in \mathscr{C}^*_{\text{onflict}}(\gamma)$.

For example, given the specific conflict regarding Mario $\text{Forb}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Des}_{\text{mario}}\gamma)$, we can derive among others $\text{Hold}_{\text{obj}}\text{Forb}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{\text{obj}}\text{Des}_{\text{mario}}\gamma)$ and $\text{Hold}_{\text{mario}}\text{Forb}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}_{\text{mario}}\text{Des}_{\text{mario}}\gamma)$.

Desire-deviant and action-deviant agents tend to ignore norms and thus tend to violate norms because they do not consider any conflict between the desire or the performance of γ and its prohibition. In this view, a distinction is made between logical conflicts (possibly leading to some inconsistencies) and violations. For example, an action-deviant agent *ag* can bring about γ (i.e. $\text{Bring}_{ag}\gamma$) while a normative statement forbids it (i.e. $\text{Forb}_{ag}\gamma$): the scenario does not lead to a logical conflict but a violation.

5.5.2 Preferences

Desire-compliant and action-compliant agents may also violate norms: whereas desire-deviant and action-deviant agents tend to violate norms because no conflicts exist between the desire or the performance of γ, and its prohibition, desire-compliant and action-compliant agents may violate norms because they have a preference of their desires of actions over norms. Indeed, conflicts are not defeats: in Section 4.3, we saw that conflicts between defeasible conclusions can be resolved on the basis of preferences expressed by an explicit strength order between rules: rules are identified by labels and a strength order is stabilized by the operator $\succ$ between two rules labels in order to indicate the relative strength of each rule. So, the formula $r_2 \succ r_1$ indicates that the rule labeled r_2 is stronger than the rule labeled by r_1. Provided two applicable rules and a strength order between these two rules, one is allowed to conclude only for the consequent of the stronger rule. In the remainder, we discuss 'local' preferences or 'global' preferences to specify strength order.

Local preferences consist in specifying as usual strength orders between conflicting rules. For instance, suppose the following rules:

$$\text{r}_1: \quad \Rightarrow \text{Des}_{\text{mario}}\neg cooperate(\text{mario}), \tag{5.107}$$

$$\mathrm{r}_2: \qquad \Rightarrow \mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario}). \tag{5.108}$$

Assuming that Mario is a desire-compliant agent, if the rule r_2 is stronger than the rule r_1 (i.e. $\mathrm{r}_2 \succ \mathrm{r}_1$) then the conclusion that Mario is obliged to cooperate is entailed whereas the conclusion that his desire to do not so is discarded. This solution is local because its scope is limited to the rules appearing in the preference relation $\succ$. Of course, preferences over rules hold for some viewpoints. To account for it, a preference will have the form $(X_i)_{1..n}(r_2 \succ r_1)$ where X_i is an epistemic operator Hold_{ag}. For instance, “It holds, from the viewpoint of Guido, that r_2 is stronger than r_1” is formulated as:

$$\mathrm{Hold}_{\mathrm{guido}}(\mathrm{r}_2 \succ \mathrm{r}_1).$$

Accordingly, a conflict between conclusions held from some viewpoints have to be resolved by some preferences held from the same viewpoints. For instance, given the rules r_1 and r_2 and the preference $\mathrm{Hold}_{\mathrm{guido}}(\mathrm{r}_2 \succ \mathrm{r}_1)$, we cannot solve the conflict because the conflicting conclusions and the preference do not hold from the same viewpoint. However, given the same rules and the preference $\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_2 \succ \mathrm{r}_1)$, we can solve the conflict and conclude that Mario is obliged to cooperate (i.e. $\mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})$). Let’s see another example. Given the following rules:

$$\mathrm{Hold}_{\mathrm{guido}}(\mathrm{r}_3: \qquad \Rightarrow \mathrm{Des}_{\mathrm{mario}} \neg \mathit{cooperate}(\mathrm{mario})),$$

$$\mathrm{Hold}_{\mathrm{guido}}(\mathrm{r}_4: \qquad \Rightarrow \mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})),$$

and the preference $\mathrm{Hold}_{\mathrm{guido}}\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_4 \succ \mathrm{r}_3)$, we can resolve the conflict between $\mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{mario}} \neg \mathit{cooperate}(\mathrm{mario})$ and $\mathrm{Hold}_{\mathrm{guido}}\mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})$, to conclude $\mathrm{Hold}_{\mathrm{guido}}\mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})$.

Though we discard dynamic preferences for the sake of simplicity, we allow for some simple epistemic reasoning on them:

- Given $(X_i)_{1..n}(r_2 \succ r_1)$, we have $\mathrm{Hold}_{\mathrm{obj}}(X_i)_{1..n}(r_2 \succ r_1)$, and vice versa,
- Given $\mathrm{Hold}_{ag}(X_i)_{1..n}(r_2 \succ r_1)$, we have $\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}(X_i)_{1..n}(r_2 \succ r_1)$, and vice versa.

The first item indicates that any preference holds objectively, and vice versa. The second item expresses introspection and reflexion over preferences. For example, given the rules r_1 and r_2 in (5.107) and (5.107), we can derive the following:

$$\mathrm{Hold}_{\mathrm{obj}}(\mathrm{r}_1: \qquad \Rightarrow \mathrm{Des}_{\mathrm{mario}} \neg \mathit{cooperate}(\mathrm{mario})), \tag{5.109}$$

$$\mathrm{Hold}_{\mathrm{obj}}(\mathrm{r}_2: \qquad \Rightarrow \mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})). \tag{5.110}$$

Likewise, given $\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_2 \succ \mathrm{r}_1)$, we have $\mathrm{Hold}_{\mathrm{obj}}\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_2 \succ \mathrm{r}_1)$. The conflict between $\mathrm{Hold}_{\mathrm{obj}}\mathrm{Des}_{\mathrm{mario}} \neg \mathit{cooperate}(\mathrm{mario})$ and $\mathrm{Hold}_{\mathrm{obj}}\mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})$ is thus resolved and we derive $\mathrm{Hold}_{\mathrm{obj}}\mathrm{Obl}_{\mathrm{mario}} \mathit{cooperate}(\mathrm{mario})$.

The assumption that a conflict between conclusions held from some viewpoints have to be resolved by some preferences held from the same viewpoints raises some issues when there is an 'epistemic asymmetry of viewpoints' between two conclusions.

For example, suppose that Mario is an agent who is 1-introspective consistent, i.e. $\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}{\sim}\gamma \in \mathscr{C}_{\mathrm{onflict}}(\mathrm{Hold}_{\mathrm{mario}}\gamma)$ (see Table 5.1) and suppose the following rules:

$$\mathrm{r}_5: \qquad \Rightarrow \mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{c}, \tag{5.111}$$

$$\mathrm{r}_6: \qquad \Rightarrow \mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathrm{c}. \tag{5.112}$$

Suppose that the rule r_5 is stronger than the rule r_6. In order to resolve the conflict between $\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{c}$ and $\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathrm{c}$, do we need to consider the preference $\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_5 \succ \mathrm{r}_6)$ or the preference $\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_5 \succ \mathrm{r}_6)$? Fortunately, if we have the first preference then we have the second, and vice versa. In other words, both are acceptable to resolve the conflict, and both provide us with the same outcome.

In the general case, given two conflicting conclusions having an asymmetry of viewpoints, we assume that the preference has to hold from the viewpoints of the undefeated conclusion.

Another solution to resolve conflict is to provide a global preference between conflicting statements.

For example, the profile of Mario can be described by asserting that, by default, the prohibition of γ defeats the desire of γ so that the conclusion that Mario desires to not cooperate is defeated by its obligation to cooperate (i.e. the prohibition to not cooperate). We shall write $\beta \in \mathscr{D}_{\mathrm{efeat}}(\gamma)$ to denote a statement β defeats another statement γ. For instance, in the case of Mario, we can write $\mathrm{Forb}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}_{\mathrm{mario}}\gamma)$ to indicate that, by default, prohibitions defeat desires.

Types of global preferences over statements can be associated to some types of agent. For example, if prohibitions defeat desires w.r.t. an agent then we say that this agent is desire-social. Table 5.2 parses possible types of defeats between statements, and associated types of agent.

Defeat type	Agent type
$\mathrm{Forb}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}_{ag}\gamma)$	Desire-social
$\mathrm{Des}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Forb}_{ag}\gamma)$	Desire-unsocial
$\mathrm{Forb}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Bring}_{ag}\gamma)$	Action-social
$\mathrm{Bring}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Forb}_{ag}\gamma)$	Action-unsocial
$\mathrm{Des}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Bring}_{ag}{\sim}\gamma)$	Willful
$\mathrm{Bring}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}_{ag}{\sim}\gamma)$	Involuntary

Table 5.2. Types of defeat.

Of course, based on different frameworks, other agent types can be specified (see e.g. [204, 46, 45, 59, 93]). For example, it is sometimes argued that beliefs should override desires. In [204] for instance, R. Thomason considers that (i) if you believe that it is going to rain, and that (ii) if it rains, you will get wet, and (iii) you would not like to get wet, then you should believe you will get wet. On the assumption that the desire and the belief conflict, then you should believe that you will get wet. It is thus argued that beliefs defeat conflicting desires in order to avoid *wishful thinking* agents. Though the present framework can easily capture the idea by adding $\mathrm{Hold}_{ag}\gamma \in \mathrm{Defeat}(\mathrm{Des}_{ag}{\sim}\gamma)$, we do not feel any necessity to do so. Indeed, we do not assume any conflict between believing something and desiring the contrary. For example, you can be wet as a matter of fact, still you would not like to be wet. Assuming a conflict between a belief and the desire of the contrary, and the defeat of desires by believes would also imply that an agent cannot change his environment. In the example, you will never be able to make a plan to avoid being wet, for instance by taking an umbrella.

Likewise conflicts and local preferences, we discard dynamic global preferences for the sake of simplicity but we allow for some epistemic reasoning on them:

- Given $X_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(X_{ag}\gamma)$, we have $\mathrm{Hold}_{\mathrm{obj}}X_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{\mathrm{obj}}X_{ag}\gamma)$, and vice versa,
- Given $X_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(X_{ag}\gamma)$, we have $\mathrm{Hold}_{ag}X_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{ag}X_{ag}\gamma)$, and vice versa,
- Given $\beta' \in \mathscr{D}_{\mathrm{efeat}}(\gamma)$ and $\beta \equiv \beta'$, we have $\mathrm{Hold}_{ag}\beta \in \mathscr{D}_{\mathrm{efeat}}(\gamma)$.

For example, given $\mathrm{Forb}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}_{\mathrm{mario}}\gamma)$, we can derive among others $\mathrm{Hold}_{\mathrm{obj}}\mathrm{Forb}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{\mathrm{obj}}\mathrm{Des}_{\mathrm{mario}}\gamma)$ and $\mathrm{Hold}_{\mathrm{mario}}\mathrm{Forb}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{\mathrm{mario}}\mathrm{Des}_{\mathrm{mario}}\gamma)$.

Local preferences and global preferences may collide. For example, suppose the following rules

$$\mathrm{Hold}_{\mathrm{obj}}(\mathrm{r}_7\!: \quad \Rightarrow \mathrm{Des}_{\mathrm{luigi}}\neg \mathit{cooperate}(\mathrm{luigi})), \tag{5.113}$$

$$\mathrm{Hold}_{\mathrm{obj}}(\mathrm{r}_8\!: \quad \Rightarrow \mathrm{Obl}_{\mathrm{luigi}}\mathit{cooperate}(\mathrm{luigi})). \tag{5.114}$$

Suppose that we have the conflict $\mathscr{C}_{\mathrm{onflict}}(\mathrm{Des}_{\mathrm{luigi}}\neg \mathit{cooperate}(\mathrm{luigi})) \supseteq \{\mathrm{Obl}_{\mathrm{luigi}}\mathit{cooperate}(\mathrm{luigi})\}$, the local preference $\mathrm{Hold}_{\mathrm{mario}}(\mathrm{r}_7 \succ \mathrm{r}_8)$ and the global preference $\mathrm{Obl}_{\mathrm{luigi}}\gamma \in \mathrm{Defeat}(\mathrm{Des}_{\mathrm{luigi}}{\sim}\gamma)$. According the local preference, we are pushed to conclude $\mathrm{Des}_{\mathrm{luigi}}\neg \mathit{cooperate}(\mathrm{luigi})$ whereas according the global preference, we are pushed to conclude $\mathrm{Obl}_{\mathrm{luigi}}\mathit{cooperate}(\mathrm{luigi})$. In order to cope with this issue, we assume that a global preference hold by default in the sense that if a local preference is in contradiction with the later then the local preference take precedence over the global preference. Thus, in our example, we conclude $\mathrm{Des}_{\mathrm{luigi}}\neg \mathit{cooperate}(\mathrm{luigi})$ and discard $\mathrm{Obl}_{\mathrm{luigi}}\mathit{cooperate}(\mathrm{luigi})$.

In this Chapter, we have seen that profiles of agents deliberation and thus normative bindingness can be characterized by stating conflicts between statements and preferences to solve these conflicts. In the next Chapter, we shall accommodate this framework to integrate some temporal aspects.

6

Temporal model

Dynamic environments are an essential motivation for the need of autonomous agents. In a dynamic environment, things may not pass at the same instant, and time allows us to express things which pass at different instants.

In the following, we shall extend the model presented in Chapter 5 to give an account of some temporal aspects with regard to both mental attitudes and deontic provisions. We shall try to abstract from some philosophical issues concerning the nature of time and how it can be accommodated within some metaphysical models. We shall just assume that mental attitudes and normative provisions can be related to temporal references and the passage of time allows change of these elements.

In Section 6.1, we investigate the integration of temporal dimensions to statements. Reasoning schemata with respect to the temporal setting are discussed in Section 6.2. Conflicts between temporal statements are addressed in Section 6.3 and modifications in Section 6.4.

6.1 Temporal statements

Ordinarily, two types of time reference are considered in the literature. One type is absolute time reference (for example a date as 28 February 2007) while the other type is relative time reference (for example by the use of the term “before”, “after” etc.). Roughly, a relative time reference is called so because it stands in relation to certain conditions or circumstances while an absolute time reference is such that it does not stand in relation to any conditions or circumstances. However, this distinction is somewhat dubious since an absolute time reference is still relative to another absolute reference (for example the birth of Jesus for western datation). Hence, the difference of between the two qualifications of time references are standardly captured as follows:

- An absolute time reference is defined in relation to a global event (origin of time) of the system.
- A relative time reference is defined in relation to a local event of the system.

In our framework, the passage of time allows for the change of things (as mental attitudes and provisions), and these things are expressed by some statements related to temporal references.

6.1.1 Temporal modal literals

In the previous Chapter, epistemic, practical and normative reasoning was captured by modal operators. In this setting, temporal references can be associated to the modalities. For example, Mario believes in 2007 that he is a student. Guido desires in 2007 to have discount. Guido is obliged in 2007 to reduce pollution. To temporalise modalities, we supscript them with an absolute time reference. For instance, the sentence "Mario believes in 2008 to be a student" can be formulated as:

$$\mathrm{Hold}^{2008}_{\mathrm{mario}} student(\mathrm{mario}). \tag{6.1}$$

Intuitively, $\mathrm{Hold}^{t}_{\mathrm{ag}}\gamma$ means that γ holds at a time t for the point of view of an agent ag. To indicate that γ holds objectively at time t, we write $\mathrm{Hold}^{t}_{\mathrm{obj}}\gamma$. For instance, the statement "It holds in 2008, that Mario is a student" can be formulated as:

$$\mathrm{Hold}^{2008}_{\mathrm{obj}} student(\mathrm{mario}). \tag{6.2}$$

We admit sequences of modalities. For instance, the sentence "Mario believes in 2009 that he was student in 2008" shall be formulated as:

$$\mathrm{Hold}^{2009}_{\mathrm{mario}}\mathrm{Hold}^{2008}_{\mathrm{obj}} student(\mathrm{mario}). \tag{6.3}$$

This last formulation may appear clumsy because we may assume that an agent cannot believe objectively something: a belief remains always subjective to an agent. On this aspect, it is perhaps more appropriate to reformulate (6.3) into:

$$\mathrm{Hold}^{2009}_{\mathrm{mario}}\mathrm{Hold}^{2008}_{\mathrm{mario}} student(\mathrm{mario}). \tag{6.4}$$

Notice also that, according our interpretation of formulas $\mathrm{Hold}^{t}_{ag}\gamma$ as "It holds, from the viewpoint of agent ag at time t, that γ", sequences of operators of the form $\mathrm{Hold}^{t'}_{ag}\mathrm{Hold}^{t}_{ag}$ are also able to express a belief at certain time t' about something holding at a different time t and which is not believed at this time t. For example, if a retroactive rule r is published in 2008 and is efficacious in 2007, then Mario could not believe in 2007 that the rule r was efficacious. Hence, whereas it does not make sense to state that Mario believes in 2008 that Mario believes in 2007 that the rule is efficacious, it does make sense to say that "It holds, from the point of view of Mario and in 2008, that it holds, from the viewpoint of Mario and in 2007, that the rule is efficacious":

$$\mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}} efficacious(r). \tag{6.5}$$

Other operators can also be associated to a temporal reference. For example, the statement "Guido desires in 2007 to have discount" shall be formulated as:

$$\mathrm{Des}^{2007}_{\mathrm{guido}} discount(\mathrm{guido}). \tag{6.6}$$

The productive action “In 2006, Guido brings about his membership” can be expressed by:

$$\mathrm{Bring}^{2006}_{\mathrm{guido}} member(\mathrm{guido}). \tag{6.7}$$

Deontic modalities can be temporalised too. For instance, the provision “Guido is obliged in 2009 to pay tax” can be formulated as:

$$\mathrm{Obl}^{2009}_{\mathrm{guido}} pay\_tax(\mathrm{guido}). \tag{6.8}$$

We may use the universal quantifier and some constraints over time. For instance, the statement “Mario is an adult in 1973 onwards” shall be formulated as:

$$\forall t, t \geq 1973, \quad \mathrm{Hold}^{t}_{ag} adult(\mathrm{mario}). \tag{6.9}$$

We leave the the use of existential quantifier over time as a matter of future investigation. Quantification over time can have for scope a modality and the content of the modality. For example, the statement “Mario desires this week to have a discount next week” shall be formulated as:

$$\forall t, 1 \geq t \geq 7, \forall t', 8 \geq t' \geq 14, \quad \mathrm{Des}^{t}_{\mathrm{mario}} \mathrm{Hold}^{t'}_{ag} discount(\mathrm{mario}). \tag{6.10}$$

In order to simplify the writing of bounded quantification over time, we shall use time intervals. A time interval denoted $[t_i, t_f]$ is a set of instants t bounded by t_i and t_f such that for all t, $t \geq t_i$ and $t \leq t_f$. Accordingly, the formula (6.10) shall be rewritten in the interval based notation as:

$$\mathrm{Des}^{[1,7]}_{\mathrm{mario}} \mathrm{Hold}^{[8,14]}_{\mathrm{mario}} discount(\mathrm{mario}). \tag{6.11}$$

When we are interested in figuring out in which temporal intervals a conclusion holds, it is perhaps more perspicuous to represent it using a graph where each axis represents the time arrow in a different temporal dimension. For example, $\mathrm{Hold}^{[0,50]}_{\mathrm{mario}}\mathrm{Hold}^{[0,50]}_{\mathrm{mario}}\mathrm{c}$ can be represented graphically in Figure 6.1.

6.1.2 Temporal modal rules

A temporal rule relates some temporal statements $\alpha_1 \ldots \alpha_n$ to another temporal statement β and has the following form:

$$r: \quad \alpha_1, \ldots, \alpha_n \hookrightarrow \beta. \tag{6.12}$$

where $\hookrightarrow$ stands for $\rightarrow$, $\Rightarrow$ and $\leadsto$. For example, the provision stating that “members are obliged to cooperate” can be formulated as:

$$r: \quad \mathrm{Hold}^{t}_{ag} member(x) \Rightarrow \mathrm{Obl}^{t}_{ag} cooperate(x). \tag{6.13}$$

Our formalization allows us to express rules whose consequent outlives the antecedent, and such rules shall be called P-rules. For example, in legal setting as in

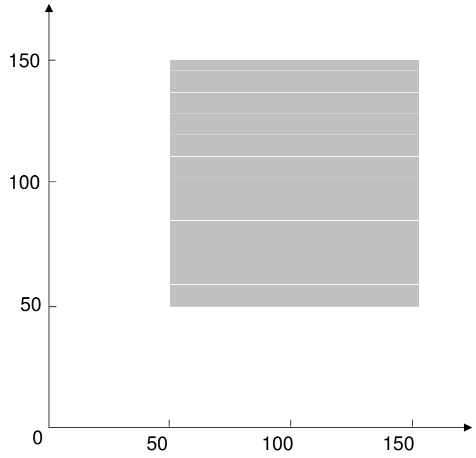


Fig. 6.1. Graphical representation of $\mathrm{Hold}^{[0,50]}_{\mathrm{mario}}\mathrm{Hold}^{[0,50]}_{\mathrm{mario}}\mathrm{c}$.

the Italian normative system, the following P-rule indicates that the date of entry of force of a provision is established after a 15-day period (called *vacatio legis*) starting at the date of publication:

$$vl\colon \quad \mathrm{Hold}^{t}_{ag} published(r) \Rightarrow \mathrm{Hold}^{[t+15,max]}_{ag} force(r). \tag{6.14}$$

It is important to not assimilate P-rules to some persistence of some sort. For example, a rule of the form:

$$r\colon \quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{[t,max]}_{ag}\beta$$

is a notational short-cut for the rules:

$$r\colon \quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{t}_{ag}\beta,$$

$$r\colon \quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{t+1}_{ag}\beta,$$

$$r\colon \quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{t+2}_{ag}\beta,$$

$$\dots$$

$$r\colon \quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{max}_{ag}\beta,$$

We assume that the persistence of γ is the self-renewal of γ through time. Based on this assumption, persistence can be accurately formalized, for example, as the following schema rule:

$$r_{6.15}\colon \qquad \mathrm{Hold}^{t}_{ag}\gamma \Rightarrow \mathrm{Hold}^{t+1}_{ag}\gamma. \tag{6.15}$$

Such rules shall be called rules of persistence. For instance, instead of capturing the statement “If one is born then one is alive onwards.” by the following P-rule:

$$r\colon \quad \mathrm{Hold}^{t}_{ag} born(x) \Rightarrow \mathrm{Hold}^{[t,max]}_{ag} alive(x), \tag{6.16}$$

we shall capture the persistence of aliveness by the following rule of persistence:

$$r_{6.17}\colon \qquad \mathrm{Hold}^{t}_{ag} alive(x) \Rightarrow \mathrm{Hold}^{t+1}_{ag} alive(x). \tag{6.17}$$

As in most cases, the persistence of a thing is initiated by other things, then a rule of persistence shall be accompanied with a triggering rule. For instance, the statement “If one is born then one is alive.” could captured by the following rule:

$$r\colon \quad \mathrm{Hold}^{t}_{ag} born(x) \Rightarrow \mathrm{Hold}^{t}_{ag} alive(x). \tag{6.18}$$

The distinction between P-rules and persistence rules allows us to express appropriately maintenance and achievement statements, for example between *achievement* obligations and *maintenance* obligations (cf. [89]).

An example of achievement obligation is a provision indicating that customers must pay within 30 days after receiving the invoice. For an achievement obligation, the state of affairs ϕ has to occur before a deadline condition δ, otherwise a violation is detected. An achievement obligation can be formalized by the following template rules:

$$\begin{array}{lll}
r_{init}\colon & \alpha_1, ..., \alpha_n \Rightarrow \mathrm{Obl}^{t}_{ag}\phi, & \text{(initialization)} \\
r_{pers}\colon & \mathrm{Obl}^{t}_{ag}\phi \Rightarrow \mathrm{Obl}^{t+1}_{ag}\phi, & \text{(persistence)} \\
r_{term}\colon & \mathrm{Obl}^{t}_{ag}\phi, \mathrm{Hold}^{t}_{ag}\phi \rightsquigarrow \neg\mathrm{Obl}^{t}_{ag}\phi, & \text{(termination)} \\
r_{viol}\colon & \mathrm{Hold}^{t_\delta}_{ag}\delta, \mathrm{Obl}^{t_\delta}_{ag}\phi \Rightarrow \mathrm{Hold}^{t_\delta}_{ag}\mathrm{viol}(r). & \text{(violation)}
\end{array}$$

The violation fact $\mathrm{viol}(r)$ is a specific literal, indexed by the name of the group of rules. Generally, a deadline signals that a violation of the obligation has occurred (rule inv_{viol}). This may or may not trigger an explicit sanction (see below). Note that the obligation itself may even persist after the deadline.

In maintenance obligations, the state of affair has to hold for any instants. For example, customers must keep a positive balance, for 30 days after opening an bank account. Prohibitions, i.e. obligations to avoid some states, form a large class of maintenance obligations. The generic formalization for maintenance obligations consists of the following template rules.

$$\begin{array}{lll}
r_{init\_main}\colon & \mathrm{Hold}^{t}_{ag}\chi, \mathrm{Hold}^{t_\delta}_{ag}\delta \Rightarrow \mathrm{Obl}^{[t,t_\delta]}_{ag}\phi, & \text{(initialization maintenance)} \\
r_{viol\_main}\colon & \mathrm{Obl}^{t}_{ag}\phi, \mathrm{Hold}^{t}_{ag}\neg\phi \Rightarrow \mathrm{Hold}^{t}_{ag}\mathrm{viol}(r). & \text{(violation of maintenance)}
\end{array}$$

Note that neither termination rule nor persistence rules are needed. Here, the deadline only signals that the obligation is terminated. A violation occurs, when the obliged state does not obtain at some point before the deadline.

For an achievement obligation, the condition ϕ must occur at least once before the deadline. For maintenance obligations, condition ϕ must obtain during all instants before the deadline. Likewise maintenance and achievement obligations, P-rules and persistence rules allow us to express maintenance goals to maintain some particular state and achievement goals to attain (achieve) some state.

6.1.3 Legal temporal dimensions

At this stage, it is important to notice that legal reasoning is characterized by various temporal dimensions (see [156, 157]): if we intend to give an accurate account of temporal aspects of norms and therefore to be consistent with legal principles, then we have to identify, distinguish and capture these various times. These distinctions are, for example, of the utmost importance to express ultra-active and retroactive provisions as we shall see soon.

Our analysis shall be centered upon the distinction of two temporal profiles in legal norms. A first distinction deals with so-called *external times* and *internal times* of norms.

- The internal times are associated to a norm or provision which is specified within the norm or provision.
- The external times are associated to a norm or provision which is specified in a different norm or provision.

A second distinction deals with so-called *static times* and *dynamic times* of norms.

- The static times are associated to a norm or provision which cannot change on the basis of events,
- The dynamic times are associated to a norm or provision which can change on the basis of future events (typically as normative modifications).

In general, static times are defined at the document level, and the embedded provisions inherit the same static times of its containing document. Commonly cited static times are the *date of existence*, the *date of enactment*, the *date of publication*, the *date of entry of force*:

- The date of existence of a document is the date when the lawmaking body (such as a senate or a lower house) ‘freezes’ the document in its final form.
- The date of enactment, or date of delivery, or date of assent, succeeds the date of existence and is the date when the competent authorities finalize the document by affixing their signatures to it (e.g. promulgation by a president, signature by a king or queen). In general, this date is clearly indicated in the document.
- The date of publication is the date when the normative document is published in an official journal.
- The date of entry into force of a document marks the beginning of its period of force. In general, the date of entry of force is established on function of the date of document’s publication in an official journal. For example, in the Italian normative system, the date of entry of force is establish after a 15-day period (called *vacatio legis*) starting at the date of publication.

Common important external and dynamic dates are the *date of republication* and the *date of transposition*:

- the date of republication is the date when the document is republished into the official journal. Though the date of publication is unique, a document can be republished several times, and thus can have several republication dates.

- the date of transposition is the date when member states brings into force the laws necessary to comply with a directive.

We concentrate on three external and dynamic times associated to norms, namely the *time of force*, the *time of efficacy*, and the *time of applicability*:

- The time of force is the time during which the legal norm is in force or valid if by validity of a norm we mean its partaking to the normative system. Generally, it is a period of time which boundaries may change over time through temporal modifications. By default, the beginning of the period of force is determined by other rules, and the end of the period of force is established by derogation rules.
- The time of efficacy, sometimes called the time of enforceability, is the time during which the provision *causes* its legal effect. In this regard, we have to make a distinction between unconditioned provisions and unconditioned provisions. For a unconditioned provision, the time of efficacy corresponds to the time when the provision is operative. For instance, if “It is forbidden to smoke” is efficacious during $[t_1, t_2]$ then during that period smoking is prohibited. For conditioned provisions, the time of efficacy corresponds to the time during which the provision’s conditions can be instantiated. For instance, “If one smokes, then one is subject to 10 Euro fine” is efficacious during $[t_1, t_2]$ then during that time the fact that one person smokes causes that persons subjection to the sanction.
- The time of applicability, or the time of the effect, is the time at which the intented effect of the provision is applied. For unconditioned provisions, it is by default the time when the provision becomes operative if the provision has an instantaneous effect, or the period during which the provision remains operative. For conditioned provisions, it is a function of the time when an instance of the provision’s conditions have taken place. By default it is contemporary but a different time can also be established. For instance, “If you drive dangerously, your license will be suspended for one year”. In this case, if one drives dangerously at time t, then the license will be suspended from t to $t+1$ year. Then $[t, t+1\text{year}]$ is the time of the effect related to the preconditions instance of time t. If the provision is instantiated to a concrete case, then some temporal values are assigned to the time of applicability: the time values assigned to the time of applicability w.r.t. to a concrete case is called the *time of occurrence* of a provision w.r.t. to a concrete case. For example, if Mario drives dangerously in 2007, then his license will be suspended from 2007 to 2008: the time of occurrence is the period made of the year 2007 and 2008.

Notice that for unconditioned provisions, the time of applicability is the time of efficacy. In general, the provision’s period of efficacy and applicability coincide with its period of force, but in some cases they are different as for example in case of conditioned retroactive or ultra-active norms. An example of a provision illustrating the different temporal dimensions is provided in Table 6.1.

In a such complex temporal setting, as pointed out by [133], we can use either an analytical approach or a synthetical approach for the representation of provisions. In the synthetical approach all temporal and substantial elements of the norm are represented within the same sentence. For example:

Table 6.1.

Art. 3 If one drives dangerously then the license is suspended for two years.
Mario drives dangerously on 1^{st} January 2008.
Date of publication: 1^{st} September 2007.
Date of entry in force: 15^{th} September 2007.
Period of force: 15^{th} September 2007 onwards.
Period of efficacy: 15^{th} September 2006 onwards.
Period of application: $[t, t+1]$ (where t is the time when an instance of the provision's condition has taken place).
Period of occurrence: from 1^{st} January 2008 to 1^{st} January 2009.

- during the period from 10/6/1997 to 16/6/1997, anyone who parks in front of the station is liable to a fine.

In the analytical approach one sentence represents the substantive content of the norm, and other sentences specify its temporal features. In the example given above, instead of having just one sentence, we could have had two sentences:

- Anyone who parks in front of the station is liable to a fine.
- Norm 1 is in force from 10/6/1997 to 16/6/1997.

or even three sentences:

- Anyone who parks in front of the station is liable to a fine.
- Norm 1 starts to be in force at 10/6/1997.
- Norm 1 ceases to be in force at 16/6/1997.

On the one hand, analytical representations of time have some drawbacks. In particular, more than one sentence needs to be considered in order to determine both the substantial content of the norm and its subsumption interval. Moreover, inferences from analytical representations must take into account the interaction between substantive norms and norms which regulate subsumption intervals.

On the other hand, the analytical approach has a number of advantages: (i) it is modular and clear, (ii) temporal elements do not need to be always directly specified, but they can be made dependent upon future and possibly not yet known facts.

In the remainder, we adopt the analytical approach. Accordingly, we indicate the time of force and the time of efficacy of a provision by temporal literals. A provision, represented by a rule r, efficacious at time t shall be formulated as:

$$\mathrm{Hold}^{t}_{ag} efficacious(r). \tag{6.19}$$

The following formulate a provision represented by a rule r in force at time t:

$$\mathrm{Hold}^{t}_{ag} force(r). \tag{6.20}$$

In general, the date of entry of force is established on function of the date of document's publication in an official journal. For example, in the Italian normative system, the date of entry of force is establish after a 15-day period (called *vacatio legis*) starting at the date of publication. We shall formulate the vactio legis as follows:

$$vl\colon\quad \mathrm{Hold}_{ag}^{t} published(r) \Rightarrow \mathrm{Hold}_{ag}^{[t+15,max]} force(r). \tag{6.21}$$

The time of existence t_e of a rule is captured by prefixing the rule by $\mathrm{Hold}_{ag}^{t_e}$:

$$\mathrm{Hold}_{ag}^{t_e}(r\colon\quad \alpha_1,\ldots,\alpha_n \Rightarrow \beta). \tag{6.22}$$

Intuitively, the formula means that the rule r holds at time t_e from the viewpoint of agent ag. The time t_e here is not intented to capture the time of force, instead it is the time for which the rule holds and is interpreted and the time of existence of the represented provision. For instance, the provision, existing in 1990 onwards, stating that members are obliged to cooperate, can be formulated as:

$$\mathrm{Hold}_{ag}^{[1990,max]}(r\colon\quad \mathrm{Hold}_{ag}^{t} member(x) \Rightarrow \mathrm{Obl}_{ag}^{t} cooperate(x)). \tag{6.23}$$

As any provision is formalized as a rule relating some antecedents to a consequent, the legal time of efficacy corresponds either to the time of the antecedents (in case of conditioned provisions) or to the time of the consequent (in case of unconditioned provisions). For our purposes, and for the sake of simplicity, in the remainder the time of efficacy shall always refer to the time of antecedents of the rule. In case of unconditioned provisions, as the set of antecedents of the rule is empty, there is no need to specify the time of efficacy (though the time of legal efficacy corresponds to the time of application of the rule).

6.2 Temporal reasoning schemata

Our reasoning schemata must be adapted to the introduction of temporal aspects, in particular to legal ones. Next, we analyze the integration of these aspects to the detachment schema.

6.2.1 Temporal detachment schemata

The different legal temporal dimensions identified in the previous Section raises some issues on the applicability of rules. For example, consider the following rule:

$$r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta. \tag{6.24}$$

To be applied, such rule should be accompanied by $\mathrm{Hold}_{ag}^{t_\alpha}\alpha$ with a temporal reference t_α which has to fulfill some temporal constraints regarding the time of force and of efficacy of the rule r. To conclude defeasibly $\mathrm{Hold}_{ag}^{t_\alpha}\beta$, a first constraint is that the rule r has to be in force and efficacious at the same time than α, i.e. $\mathrm{Hold}_{ag}^{t_\alpha} force(r)$ and $\mathrm{Hold}_{ag}^{t_\alpha} efficacious(r)$. We express this by the following temporal detachment:

$$\frac{\begin{array}{l}\mathrm{Hold}_{ag}^{t_\alpha}\alpha,\\ r\text{:}\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta,\\ \mathrm{Hold}_{ag}^{t_f} force(r),\\ \mathrm{Hold}_{ag}^{t_e} efficacious(r),\\ t_f = t_e = t_\alpha\end{array}}{\mathrm{Hold}_{ag}^{t_\alpha}\beta.} \tag{6.25}$$

For instance, given the provision “Members are obliged to cooperate”, the facts that this provision is in force and efficacious in 2007, and the statement “Mario believes that he is a member in 2007”, we can conclude defeasibly that “Mario is obliged in 2007 to cooperate”. This can be expressed instantiating the reasoning schema given above:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{mario}}^{2007} member(\mathrm{mario}),\\ \mathrm{r}\text{:}\quad \mathrm{Hold}_{ag}^{t} member(x) \Rightarrow \mathrm{Obl}_{ag}^{t} cooperate(x),\\ \mathrm{Hold}_{\mathrm{mario}}^{2007} force(\mathrm{r}),\\ \mathrm{Hold}_{\mathrm{mario}}^{2007} efficacious(\mathrm{r})\end{array}}{\mathrm{Obl}_{\mathrm{mario}}^{2007} cooperate(\mathrm{mario})} \tag{6.26}$$

An issue arises when the rule is not in force and efficacious at the same time, that is, when $t_f \neq t_e$. To overcome such issue, a solution is to use the notion of temporal viewpoint. Temporal viewpoints are of the utmost importance when one has to deal with retroactive or ultra-active. Roughly, an ultra-active norm involves a time of efficacy which is posterior to the time of force, while a retroactive norm involves a time of efficacy which is anterior to the time of force.[1] In our example, the rule r is applicable in the logical sense if, in a first temporal viewpoint, α holds when the rule r is in force, and, in a second temporal viewpoint, holds when the rule r is efficacious, that is, the rule r is applicable if we have $\mathrm{Hold}_{ag}^{t_f}\mathrm{Hold}_{ag}^{t_e}\alpha$. We express this by the following temporal detachment:

$$\frac{\begin{array}{l}\mathrm{Hold}_{ag}^{t_f}\mathrm{Hold}_{ag}^{t_\alpha}\alpha,\\ r\text{:}\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta,\\ \mathrm{Hold}_{ag}^{t_f} force(r),\\ \mathrm{Hold}_{ag}^{t_f}\mathrm{Hold}_{ag}^{t_e} efficacious(r),\\ t_\alpha = t_e\end{array}}{\mathrm{Hold}_{ag}^{t_f}\mathrm{Hold}_{ag}^{t_\alpha}\beta} \tag{6.27}$$

[1] Retroactive laws are sometimes referred in the literature as *ex post facto* laws. Notice that in some domains, retroactive rules are not permitted. This is the case of criminal law, where the principle “Nullum crimen, nulla poena sine praevia lege poenali” is in most cases applied (see [142, 143] for an investigation of the justifiability of retroactivity).

For instance, suppose that the rule given in (6.26) is in force in 2007 and efficacious in 2006 (thus the rule expresses a retroactive provision). We can instantiate the schema given in (6.27) as follows:

$$\begin{array}{l} \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{member}(\mathrm{mario}), \\ \mathrm{r}\colon\quad \mathrm{Hold}^{t}_{ag}\mathit{member}(x) \Rightarrow \mathrm{Obl}^{t}_{ag}\mathit{cooperate}(x), \\ \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathit{force}(r), \\ \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{efficacious}(r) \\ \hline \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Obl}^{2006}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}) \end{array} \tag{6.28}$$

Another complication arises when the rule to apply is itself temporalised. For example, suppose that the rule $r\colon\quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{t}_{ag}\beta$ holds at time t_r:

$$\mathrm{Hold}^{t_r}_{ag}(r\colon\quad \mathrm{Hold}^{t}_{ag}\alpha \Rightarrow \mathrm{Hold}^{t}_{ag}\beta) \tag{6.29}$$

We assume that the conditions to apply such rule are similar to those we previously set, but the time t_r of the rule has to equal its time of force t_f, that is $t_r = t_f$. We have thus the following schema:

$$\begin{array}{l} \mathrm{Hold}^{t_f}_{ag}\mathrm{Hold}^{t_\alpha}_{ag}\alpha, \\ \mathrm{Hold}^{t_r}_{ag}(r\colon\quad \mathrm{Hold}^{t}_{ag}\beta \Rightarrow \mathrm{Hold}^{t}_{ag}\beta), \\ \mathrm{Hold}^{t_f}_{ag}\mathit{force}(r), \\ \mathrm{Hold}^{t_f}_{ag}\mathrm{Hold}^{t_e}_{ag}\mathit{efficacious}(r), \\ t_r = t_f, \\ t_\alpha = t_e \\ \hline \mathrm{Hold}^{t_f}_{ag}\mathrm{Hold}^{t_\alpha}_{ag}\beta \end{array} \tag{6.30}$$

For instance, given the policy "In 2007, members are obliged to cooperate", the facts that this policy is in force in 2007 and efficacious in 2006, and the statement "In 2007, Mario is a member in 2006", we can conclude defeasibly "In 2007, Mario is obliged in 2007 to cooperate."

$$\begin{array}{l} \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{ag}\mathit{member}(\mathrm{mario}), \\ \mathrm{Hold}^{2007}_{\mathrm{mario}}(\mathrm{r}\colon\quad \mathrm{Hold}^{t}_{ag}\mathit{member}(x) \Rightarrow \mathrm{Obl}^{t}_{ag}\mathit{cooperate}(x)), \\ \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathit{force}(r), \\ \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{efficacious}(r) \\ \hline \mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Obl}^{2007}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}) \end{array} \tag{6.31}$$

At this stage, the model of temporal reasoning is incomplete with regards to situation in which some pieces of temporal information are explicitly missing. For example, suppose that the rule r holding in 2007 is in force in 2007 and efficacious in 2007. We have the following knowledge base KB:

$$\begin{aligned}\mathrm{KB} = \{\ & \mathrm{Hold}_{ag}^{2007}\alpha,\\ & \mathrm{Hold}_{ag}^{2007}(r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta),\\ & \mathrm{Hold}_{ag}^{2007} force(r),\\ & \mathrm{Hold}_{ag}^{2007}\mathrm{Hold}_{ag}^{2007} efficacious(r)\}\end{aligned}$$

Following the schema (6.30), we cannot conclude $\mathrm{Hold}_{ag}^{2007}\mathrm{Hold}_{ag}^{2007}\beta$ whereas we should intuitively do. How to reconciliate our intuition with the present formalization? A solution is to treat time dimensions in a Russian-dolls like fashion. Let us illustrate it by means of temporal statements as exposed in the knowledge base KB given above. The first step consists in deriving all the statements holding in 2007 to constitute a knowledge base KB_{ag}^{2007} in which any element holds in 2007 from the viewpoint of agent *ag*:

$$\begin{aligned}\mathrm{KB}_{ag}^{2007} = \{\ & \alpha,\\ & r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta,\\ & force(r),\\ & \mathrm{Hold}_{ag}^{2007} efficacious(r)\}\end{aligned}$$

The idea is to make derivation using the statements contained in KB_{ag}^{2007}. At this stage, the statement α is atemporal while the antecedent of the rule is temporalised. To overcome such issue, a solution is to consider that α holds in 2007 from the viewpoint of agent *ag* since it is in the knowledge base constituted of statements holding from the viewpoint of agent *ag* in 2007 . Doing so, we can apply the rule $r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta$ (because from the previous step we have $\mathrm{Hold}_{ag}^{2007}\alpha$) and hence we can derive $\mathrm{Hold}_{ag}^{2007}\beta$. Since the last result holds in the knowledge base KB_{ag}^{2007} in which statements hold for 2007, we can conclude $\mathrm{Hold}_{ag}^{2007}\mathrm{Hold}_{ag}^{2007}\beta$ which satisfies our initial expectations.

Treating time dimensions in a such Russian-dolls like fashion can be captured by accepting the reasoning schema according which from an assertion $\mathrm{Hold}_{ag}^{t}\gamma$ we infer $\mathrm{Hold}_{ag}^{t}\mathrm{Hold}_{ag}^{t}\gamma$:

$$\frac{\mathrm{Hold}_{ag}^{t}\gamma}{\mathrm{Hold}_{ag}^{t}\mathrm{Hold}_{ag}^{t}\gamma} \tag{6.32}$$

While the reasoning schema given above permits to reincorporate some intuitions into our formalization, some other intuitions are not captured in other cases in which some pieces of temporal information are missing. Consider for example the following knowledge base:

$$\begin{aligned}\mathrm{KB} = \{\ & \mathrm{Hold}_{ag}^{2006}\alpha,\\ & \mathrm{Hold}_{ag}^{2007}(r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta),\\ & \mathrm{Hold}_{ag}^{2007} force(r),\\ & \mathrm{Hold}_{ag}^{2006} efficacious(r)\}\end{aligned}$$

Following our approach, the rule r is not applicable and hence we cannot derive $\mathrm{Hold}_{ag}^{2007}\mathrm{Hold}_{ag}^{2006}\beta$. To make it applicable, we need explicitly the statement

$\mathrm{Hold}_{ag}^{2007}\mathrm{Hold}_{ag}^{2006}\alpha$. A solution is to extend the reasoning schema given in 6.32 so that $\mathrm{Hold}_{ag}^{t}\gamma$ entails $\mathrm{Hold}_{ag}^{t'}\mathrm{Hold}_{ag}^{t}\gamma$ with $t \leq t'$:

$$\frac{\begin{array}{l}\mathrm{Hold}_{ag}^{t}\gamma,\\ t \leq t'\end{array}}{\mathrm{Hold}_{ag}^{t'}\mathrm{Hold}_{ag}^{t}\gamma} \tag{6.33}$$

This schema is the inferential counter-part of an introspective schema rule that we shall introduce later in Section 6.2.3.

Another issue arises when some statements are not associated with any temporal information at all, for example, when a rule $r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta$ is not prefixed by any temporal operator. To overcome such issue we assume that atemporal statements hold objectively at any time:

$$\frac{\gamma}{\mathrm{Hold}_{\mathrm{obj}}^{t}\gamma} \tag{6.34}$$

where γ is any statement. For example, consider the following knowledge base:

$$\begin{array}{ll}\mathrm{KB} = \{ & \mathrm{Hold}_{\mathrm{obj}}^{2007}\mathrm{Hold}_{\mathrm{obj}}^{2006}\alpha,\\ & r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta,\\ & \mathrm{Hold}_{\mathrm{obj}}^{2007}force(r),\\ & \mathrm{Hold}_{\mathrm{obj}}^{2007}\mathrm{Hold}_{\mathrm{obj}}^{2006}efficacious(r)\}\end{array}$$

The rule r is not prefixed with any temporal operator, so we instantiate the schema (6.34) as follows:

$$\frac{r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta}{\mathrm{Hold}_{\mathrm{obj}}^{t}(r\colon\quad \mathrm{Hold}_{ag}^{t}\alpha \Rightarrow \mathrm{Hold}_{ag}^{t}\beta).} \tag{6.35}$$

so that we can conclude $\mathrm{Hold}_{\mathrm{obj}}^{2007}\mathrm{Hold}_{\mathrm{obj}}^{2006}\beta$ by means of schema (6.30).

6.2.2 General reasoning schema

In the foregoing, we have identified several types of temporal detachment schemata. All these detachment schemata can be accounted by a general reasoning schema. Let Φ denote a sequence of operator $(\phi_i^{t_i})_{1..n}$ where ϕ_i is an operator Hold_{ag} (the sequence Φ can be empty), and ψ_{ag}^{t} is any operator. We have:

$$\frac{\begin{array}{l}\Phi\mathrm{Hold}_{ag}^{t_f}\psi_{1ag'}^{t_e}\alpha_1,\ldots,\Phi\mathrm{Hold}_{ag}^{t_f}\psi_{nag'}^{t_e}\alpha_n,\\ \Phi\mathrm{Hold}_{ag}^{t_f}(r\colon\quad \psi_{1ag'}^{t_1}\alpha_1,\ldots,\psi_{nag'}^{t_n}\alpha_n \hookrightarrow \beta),\\ \Phi\mathrm{Hold}_{ag}^{t_f}force(r),\\ \Phi\mathrm{Hold}_{ag}^{t_f}\mathrm{Hold}_{ag}^{t_e}efficacious(r),\ldots,\Phi\mathrm{Hold}_{ag}^{t_f}\mathrm{Hold}_{ag}^{t_e}efficacious(r),\end{array}}{\Phi\mathrm{Hold}_{ag}^{t_f}\beta} \tag{6.36}$$

where $\hookrightarrow$ stands for either $\rightarrow$, $\Rightarrow$ or $\rightsquigarrow$.

With the reasoning schema given above, an agent can emulate its own reasoning: For instance, given that Mario believes in 2008 that, (i) from his viewpoint in 2007, he was student in 2006, and (ii) the policy indicating that members are obliged to cooperate holds in 2007, (iii) the policy is in force in 2007 and (iv) is efficacious in 2006, we can conclude defeasibly that Mario believes in 2008 that, from his viewpoint in 2007, he was obliged in 2006 to cooperate.

$$\frac{\begin{array}{l}\mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{member}(\mathrm{mario}),\\ \mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}(r\colon\quad \mathrm{Hold}^{t}_{ag}\mathit{member}(x)\Rightarrow \mathrm{Obl}^{t}_{ag}\mathit{cooperate}(x)),\\ \mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathit{force}(r),\\ \mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{efficacious}(r)\end{array}}{\mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Obl}^{2006}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario})} \tag{6.37}$$

If an agent can emulate its own reasoning schema then he may also emulate the reasoning schema of other agents. For instance, giving that Guido believes in 2008 that, (i) from the point of view of 2007, Mario was student in 2006, and (ii) the policy indicating that members are obliged to cooperate holds in 2007, (iii) the policy is in force in 2007 and (iii) is efficacious in 2006, we can conclude defeasibly that Guido believes in 2008 that, from the viewpoint of Mario in 2007, Mario was obliged in 2006 to cooperate.

$$\frac{\begin{array}{l}\mathrm{Hold}^{2008}_{\mathrm{guido}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{member}(\mathrm{mario}),\\ \mathrm{Hold}^{2008}_{\mathrm{guido}}\mathrm{Hold}^{2007}_{\mathrm{mario}}(r\colon\quad \mathrm{Hold}^{t}_{ag}\mathit{member}(x)\Rightarrow \mathrm{Obl}^{t}_{ag}\mathit{cooperate}(x)),\\ \mathrm{Hold}^{2008}_{\mathrm{guido}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathit{force}(r),\\ \mathrm{Hold}^{2008}_{\mathrm{guido}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\mathit{efficacious}(r)\end{array}}{\mathrm{Hold}^{2008}_{\mathrm{guido}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Obl}^{2006}_{\mathrm{mario}}\mathit{cooperate}(\mathrm{mario}).} \tag{6.38}$$

Next, we account for introspection and reflexion in the temporal setting.

6.2.3 Introspection and reflexion with time

In temporal setting, the reasoning schemata of introspection and emulation given in Section 5.1.3 must be somewhat adapted.

We assume introspection about statements of the form $\Phi_{ag}\gamma$ where γ is any statement, and Φ is an element of the set $\{\mathrm{Hold}_{ag}$, $\neg\mathrm{Hold}_{ag}$, Des_{ag}, $\neg\mathrm{Des}_{ag}$, Bring_{ag}, $\neg\mathrm{Bring}_{ag}$, Obl_{ag}, $\neg\mathrm{Obl}_{ag}$, Forb_{ag}, $\neg\mathrm{Forb}_{ag}$, Perm_{ag}, $\neg\mathrm{Perm}_{ag}$, Fac_{ag}, $\neg\mathrm{Fac}_{ag}$ $\}$. Accordingly, we have the following defeasible reasoning schema:

$$\frac{\Phi^{t}_{ag}\gamma}{\mathrm{Hold}^{[t,max]}_{ag}\Phi^{t}_{ag}\gamma} \tag{6.39}$$

where *max* is the higher boundary of our bounded ordered set of instants of time. For instance, if Mario believes in 2007 that he will not be a student in 2008, then we can conclude defeasibly that Mario believes in 2007 that Mario believes in 2007 onwards that he will not be a student in 2008:

$$\frac{\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2008}_{\mathrm{mario}}\neg \mathit{student}(\mathrm{mario})}{\mathrm{Hold}^{[2007,max]}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2008}_{\mathrm{mario}}\neg \mathit{student}(\mathrm{mario})} \tag{6.40}$$

Inversely, we assume temporal reflexion as follows:

$$\frac{\mathrm{Hold}^{t}_{ag}\Phi^{t}_{ag}\gamma}{\Phi^{t}_{ag}\gamma} \tag{6.41}$$

For instance, suppose that Mario believes in 2007 that Mario believes in 2007 that Mario believed in 2006 that Mario was not a student, we conclude defeasibly that Mario believes in 2007 that Mario believed in 2006 that Mario was not a student:

$$\frac{\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\ \neg \mathit{student}(\mathrm{mario})}{\mathrm{Hold}^{2007}_{\mathrm{mario}}\mathrm{Hold}^{2006}_{\mathrm{mario}}\ \neg \mathit{student}(\mathrm{mario})} \tag{6.42}$$

In the foregoing, both phenomena of introspection and reflexion are captured at the level of reasoning schemata. However, it is arguable that both phenomena are defeasible with respect to the substantial content of statements and some preferences. Since preferences stand between rules, both phenomena of introspection and reflexion are more natural expressed at the rule level. For this reason, the introspection schema (6.39) shall be replaced by the following defeasible schema rule:

$$r_{\Phi^{t}_{ag}\gamma \Rightarrow \mathrm{Hold}^{[t,max]}_{ag}\Phi_{ag}\gamma}: \quad \Phi^{t}_{ag}\gamma \Rightarrow \mathrm{Hold}^{[t,max]}_{ag}\Phi^{t}_{ag}\gamma. \tag{6.43}$$

Notice that the schema rule is labeled by its own content to insure its unique labeling, and so that a preference can be set among rule labels.

Likewise, we replace the schemata (5.20) by the following defeasible schema rule:

$$r_{\mathrm{Hold}^{t}_{ag}\Phi^{t}_{ag}\gamma \Rightarrow \Phi^{t}_{ag}\gamma}: \quad \mathrm{Hold}^{t}_{ag}\Phi^{t}_{ag}\gamma \Rightarrow \Phi^{t}_{ag}\gamma. \tag{6.44}$$

Beside introspection and reflexion, we assume that given any statement, we can entail defeasibly that this statement holds objectively at any time, and inversely. Accordingly, we have the following rule:

$$r_{\gamma \Rightarrow \mathrm{Hold}^{t}_{\mathrm{obj}}\gamma}: \quad \gamma \Rightarrow \mathrm{Hold}^{t}_{\mathrm{obj}}\gamma, \tag{6.45}$$

$$r_{\mathrm{Hold}^{[min,max]}_{\mathrm{obj}}\gamma \Rightarrow \gamma}: \quad \mathrm{Hold}^{[min,max]}_{\mathrm{obj}}\gamma \Rightarrow \gamma. \tag{6.46}$$

where *min* and *max* are the lower and higher boundaries of a bounded ordered set of instants of time.

6.3 Collisions of temporal conclusions

In Section 5.5, standard conflicting sets of modal statements were provided and we introduced agent types to detect and solve conflicts between the different components of the cognitive profiles of agents deliberation. In this Section, we investigate sets of conflicting statements, and how to handle such conflicts in a temporal setting.

6.3.1 Conflicts

We identified in Section 5.5 the criteria defining conflicts between statements. In a temporal setting, these criteria have to include time. Our assumption is that conflicting statements in the temporal setting are those of the atemporal setting positioned at the same temporal coordinate. Accordingly, let ϕ be any operator of the set $\{\mathrm{Hold}_{ag}, \mathrm{Des}_{ag}, \mathrm{Bring}_{ag}\}$, we have:

- $\mathscr{C}_{\mathrm{onflict}}(\neg\phi^t\gamma) = \{\phi^t\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\phi^t\gamma) = \{\phi^t{\sim}\gamma\}$.
- $\mathscr{C}_{\mathrm{onflict}}(\mathrm{Obl}^t_{ag}\gamma) = \{\neg\mathrm{Obl}^t_{ag}\gamma,\ \mathrm{Obl}^t_{ag}{\sim}\gamma,\ \neg\mathrm{Perm}^t_{ag}\gamma,\ \mathrm{Perm}^t_{ag}{\sim}\gamma,\ \mathrm{Forb}^t_{ag}\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\neg\mathrm{Obl}^t_{ag}\gamma) = \{\mathrm{Obl}^t_{ag}\gamma,\ \neg\mathrm{Perm}^t_{ag}\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\mathrm{Perm}^t_{ag}\gamma) = \{\mathrm{Obl}^t_{ag}{\sim}\gamma,\ \neg\mathrm{Perm}^t_{ag}\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\neg\mathrm{Perm}^t_{ag}\gamma) = \{\neg\mathrm{Obl}^t_{ag}{\sim}\gamma,\ \mathrm{Perm}^t_{ag}\gamma,\ \neg\mathrm{Perm}{\sim}\gamma,\ \mathrm{Obl}\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\mathrm{Forb}^t_{ag}\gamma) = \{\mathrm{Obl}^t_{ag}\gamma,\ \mathrm{Perm}^t_{ag}\gamma,\ \mathrm{Forb}{\sim}\gamma,\ \neg\mathrm{Forb}\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\neg\mathrm{Forb}^t_{ag}\gamma) = \{\neg\mathrm{Obl}^t_{ag}\gamma,\ \mathrm{Forb}^t_{ag}\gamma,\ \neg\mathrm{Forb}^t_{ag}{\sim}\gamma\}$,
- $\mathscr{C}_{\mathrm{onflict}}(\mathrm{Fac}^t_{ag}\gamma) = \{\mathrm{Obl}^t_{ag}{\sim}\gamma, \neg\mathrm{Perm}^t_{ag}\gamma, \mathrm{Forb}^t_{ag}\gamma, \mathrm{Obl}^t_{ag}\gamma, \neg\mathrm{Perm}^t_{ag}{\sim}\gamma,\ \mathrm{Forb}^t_{ag}{\sim}\gamma\}$.

For example, we have $\mathrm{Forb}^{2009}_{\mathrm{guido}}\mathit{member} \in \mathscr{C}_{\mathrm{onflict}}(\mathrm{Perm}^{2009}_{\mathrm{guido}}\mathit{member})$, and $\mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Forb}^{2009}_{\mathrm{guido}}\mathit{member} \in \mathscr{C}_{\mathrm{onflict}}(\mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Perm}^{2009}_{\mathrm{guido}}\mathit{member})$. However, $\mathrm{Hold}^{2009}_{\mathrm{mario}}\mathrm{Forb}^{2009}_{\mathrm{guido}}\mathit{member}$ and $\mathrm{Hold}^{2008}_{\mathrm{mario}}\mathrm{Perm}^{2009}_{\mathrm{guido}}\mathit{member}$ do not conflict because they are not positioned at the same temporal coordinate.

As in the atemporal setting, we can distinguish conflicts valid for any agent and conflicts specific to some particular agents (see Section 5.5.1). Likewise conflicts valid for any agent in the temporal setting, we assume that specific conflicts in the temporal setting are those of the atemporal setting positioned at the same temporal viewpoints (see Table 6.2).

Furthermore, the epistemic reasoning on atemporal conflicts is extended to the temporal ones as follows:

- Given $Y^t_{ag}\beta \in \mathscr{C}^*_{\mathrm{onflict}}(X^t_{ag}\gamma)$, we have $\mathrm{Hold}^{t'}_{\mathrm{obj}}Y^t_{ag}\beta \in \mathscr{C}^*_{\mathrm{onflict}}(\mathrm{Hold}^{t'}_{\mathrm{obj}}X^t_{ag}\gamma)$,
- Given $Y^t_{ag}\beta \in \mathscr{C}^*_{\mathrm{onflict}}(X^t_{ag}\gamma)$, we have $\mathrm{Hold}^{t'}_{ag}Y^t_{ag}\beta \in \mathscr{C}^*_{\mathrm{onflict}}(\mathrm{Hold}^{t'}_{ag}X^t_{ag}\gamma)$ such that $t' \geq t$,
- Given $\mathrm{Hold}^t_{ag}Y^t_{ag}\beta \in \mathscr{C}^*_{\mathrm{onflict}}(\mathrm{Hold}^t_{ag}X^t_{ag}\gamma)$, we have $Y^t_{ag}\beta \in \mathscr{C}^*_{\mathrm{onflict}}(X^t_{ag}\gamma)$,
- Given $\beta' \in \mathscr{C}^*_{\mathrm{onflict}}(\gamma)$ and $\beta \equiv \beta'$, we have $\beta \in \mathscr{C}^*_{\mathrm{onflict}}(\gamma)$.

Conflict type	Agent type
$\text{Forb}^t_{ag}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Des}^t_{ag}\gamma)$	Desire-compliant
$\text{Forb}^t_{ag}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Bring}^t_{ag}\gamma)$	Action-compliant
$\text{Des}^t_{ag}{\sim}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Bring}^t_{ag}\gamma)$	Slothful
$\text{Hold}^t_{ag}\text{Hold}_{ag}{\sim}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}^t_{ag}\gamma)$	1-introspective consistent
$\text{Hold}^t_{ag}\neg\text{Hold}_{ag}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}^t_{ag}\gamma)$	2-introspective consistent
$\text{Forb}^t_{ag}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Des}^t_{ag}\gamma)$	Desire-deviant
$\text{Forb}^t_{ag}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Bring}^t_{ag}\gamma)$	Action-deviant
$\text{Des}^t_{ag}{\sim}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Bring}^t_{ag}\gamma)$	Unstable
$\text{Hold}^t_{ag}\text{Hold}_{ag}{\sim}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Hold}^t_{ag}\gamma)$	1-introspective inconsistent
$\text{Hold}^t_{ag}\neg\text{Hold}_{ag}\gamma \notin \mathscr{C}^*_{\text{onflict}}(\text{Hold}^t_{ag}\gamma)$	2-introspective inconsistent

Table 6.2. Types of conflict.

For example, given the specific conflict regarding Mario $\text{Forb}^{2008}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Des}^{2008}_{\text{mario}}\gamma)$, we can derive among others $\text{Hold}^{2010}_{\text{obj}}\text{Forb}^{2008}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}^{2010}_{\text{obj}}\text{Des}^{2008}_{\text{mario}}\gamma)$ and $\text{Hold}^{2010}_{\text{mario}}\text{Forb}^{2008}_{\text{mario}}\gamma \in \mathscr{C}^*_{\text{onflict}}(\text{Hold}^{2010}_{\text{mario}}\text{Des}^{2008}_{\text{mario}}\gamma)$.

6.3.2 Preferences

In the atemporal framework, we saw that conflicts between defeasible conclusions can be resolved on the basis of local preferences expressed by an explicit strength order between rules or global preferences between types of statements. In the following, we accommodate local and global preferences in our temporal framework.

Concerning local preferences, the determination of the strength order can be informed by the principle *lex posterior derogat legi priori* according to which a subsequent rule derogates the previous one. For instance, consider a fiscal provision stating that if one has the income in excess of fifty thousand in 2007 then one has to pay the tax A in 2008. The following formalize it:

$$\text{Hold}^{[2007,max]}_{\text{obj}}(r_{6.54}\text{:}\quad \text{Hold}^{2007}_{ag}\,income(x) > 50 \Rightarrow \text{Obl}^{2008}_{x}\,taxA). \tag{6.47}$$

The provision enters in force and is efficacious in 2007. Suppose now that, for whatever reason, the legislator provides a retroactive provision stating that if one has the income in excess of fifty thousand in 2007 then one *does not* have to pay the tax A in 2008:

$$\text{Hold}^{[2008,max]}_{\text{obj}}(r_{6.48}\text{:}\quad \text{Hold}^{2007}_{ag}\,income(x) > 50 \Rightarrow \neg\text{Obl}^{2008}_{x}\,taxA). \tag{6.48}$$

The provision enters in force in 2008 and is efficacious in 2007. Suppose as a fact that Mario had an income in excess of fifty thousand in 2007:

$$\mathrm{Hold}_{\mathrm{obj}}^{[2007,max]}\mathrm{Hold}_{\mathrm{obj}}^{2007} income(\mathrm{mario}) > 50.$$

Both rules $r_{6.54}$ and $r_{6.48}$ are applicable in 2008: the rule $r_{6.54}$ supports the conclusion $\mathrm{Hold}_{ag}^{2008}\mathrm{Obl}_{\mathrm{mario}}^{2007} taxA$ and the rule $r_{6.48}$ supports the conclusion $\mathrm{Hold}_{ag}^{2008}\neg\mathrm{Obl}_{\mathrm{mario}}^{2007} taxA$, and thus we obtain a conflict. The rule $r_{6.48}$ is subsequent to the rule $r_{6.54}$ and thus overrides it: we conclude $\mathrm{Hold}_{ag}^{2008}\neg\mathrm{Obl}_{\mathrm{mario}}^{2007} taxA$ and discard $\mathrm{Hold}_{ag}^{2008}\mathrm{Obl}_{\mathrm{mario}}^{2007} taxA$.

We do not address explicitly in our framework the principle 'lex posterior derogat legi priori', and we assume that a strength order implicitly taking this principle into account is stabilized between two rules labels in order to indicate the relative strength of each rule. For example, we write $r_{6.48} \succ r_{6.54}$ to indicate that the rule $r_{6.48}$ is stronger than $r_{6.54}$ and assume that it implcitly accounts for the principle 'lex posterior derogat legi priori'.

Nevertheless, preferences over rules can hold from some viewpoints and will have the form $(X_i)_{1..n}(r_2 \succ r_1)$ where X_i is an epistemic operator Hold_{ag}^{t}. For instance, "It holds objectively in 2008 that $r_{6.48}$ is stronger than $r_{6.54}$" is formulated as:

$$\mathrm{Hold}_{\mathrm{obj}}^{2008}(r_{6.48} \succ r_{6.54}).$$

Accordingly, a conflict between conclusions held from some viewpoints have to be resolved by some preferences held from the same viewpoints.

The epistemic reasoning on atemporal strength orders is extended to the temporal ones as follows:

- Given $(X_i)_{1..n}(r_2 \succ r_1)$, we have $\mathrm{Hold}_{\mathrm{obj}}^{t'}(X_i)_{1..n}(r_2 \succ r_1)$,
- Given $\mathrm{Hold}_{ag}^{t}(X_i)_{1..n}(r_2 \succ r_1)$, we have $\mathrm{Hold}_{ag}^{t'}\mathrm{Hold}_{ag}^{t}(X_i)_{1..n}(r_2 \succ r_1)$ such that $t' \geq t$,
- Given $\mathrm{Hold}_{ag}^{t}\mathrm{Hold}_{ag}^{t}(X_i)_{1..n}(r_2 \succ r_1)$, we have $\mathrm{Hold}_{ag}^{t}(X_i)_{1..n}(r_2 \succ r_1)$.

The first item indicates that any preference holds objectively at any time. The second and third item express temporal introspection and reflexion over preferences, respectively.

The use of local preferences to resolved conflicts are illustrated in Figures 6.2 to 6.9. In each Figure, the literal c holds in the gray areas while ¬c holds in the black areas, and neither c nor ¬c hold in the white areas.

Concerning global preferences, likewise our assumption that specific conflicts in the temporal setting are those of the atemporal setting positioned at the same temporal viewpoints, we assume that global preferences in the temporal setting are those of the atemporal setting positioned at the same temporal viewpoints (see Table 6.3). We assume similar temporal epistemic inferences:

- Given $Y_{ag}^{t}\beta \in \mathscr{D}_{\mathrm{efeat}}(X_{ag}^{t}\gamma)$, we have $\mathrm{Hold}_{\mathrm{obj}}^{t'}Y_{ag}^{t}\beta \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{\mathrm{obj}}^{t'}X_{ag}^{t}\gamma)$,
- Given $Y_{ag}^{t}\beta \in \mathscr{D}_{\mathrm{efeat}}(X_{ag}^{t}\gamma)$, we have $\mathrm{Hold}_{ag}^{t'}Y_{ag}^{t}\beta \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{ag}^{t'}X_{ag}^{t}\gamma)$ such that $t' \geq t$,
- Given $\mathrm{Hold}_{ag}^{t}Y_{ag}^{t}\beta \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}_{ag}^{t}X_{ag}^{t}\gamma)$, we have $Y_{ag}^{t}\beta \in \mathscr{D}_{\mathrm{efeat}}(X_{ag}^{t}\gamma)$,

- Given $\beta' \in \mathscr{D}_{\mathrm{efeat}}(\gamma)$ and $\beta \equiv \beta'$, we have $\beta \in \mathscr{D}_{\mathrm{efeat}}(\gamma)$.

For example, given $\mathrm{Forb}^{2008}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}^{2008}_{\mathrm{mario}}\gamma)$, we can derive among others $\mathrm{Hold}^{2010}_{\mathrm{obj}}\mathrm{Forb}^{2008}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}^{2010}_{\mathrm{obj}}\mathrm{Des}^{2008}_{\mathrm{mario}}\gamma)$ and $\mathrm{Hold}^{2010}_{\mathrm{mario}}\mathrm{Forb}^{2008}_{\mathrm{mario}}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Hold}^{2010}_{\mathrm{mario}}\mathrm{Des}^{2008}_{\mathrm{mario}}\gamma)$.

Defeat type	Agent type
$\mathrm{Forb}^{t}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}^{t}_{ag}\gamma)$	Desire-social
$\mathrm{Des}^{t}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Forb}^{t}_{ag}\gamma)$	Desire-unsocial
$\mathrm{Forb}^{t}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Bring}^{t}_{ag}\gamma)$	Action-social
$\mathrm{Bring}^{t}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Forb}^{t}_{ag}\gamma)$	Action-unsocial
$\mathrm{Des}^{t}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Bring}^{t}_{ag}{\sim}\gamma)$	Willful
$\mathrm{Bring}^{t}_{ag}\gamma \in \mathscr{D}_{\mathrm{efeat}}(\mathrm{Des}^{t}_{ag}{\sim}\gamma)$	Involuntary

Table 6.3. Types of defeat.

Finally, as in the atemporal setting, we assume that a global preference hold by default in the sense that if a local preference is in contradiction with the later then the local preference take precedence over the global preference.

6.4 Normative modifications

Following [157], a *legal system* is defined as a set of documents fixed at a defined time t and which have been issued by an authority and whose validity depends on rules that determine, for any given time, whether a single document belongs to the system. Formally:

$$LS(t) = \{D_1(t), D_2(t), D_3(t), \dots, D_n(t)\} \tag{6.49}$$

where $D_n(t)$ denotes a document at a fixed time t in a discrete representation.

A *normative system*, in turn, takes the documents belonging to a legal system and organizes them to reflect their evolution over time. A normative system should therefore be more precisely defined as a particular discrete time-series of legal systems that evolves over time. In formal terms:

$$NS = \{LS(t_1), LS(t_2), LS(t_3), \dots, LS(t_j)\} \tag{6.50}$$

Changes of a normative system over time can be effected by normative modifications. Normative modifications are provisions which modify other provisions or the arrangement of provisions inside a normative system.

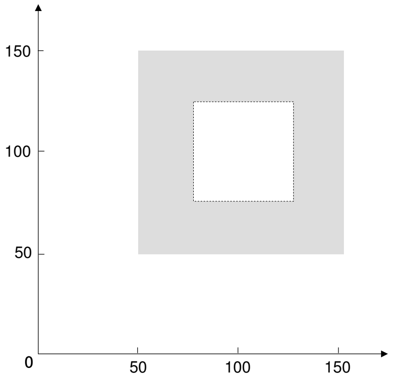


Fig. 6.2.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c}),$
$\mathrm{Hold}_{\mathrm{ag}}^{[75,125]}(s{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[75,125]}\neg\mathrm{c}),$
$>= \varnothing.$

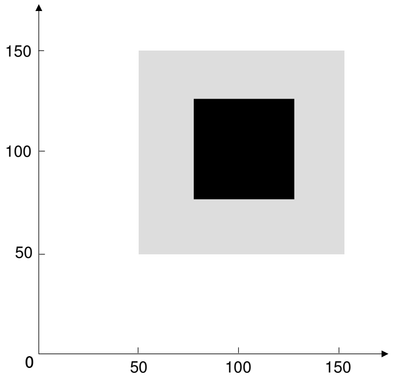


Fig. 6.3.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c}),$
$\mathrm{Hold}_{\mathrm{ag}}^{[75,125]}(s{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[75,125]}\neg\mathrm{c}),$
$>= \{s \succ r\}.$

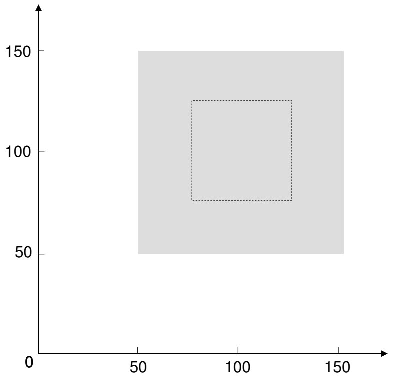


Fig. 6.4.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c}),$
$\mathrm{Hold}_{\mathrm{ag}}^{[75,125]}(s{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[75,125]}\neg\mathrm{c}),$
$>= \{s \prec r\}.$

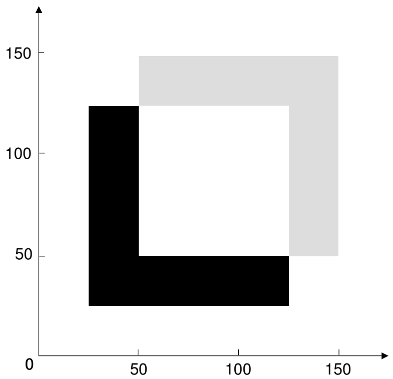


Fig. 6.5.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c}),$
$\mathrm{Hold}_{\mathrm{ag}}^{[25,125]}(s{:}\qquad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[25,125]}\neg\mathrm{c}),$
$>= \varnothing.$

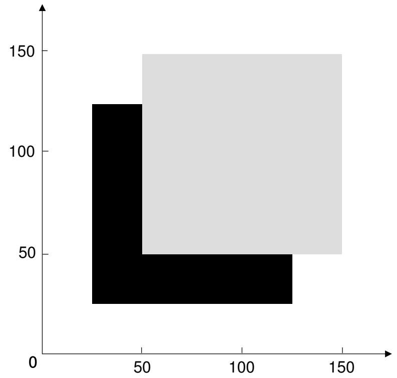


Fig. 6.6.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c})$,
$\mathrm{Hold}_{\mathrm{ag}}^{[25,125]}(s\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[25,125]}\neg\mathrm{c})$,
$>= \{s \prec r\}$.

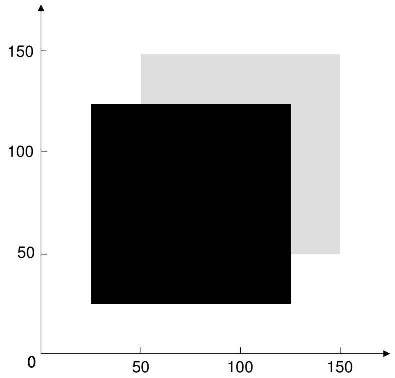


Fig. 6.7.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c})$,
$\mathrm{Hold}_{\mathrm{ag}}^{[25,125]}(s\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[25,125]}\neg\mathrm{c})$,
$>= \{s \succ r\}$.

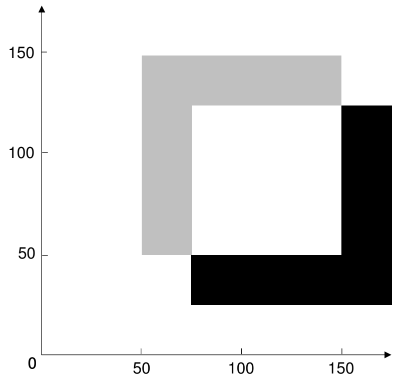


Fig. 6.8.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c})$,
$\mathrm{Hold}_{\mathrm{ag}}^{[25,125]}(s\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[75,175]}\neg\mathrm{c})$,
$>= \varnothing$.

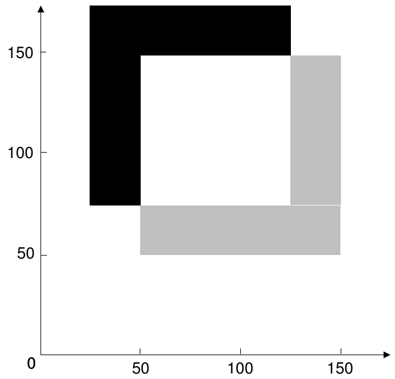


Fig. 6.9.
$\mathrm{Hold}_{\mathrm{ag}}^{[50,150]}(r\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[50,150]}\mathrm{c})$,
$\mathrm{Hold}_{\mathrm{ag}}^{[75,175]}(s\colon \quad \Rightarrow \mathrm{Hold}_{\mathrm{ag}}^{[25,125]}\neg\mathrm{c})$,
$>= \varnothing$.

6.4.1 Categorization of modifications

A well-documented analysis of normative modifications can be found in [155]. The modifying provision is called the active provision, while a modified provision is called the passive provision. The identification of the passive norms is generally done by means of a normative reference. If the reference is complete and accurate then the reference is said to be explicit, whereas if the reference is incomplete or not accurate then it is said implicit. Furthermore, if both the modifying action and the passive norm are certain, then the corresponding modification is explicit. Accordingly, implicit modifications are any modification whose the modifying action is uncertain or the passive norm is implicit. Implicit modifications are classically introduced by a formula of the form “All norms in conflict with present article are hereby repealed”.

There exist different types of modifying actions. We adopt the categorization provided in [155], and accordingly, normative modifications are divided in two broad categories:

- norm-specific modifications, and
- legal-system modifications.

These top level categories are then refined into sub-categories and these in turn into a third level which specifies types of modifications (see Table 6.4).

Table 6.4. Categorizations of normative modifications

First level	Second level	Third level
Norm-specific modification	Modifications of content	Textual modifications
		Modifications of meaning
	Modifications of scope	Derogations
		Extensions
	Temporal modifications	Modifications of force
		Modifications of efficacy
Legal-system modification		

Norm-specific modifications modify other normative provisions or normative acts singly. Modifications on a single disposition can be further classified as below.

- Content modifications which can be further classified into modifications of meaning and textual modifications. A modification of meaning modifies the meaning of a provision without modifying the text of it. A textual modifications modifies the text of the passive provision. Textual modifications includes substitutions, integrations, relocations and repeal.
- Modifications of scope do not alter the text but restrict or extend the range of application of the passive provision. Modifications of scope includes derogations and extensions.

- Temporal modifications impact on the time of force, the time of efficacy of the passive provisions. Examples of modifications of the time of force are annulment, renewal, prorogations, initiation and termination of the period of force. Modifications of the time of efficacy includes prorogation, suspension, initiation or termination of the period of efficacy,

Legal-system modifications modify the relationships between the norms included in a legal system instead of a single norm. Such modifications generally occur by altering the normative functions of the passive provisions. An example is the ratification of an international treaty: the treaty is already in force in international law and must be integrated to the Italian normative system by means of a ratification. Another example is about the relations holding between delegating norms and delegated norm, as when a government issues an emergency legislative decree by delegation by the parliament. A delegated norm does not have any reason exist without a constitutional relation with a delegating norm.

6.4.2 Temporal dimensions of modifications

As any provision, a modificatory provision has a time of force and a time of efficacy. The period of force is the time during which the modificatory provision partakes in the normative system. The time of efficacy is (i) for unconditioned provisions, the time of application of the modification (see below) (ii) for conditioned provisions, the time associated to the conditions. We distinguish two times with respect to modifications:

- the time of the application of the modification associated to the modifying action itself, and
- the time of the effect of the modification associated to the effect of the modifying action.

Note that for an unconditioned modificatory provision, the time of the application of the modification is the time of legal efficacy of the provision. Hereafter, as an unconditioned provision is modeled for our purposes as a rule with empty conditions, the time of efficacy shall always refer to the time of conditions of the rule. If the conditions are empty, then there is no need to specify the time of efficacy, and the time of legal efficacy shall correspond to the time of the application.

In general, the time of force, the time of the application of the modification, and the time of effect of the modification coincide, but in some cases they are different as for example retroactive modifications or modifications in the past.

- A modification is *retroactive* if the time of causation of the modification is in the past w.r.t. the time of force.
- A modification is *ultra-active* if the time of causation of the modification is in the future w.r.t. the time of force.
- A modification is *in the past* if the time of causation of the modification and its time of force are contemporaneous and the time of the effect of the modification is in the past w.r.t. the time of force.

- A modification is *in the future* if the time of causation of the modification and its time of force are contemporaneous and the time of the effect of the modification is in the future w.r.t. the time of force.

Examples of modification in the past and in the future are illustrated in the Tables 6.5 and 6.6 respectively.

Table 6.5. An example of modification in the past.

Art. 3 The article 45 of the Act n.20/2004 is substituted as following from the 1^{st} January 2005. “Art. 45 1. All the citizen with three children in 2009 should be have a discharge of the tax of 3000 Euro in 2010” Art. 15 This act enters into operation the 1^{st} January 2006.
Time of enter into force: 1^{st} January 2006.
Time of efficacy: 1^{st} January 2006.
Date of the application of the modification: 1^{st} January 2006.
Date of the effect of the modification: 1^{st} January 2005.

Table 6.6. An example of modification in the future.

Art. 3 The article 45 of the Act n.30/2006 is repealed from the 1^{st} January 2008. Art. 10 This act enters into operation the 1^{st} June 2007.
Time of enter into force: 1^{st} January 2007.
Time of efficacy: 1^{st} June 2007.
Period of the application of the modification: 1^{st} June 2007.
Time of the effect of the modification: 1^{st} January 2008.

6.4.3 Temporal modifications

A temporal modificatory provision shall be modeled as a conditional statement relating some conditions $\alpha_1 \ldots \alpha_n$ to a temporal modification represented as $\mathrm{Hold}_{ag}^{[t_1,t_2]}\mathrm{Hold}_{ag}^{[t'_1,t'_2]}\beta$ where β stands for $force(r)$, $\neg force(r)$, $efficacious(r)$ or $\neg efficacious(r)$, depending on the type of temporal modifications. The modified provision is represented by the rule label r. The interval $[t_1,t_2]$ represents the time of the application of the modification while the interval $[t_1,t_2]$ captures the time of the effect of the modification.

$$r\colon \quad \alpha_1 \ldots \alpha_n \Rightarrow \mathrm{Hold}_{ag}^{[t_1,t_2]}\mathrm{Hold}_{ag}^{[t'_1,t'_2]}\beta. \tag{6.51}$$

For example, the unconditioned abrogation in 2007 of a provision represented by the rule r can be formulated as:

$$mr\colon \qquad \Rightarrow \mathrm{Hold}_{ag}^{[2007,max]}\mathrm{Hold}_{ag}^{[2007,max]}\neg force(r). \tag{6.52}$$

The unconditioned suspension between $[2007,2008]$ of a rule r can be formulated as:

$$mr\colon \qquad \Rightarrow \mathrm{Hold}_{ag}^{[2007,2008]}\mathrm{Hold}_{ag}^{[2007,2008]}\neg force(r). \tag{6.53}$$

A modification can be ultra-active or retroactive. A type of retroactive modifications deserving special attention concerns modifications subverting from the outset (*ex-tunc*) all the effects of the passive provision in the meantime (i.e. from the time of its entry into force to the time the modification is set forth). Often cited ex-tunc modifications are annulments (e.g. the lapsing out of force of a legislative decree) and abrogations that cancels out all the earlier effects since the passive provision came into force.

Let's illustrate an ex-tunc annulment. Consider the fiscal provision stating that if one has an income in excess of fifty thousand in 2007 then one has to pay the tax A in 2008. The provision exists and is in force in 2007 onwards.

$$\begin{array}{l}\mathrm{Hold}_{\mathrm{obj}}^{[2007,max]}(r_{6.54}\colon \quad \mathrm{Hold}_{ag}^{2007} income(x) > 50 \Rightarrow \mathrm{Obl}_{ag}^{2008} taxA),\\ \mathrm{Hold}_{\mathrm{obj}}^{[2007,max]} force(r_{6.54}).\end{array} \tag{6.54}$$

Suppose this provision is accompanied by the provision stating that if the tax A is due in 2008 then she does not have to pay the tax B in 2008. This provision exists and is force in 2007 onwards.

$$\begin{array}{l}\mathrm{Hold}_{\mathrm{obj}}^{[2007,max]}(r_{6.55}\colon \quad \mathrm{Obl}_{ag}^{2008} taxA \Rightarrow \neg\mathrm{Obl}_{ag}^{2008} taxB),\\ \mathrm{Hold}_{\mathrm{obj}}^{[2007,max]} force(r_{6.55}).\end{array} \tag{6.55}$$

Suppose now that, for whatever reason, the legislator annuls in 2008 the first provision (rule $r_{6.54}$). The annulment exists and is in force in 2009 (we do not indicate the time of efficacy since the annulment is unconditioned).

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{obj}}^{[2009,max]}(r_{6.56}\colon \qquad \Rightarrow \mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}\neg force(r_{6.54})),\\ \mathrm{Hold}_{\mathrm{obj}}^{[2009,max]} force(r_{6.56})\end{array}}{\mathrm{Hold}_{\mathrm{obj}}^{[2009,max]}\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]} force(r_{6.54}).} \tag{6.56}$$

Suppose that Mario had an income in excess of seventy thousand in 2007: the rules $r_{6.54}$ and $r_{6.55}$ can be applied. From the viewpoint of 2007, Mario has to pay in

2008 Tax A, and hence by applying $r_{6.55}$, Mario does not have to pay in 2008 Tax B. In year 2009, the annulment is applied: the first fiscal provision with its effects disappear from the normative system. Hence, from the viewpoint of 2009 onwards, we cannot derive anymore that Mario has to pay in 2008 Tax A, and consequently Mario does not have to pay Tax B in 2008.

6.4.4 Textual modifications

Textual modifications operate in a first level in which provisions are expressed in natural languages such as English, French etc. For example, a substitution replaces usually a piece of text expressed in natural language with another piece of text. Automation of textual modifications at this level are usually found into traditional consolidation techniques (see e.g. [30]) in which the logic formalization of provisions and the associated formalized reasoning play no role. As our aim is to formalize legal reasoning under logical forms, we limit textual modifications to operate in a second level in which provisions are formalized. We cannot address here the correspondence from the first level to the second level, and will investigate textual modification in the latter only. This formal reduction allows us to analyze modifications in way which can be discussed, for example to clarify temporal aspects, and which can be used in autonomous norm-governed multi-agent systems in which agents can modify norms autonomously.

A textual modificatory provision can be modeled as a conditional statement relating some conditions $\alpha_1 \dots \alpha_n$ to a textual modification captured by $textual\_modification$:

$$r\colon \quad \alpha_1 \dots \alpha_n \Rightarrow \mathrm{Hold}^{t}_{\mathrm{obj}}\mathrm{Hold}^{t'}_{\mathrm{obj}} textual\_modification. \tag{6.57}$$

where t is the time of application of the modification, and t' is the time of effect of the application. The proper textual modification $textual\_modification$ shall be modeled as a function that takes as input the passive provision to be modified and outputs the corresponding modified provision. In the remainder, we shall focus on any kind of modification that can be reduced to a textual substitution. Textual substitutions as "substitute in rule r the string $string_1$ by $string_2$" can be formulated as:

$$\mathrm{Hold}^{t}_{ag}\mathrm{Hold}^{t'}_{ag} substitute(r, Cont(r), string_1, string_2). \tag{6.58}$$

where t is the modification time, and t' is the application time, and $Cont(r)$ is the string corresponding to the serialization of the content of the rule r. The substitution returns a string $Cont'(r)$ corresponding to the serialization of the new content of the rule r in which a specified substring $string_1$ has been replaced with another substring $string_2$. Accordingly, we provide below the temporal reasoning schema:

$$\frac{\begin{array}{l}\mathrm{Hold}^{tf}_{ag}\mathrm{Hold}^{tm}_{ag}\mathrm{Hold}^{ts}_{ag}(r\colon \quad Cont(r)),\\ \mathrm{Hold}^{tf}_{ag}\mathrm{Hold}^{tm}_{ag}\mathrm{Hold}^{ts}_{ag} substitute(r, Cont(r), string_1, string_2))\end{array}}{\mathrm{Hold}^{tf}_{ag}\mathrm{Hold}^{tm}_{ag}\mathrm{Hold}^{ts}_{ag}(r\colon \quad substitute(r, Cont(r), string_1, string_2)).} \tag{6.59}$$

Let us illustrate the above reasoning schema. Consider, for example, a reform of pension system made in 2007 stating that 60 years old people can no longer retire.

$$\mathrm{Hold}_{ag}^{[2007,max]}(r\colon \quad \mathrm{Hold}_{ag}^{t} 60\_years\_old(x) \Rightarrow \neg \mathrm{Perm}_{x}^{t} retire(x)). \tag{6.60}$$

Suppose that a new government deliberates in 2008 to modify the above provision. Formally, the modification consists in a substitution of ¬Perm by Perm, its modification and application time is in 2008.

$$\mathrm{Hold}_{ag}^{2008}\mathrm{Hold}_{ag}^{2008} substitute(r, Cont(r), \neg\mathrm{Perm}, \mathrm{Perm}). \tag{6.61}$$

By instantiating the above schema we shall obtain:

$$\frac{\begin{array}{l}\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}(r\colon \quad \mathrm{Hold}_{ag}^{t} 60\_years\_old(x) \Rightarrow \neg \mathrm{Perm}_{x}^{t} retire(x)),\\ \mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}\mathrm{Hold}_{ag}^{[2008,max]} substitute(r, Cont(r), \neg\mathrm{Perm}, \mathrm{Perm}))\end{array}}{\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}\mathrm{Hold}_{\mathrm{obj}}^{[2008,max]}(r\colon \quad \mathrm{Hold}_{ag}^{t} 60\_years\_old(x) \Rightarrow \mathrm{Perm}_{x}^{t} retire(x))} \tag{6.62}$$

A modification can apply only if some constraints are fulfilled and, in some scenarios a modification cannot apply. In the remainder of this Section we investigate such constraints and pathological scenarios. In order to ease the discussion, we shall abstract from temporal references in the following, and assume that things hold at the same instants.

The most obvious constraint to modify a rule r is that this rule has to be present in the set of rules. In other words and more generally it is not possible to modify something that does not exist: for example, we cannot apply usually in sequence a substitution on the same rule. Notice that this assumption does not constrain the modified rule to be in force: indeed we can modify a provision which is not yet in force.

Other cases in which a modification cannot be applied, are those in which textual modifications conflict. Consider for example the following rules:

$$\begin{array}{ll} r_4\colon & a \Rightarrow b,\\ r_5\colon & \quad\quad \Rightarrow substitute(r_4, Cont(r_4), \Rightarrow b, \rightsquigarrow x),\\ r_7\colon & \quad\quad \Rightarrow substitute(r_4, Cont(r_4), \Rightarrow b, \rightsquigarrow d).\end{array} \tag{6.63}$$

Clearly, the rule r_5 and r_7 conflicts. How to solve this conflict? If the rule r_5 is stronger than r_7, then, intuitively, the substitution involved by the rule r_7 should apply. However, if no superiority relation holds between the rules, then conflicts are more delicate to handle. Two approaches are possible. A first possibility is an 'ambiguity propagation' flavored approach in which any scenario applying the modifications is parsed. For example, given the above rules r_4, r_5 and r_7 then we could derive:

$$\begin{array}{ll} r_4\colon & a \rightsquigarrow c,\\ r_4\colon & a \rightsquigarrow d.\end{array}$$

A second approach is an ‘ambiguity blocking’ approach, that is, no modification at all is applied. If no modification is applied then the question arises whether the target rule should be derived. In a cautious approach as ‘ambiguity blocking’, it is coherent to block the derivation of the target rule, but it is nevertheless possible to consider a weak ‘ambiguity blocking’ approach with which the unmodified target rule is derived. In the following, only the strong “ambiguity blocking” approach is considered to favor legal cautiousness and the principle that a rule cannot have different content, or in other words, we cannot derive two rules with the same label (and at the same time) with different content. Hence, several modifications applying on the same rule may not imply that they conflict. Indeed, consider the following rules:

$$\begin{array}{rl} r_8: & e \Rightarrow f, \\ r_9: & \Rightarrow substitute(r_8, Cont(r_8), \Rightarrow f, \leadsto g), \\ r_{10}: & \Rightarrow substitute(r_8, Cont(r_8), e, d). \end{array}$$

then, we should be able to derive:

$$r_8: \quad d \leadsto g,$$

In general, conflicts between modifications occur when they apply on the same element of a rule. If some modifications apply on different elements of a rule then they may not conflict. However, even if they do not conflict then pathological situations, in which the content of a rule is instable, may also arises. For example, consider the following rules:

$$\begin{array}{rl} r_{11}: & e \Rightarrow f, \\ r_{12}: & \Rightarrow substitute(r_{11}, Cont(r_{11}), \Rightarrow f, \leadsto g), \\ r_{13}: & \Rightarrow substitute(r_{11}, Cont(r_{11}), \leadsto g, \Rightarrow f). \end{array}$$

We can apply first r_{12} on r_{11}, then r_{13}, then r_{12} and so on. In such pathological cases, according to the principle that a rule cannot have different contents, neither r_{11}: $e \Rightarrow f$ nor r_{11}: $e \leadsto g$ should be derived. Another pathological case occurs when a rule triggers its own modification. For example, consider the following rule:

$$r_{15}: \quad f \Rightarrow substitute(r_{15}, Cont(r_{15}), e, d).$$

Self-modification can also occur indirectly when the consequent of a rule triggers a second rule modifying the first:

$$\begin{array}{rl} r_{14}: & e \Rightarrow f, \\ r_{15}: & f \Rightarrow substitute(r_{14}, Cont(r_{14}), e, d). \end{array}$$

In such a complex setting, the formalization of textual modification requires a complex formalization. Importantly, the automation of legal consolidation of provisions expressed in natural languages requires more than such complex formalization: indeed, it also requires the automation of the correspondence between (i) textual modifications operating on provisions expressed in natural languages, and (ii) textual modifications operating on formalized provisions.

In normative multi-agent computer systems, if textual modifications aim at the modification of normative rules then the formalization of such textual modifications seems unnecessary. The reason lies in the observation that textual modifications can be emulated by the abrogation of the rule to be modified and the introduction of a new rule similar as the modified rule would be. Doing so, a complex formalization is avoided and replaced by much more efficient mechanisms to modify normative systems.

Part III

Formal

7

Logic formalization

In the previous Part, an informal model accounting for the interaction between normative systems and cognitive agents is proposed. In this Part, we move on its formalization, by stating formal rules governing classes of expressions, as a step toward an implementation.

Many formalizations can be provided, and a selection has to be made on some features as soundness and completeness of course, but also computability, expressiveness, and ergonomics. Next, in Section 7.1, the appropriateness of logic to formalize legal reasoning as it can be often argued in the literature (e.g. see [191]) is briefly introduced. The following Sections take a gentle tour through well-known logic formalisms, and for each formalisms, its appropriateness for our purpose is investigated.

7.1 Logic and law

The use of logic to formalize legal reasoning preceded the use of computers. For example, S. Kanger [111] and L. Lindhal [123] investigations on normative positions formalized C. Hohfeld's account of complex normative concepts as duties and rights [104]. C. Alchourrón and E. Bulygin [2] used logic to analyze normative systems. With the arrival of computer applications for the legal domain, and especially artificial intelligence applied to it, logic has been used as the major tool to formalize legal reasoning and has been developed in many directions. Next, the appropriateness of logic to formalize legal reasoning as it can be often discussed in the literature (see e.g. G. Sartor in [191]) is briefly introduced.

7.1.1 Legal reasoning as deductive reasoning

An initial attraction between law and logic comes from the analogy between the logical deduction of a conclusion from a set of axioms, and the justification of a decision from legally binding sources. This analogy stems from the way both jurists and logicians may proceed to master a system by specifying rules that give account of this system. In this regard, G. Sartor in [191] quotes C. Perelman and L. Olbrechts-Tyteca [160]'s description of such logical approach:

> [An approach], which may be called logical, is that in which the primary concern is to resolve beforehand all the difficulties and problems which can arise in the most varied situations, which one tries to imagine by applying the rules, laws and norms one is accepting. This is usually the approach of the scientist, who tries to formulate laws which appear to him to govern the area of his study and which, he hopes, will account for all the phenomena which can occur in it. It is also the usual approach of someone who is developing a legal or ethical doctrine and who proposes to resolve, if not all the cases where it applies, at least the greatest possible number of those with which one might be concerned in practice. The person who in the course of his life imitates the theorists we have just referred to is regarded as a logical man, in the sense in which the French are logical and the English are practical. The logical approach assumes that one can clarify sufficiently the ideas one uses, make sufficiently clear the rules one invokes, so that practical problems can be resolved without difficulty by the simple process of deduction. This implies moreover that the unforeseen has been eliminated, that the future has been mastered, that all problems have become technically soluble.

An agent who has adopted such logical style of decision making, is guided by subsuming facts to rules in order to derive the relevant decisions. Such view is adopted typically by computer legal rule-based systems (see Section 2.2.2), and for example, M. Sergot et al. in [199] formalized convincingly the British Nationality Act using logic programming techniques. However, such deductive view may raise several problems which are briefly discussed below.

Firstly, in many cases, (legal) decision making cannot merely reach by subsuming facts to rules since such rules may not exist. For example, legislators drafting normative texts cannot envisage all the circumstances in which they will be applied so that the law can result 'incomplete'. In more general terms, it is argued that since the law has for object a world which is inherently not fully observable in space and in time, which is dynamic and unpredictable, the deductive logical approach is deemed to fail.

A response consists in arguing that a deductive reconstruction of legal reasoning represents a normative model, rather than a mere description of legal practice, so that logic formalisms are intended as an ideal instead of a description of real legal practice. Close to this position, another response lies in the proposition that logic formalisms are meant to reconstruct the external justification phase of decision making instead of guiding an internal derivation of decisions.

A second objection related to the first on the incompleteness of the deductive model concerns the fact that classical logic formalisms are *monotonic*, that is, the addition of new premises as new evidences or new rules cannot invalidate formerly derived conclusions. The monotonicity of classical logics really matters if we observe that, for example, to overcome legal incompleteness, a first law is usually provided with general scope, and then exceptions are added so that the law can fit better to the diverse actual situations.

The usual response to this objection on the monotonicity of classical logic formalisms holds in the consideration that logic formalisms can be conceived to feature *non-monotonicity*, that is, the addition of new premises can invalidate formerly derivable consequences (see Section 7.4).

A third objection related to the second lies in the observation that conflicts are central to the law but classical logic systems do not handle such conflicts since they aim at avoiding inconsistencies and that they do not permit reasoning with such inconsistencies.

Though this objection is valid for classical logics, it does not apply in general on the light of recent development on logics (particularly in non-monotonic logics) in which reasoning with inconsistent information is possible.

A fourth objection about the deductive view dwells on the observation that logic deals primarily with the truth property of descriptive sentences. However, norms like imperatives are often not understood as neither true nor false. For example, the sentences as ”Leave the room!” or ”You can enter the room” are not intented to describe some aspects of the world. It has been argued that if they are not descriptive, then they cannot have truth-values, and by consequence be compounded by truth-functional connectives as in classical logics. Hence, though a logical study of norms may intuitively exist, there cannot be a logic of it. This is the Jörgensen’s dilemma (from the name of the Danish legal theorist J. Jörgensen who discussed it, see [108]).

A response is that the central notion of logic is the idea of inference, rather than the notion of truth-value. On this view, logic as a inference tool can be applied to imperative norms. Another related response is based on the classical distinction between norms and normative statements as recalled by J. Hansen et al. in [101]:

> Though norms are neither true or false, one may state that *according to the norms*, something ought to be (be done) or is permitted: the statement “John ought to leave the room”, “Mary is permitted to enter”, are then true or false descriptions of the normative situation. Such statements are sometimes called normative statements, as distinguished from norms.

Accordingly, a possible logic can be used to reason about normative statements reflecting logical properties of underlying norms.

A fifth objection is based on the observation that legal reasoning requires more than deduction but, for example, also analogy or induction.

This objection is the basis of some works aiming at combining different types of reasoning. For example, A. Gardner in [207] has proposed to combine rule and case-based approaches, in the sense of using cases when the rules run out. In [173], H. Prakken and G. Sartor have shown that case-based reasoning may be embedded in the theory of defeasible argumentation which is usually used in rule-based reasoning system.

7.1.2 Logic as a tool to remove impreciseness

Another argument supporting the use of logic in the legal domain is rooted in the tradition of positivism where law is expected to be more certain and less ambiguous: logic formalisms can appear as a tool for removing impreciseness from the legal discourse. For example, the use of propositional logic for representing legal texts was advocated by L. Allen in [6] arguing that the syntax of propositional logic would enable legal drafters to avoid unintended syntactical ambiguities, and so prevent litigation. If syntactical impreciseness is a possible source for different interpretations, then logic could, for example, constrain the arbitrariness of some human discretional decision-making.

However, this view was also largely criticized by observing that logic formalisms can hardly capture the rich expressiveness of natural language in which legal provisions are generally stated (on this point see e.g. G. Sartor's investigations in [189, 186, 188]). This observation goes in pair with the issue of legal interpretation which is critical when laws is formulated deliberately in an rather elastic and evasive way. Indeed, as well remarked by R. Borruso in [39]:

> Specie nei regimi democratico-parlamentari, ove i governi sono sorretti da maggioranze poltiche instabili costituite dalla coalizione di partiti molto spesso diversi per ideologia, la necessità di trovare un accordo basato sul compromesso spinge spesso a formulare le leggi in maniera evasiva o elastica propio perché, altrimenti, l'accordo non si raggiungerebbe. Ma un tal modo di legiferare aumenta sempre piu - patologicamente - l'importanza della giurisprudenza e da ai giudici un potere enorme. Perché, [...] mediante l'interpretazione della legge, come di qualsiasi altro testo scritto in linguaggio naturale, si può integrare o ridurre o comunque modificare e, in taluni casi limte, capovolgere la portata di una norma, tantoché non è affatto azzardato definire creativo nel senso piu completo del termine, l'intervento della guirisprudenza.

So, even if legal provisions are certain, in practice, legal reasoning is characteristically uncertain and that refers traditionally to the pervasive problem of so-called *open texture*. The problem of open texture arises when the conditions for the application of a legal disposition are not straightforward, but instead are left to the discretion of the reasoner. Attempts to deal with open texture have led to distinguish different types of open texture. The most common types of open texture identified in the literature refer to ambiguity and vagueness.

Open texture as ambiguity refers to terms that can have more than one definition. For example, the ambiguous legal term 'valid' can have many meanings as illustrated by the long on-ongoing debate among philosophers of law.

Vagueness means that it is not always clear whether a legal term should apply to a concrete fact situation. Vague concepts are not black or white with precise boundaries: though there are central prototypical cases whose classifications seem clear, concept boundaries are gray zones and contains cases with viable competing interpretations. The substantial difference between vagueness and ambiguity is subtle and might be illustrated by observing that vagueness often come to light only when legal language is confronted with a concrete case whereas ambiguity can be detected a priori with a dictionary.

Ambiguity and vagueness can involve classification difficulties as well illustrated by H. Hart in [102] considering a local provision that prohibits vehicles from entering a municipal park. While the concept of vehicles can be expected to apply to automobiles with little disagreement, there are a number of cases for which the application of the statute is debatable as for example a bicycle.

The wide acceptance that these types of open texture are distinct to each others has led to different techniques (see e.g. [161] for some investigations), but as remarked by T. Bench-Capon and P. Visser in [29], the analytic diagnosis of open-texture has led to an isolation of these techniques and few works attempt to combine the forces of each.

7.1.3 Logic for norm-governed computer systems

Due to the many barriers as pointed out by the sketchy discussion given above, and as a matter of fact, the use of logic to model real-life legal reasoning is limited, and S. Haack in [99] questioning the prospects of logic in the law, answers “something but not all” (to which the jurist and logician E. Bulygin replies in [49] by “not all, but more than something”). Another prospect is norm-governed computer systems for which works on logic in the law could provide simply a lot.

Indeed, norm-governed computer systems require solid theorical basis but does not need all the subtleties of real-life legal reasoning as, for example, open texture. Hence, norm-governed computer systems could profit of the logical analysis and techniques developed for modeling legal reasoning in order to formalize simple normative systems tailored to govern computer systems. This view is in line with R. Conte at al. in [58] who investigate how to fill the gap between the frameworks of autonomous agents and legal theory. Indeed as pointed out by R. Conte at al., though some gaps exists between these two frameworks in terms of language and formalisms used, theories of reference, objectives etc., normative multi-agent systems requires integrating works in the legal and multi-agent domains. R. Conte at al. in [58] argues:

> Most problems concerning regulation of the interaction of autonomous agents are linked to issues traditionally addressed by legal studies, and specifically, by legal doctrine and legal theory [. . .]. This is no surprise, since law is the most pervasive and developed normative system, and it is typically concerned with the government of autonomy: the fundamental task of the law is exactly that of providing normative reasons which may restrain and co-ordinate the behaviour of autonomous agents, [. . .].

In this view, if the interaction between normative systems and software cognitive agents requires integrating works in the legal and multi-agent domains then this integration can be strongly facilitated by a common formalism appropriate for both domains. It is with this idea in mind that we shall propose in the next Chapters a formalism based on defeasible logic.

In the remainder of this Chapter, as many criticisms were the occasion for many scholars to face new challenges that resulted in developments of more adequate logical techniques, we present briefly the most studied logic formalisms with respect to our purposes.

7.2 Classical logics

By classical logics, we intend to mean the class of formal logics that have been most intensively studied and most widely used. We shall focus on the two most well-known of them, namely classical propositional logic and (classical) first-order predicate logic.

7.2.1 Classical propositional logic

Classical propositional logic, also known as propositional calculus or sentential logic, aims at studying ways of combining propositions, statements or sentences to build more complex propositions, statements or sentences. Note that the term ‘proposition’ is here used synonymously with ‘statement’ and ‘sentences’ while it is sometimes intented as an higher abstraction level to name different statements or sentences which express the same thing. For instance, the English statement, “Mario is a student”, its translation statement in French “Mario est un étudiant”, would be considered as the same proposition. For the purposes of our presentation, the term ‘proposition’, ‘statement’ and ‘sentences’ are used interchangeably.

In any natural language, a statement rarely consists of a single word, instead a statement would be generally composed of a subject along with a verb. Propositional logic considers instead atomic statements, i.e. statements as indivisible wholes, and build complex statements which combine atomic statements.

To build formulas, we assume alphanumeric symbols as $a,b,c\ldots,0,1,2,\ldots$, and any symbol or sequence of such symbols is an atomic formula (or atom or propositional variable). For instance, the following are formulas:

$$student,$$

$$over\_25.$$

As notation, we shall use Greek letters $\alpha,\beta\ldots$ as meta-variables ranging over formulas. The assignment of truth-values of atomic formulas is called an *interpretation* (or a valuation), that is, an interpretation I in propositional logic assigns to each atomic formula a truth-value (false or true).

Propositional logic aims at studying ways of combining propositions, but such combinations are limited to statements in which the values of truth of complex statements depend on the values of truth of its constituting statements. Hence, a crucial question is what are these combinations. They are 16 possible combinations for two propositional variables γ and β. These combinations can be listed as a table, each line of which expresses a possible combination of truth-values for the simpler statements to which the combination applies, along with the resulting truth-value for the complex statement formed in each combination [98]:

γ	β	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Each combination 'connects' the truth-value of a complex statement with respect to the truth-values of its constituting statements, and at each combination is associated a *connective operator*. An extensive approach would be to define 16 connectives corresponding to the 16 possible combinations. Another approach, and it is the approach commonly adopted, is to chose adequate primitive connectives from which other combinations can be built. A primitive connective is traditionally the negation symbolized by $\neg$ which inverses the truth-value of a proposition and which is associated to the following truth table:

γ	$\neg\gamma$
1	0
0	1

The second primitive connective is traditionally either the conjunction or the disjunction respectively symbolized by $\vee$ and $\wedge$. Well-formed formulas using the connectives $\vee$ and $\wedge$ have the form $\gamma \vee \beta$ and $\gamma \wedge \beta$ respectively. These connectives correspond to the following truth tables:

γ	β	$\gamma \vee \beta$
1	1	1
0	1	1
1	0	1
0	0	0

γ	β	$\gamma \wedge \beta$
1	1	1
0	1	0
1	0	0
0	0	0

Note that the conjunction or the disjunction can be defined by means of the other and the connective negation. Indeed, it is easy to see that the compound statement $\gamma \wedge \beta$ has the same truth-value than the statement $\neg\gamma \wedge \neg\beta$.

In order to have more compact formula, two other connectives $\supset$ and $\equiv$ can be introduced, keeping in mind that we could define others. Well-formed formula using the connective $\supset$ have the form $\gamma \supset \beta$ and have the same truth-value than $\neg\gamma \vee \beta$. Hence the truth table of $\supset$ is the following:

γ β	$\gamma \supset \beta$
1 1	1
0 1	1
1 0	0
0 0	1

Well-formed formula using the connective $\equiv$ have the form $\gamma \equiv \beta$ and have the same truth-value than $(\gamma \supset \beta) \wedge (\beta \supset \gamma)$. Hence the truth table of $\equiv$ is the following:

γ β	$\gamma \equiv \beta$
1 1	1
0 1	0
1 0	0
0 0	1

In the case of propositional logic, a *semantics* provides an univoque and formal meaning to the different connectives and this univocal and formal meaning to the different connectives is specified by the above truth tables. An interpretation of a formula α (i.e. as we have seen previously the assignment of truth values of atomic formulas) is a *model* of α if and only if the interpretation I of the atomic formulas constituting α assigns the truth-value true to α, and shall be noted $I \models \gamma$. A formula γ is said *valid* if it holds in every interpretation of γ, i.e. $I \models \gamma$ for all interpretations I. To determine whether a formula is true in some interpretations, we can build the corresponding truth tables, or use some correct (truth-preserving) *rules of inferences*. For our purposes, we skip the presentation of such rules of inference because the analysis of the semantics provides us with enough insights in the adequacy of propositional logic for ordinary intuitions. Accordingly, it is perhaps worth noting that in propositional logic, connectives are *truth-functional*: the truth-value of a compound sentence depends only on the truth-value of its component. For this reason, classical propositional logic is said truth-functional, and this property raises some problems which are discussed in the following.

A first common problem found in the literature is about the restricted interpretation of the material conditional connective $\supset$. Since classical propositional logic is said truth-functional, the truth-value of a material conditional statement is determined by the truth-value of its constituting statement. This is perhaps more visible if one calls back that a material conditional statement $\gamma \supset \beta$ is equivalent to $(\neg\gamma) \vee \beta$. As a consequence, a material conditional $\gamma \supset \beta$ is true if its antecedent γ is false. For example, since the statement *the_earth_revolves_around_the_moon* is false, the following material conditional is true:

$$\neg the\_earth\_revolves\_around\_the\_moon \supset mario\_is\_obliged\_to\_fly. \tag{7.1}$$

Though the above statement is true in the setting of classical propositional logic, we do not expect it. The source of the issue is that most of ordinary conditionals are not material conditionals, i.e. their truth-values do not depend on the truth-values of its components.

A second criticism related to the former concerns the use of Modus Tollens (Latin for "the way that denies by denying"), i.e. contrapositive inference. Modus Tollens consists of the following inference: If $\gamma \supset \beta$ and $\neg\beta$ are true, then $\neg\gamma$ is true. Though modus Tollens is appreciable for material conditionals, its use is not appropriate for some other kinds of conditionals, and especially for exceptions. For example, suppose that the following statement is true:

$$student \supset discount. \tag{7.2}$$

Suppose that Mario does not have a discount, i.e., the proposition $\neg discount$ is true. By Modus Tollens, we can infer that Mario is not a student, i.e. the proposition $\neg student$ is true. However, intuitively, we cannot derive that Mario is not a student because he does have not a discount: indeed he may be for example a student which is over 25. In general, Modus Tollens is banned from formal defeasible reasoning and hence from formal legal reasoning.

A third common criticism is based on the fact that being truth-functional guarantees that by additional statements the truth-value of a compound statement does not change. In other words, no new statement can change the formally computed truth-value of a statement. As a consequence, the computed truth-value of a statement cannot be changed on light of further information. To qualify this feature, classical propositional logic is said *monotonic*. For example, consider that the informal sentences "if Mario is a student then Mario has a discount" and "Mario is a student" are respectively formalized as follows:

$$student \supset discount, \tag{7.3}$$

$$student. \tag{7.4}$$

Following the above truth tables, one can infer that the proposition *discount* is true. Suppose now that the informal sentences that "if Mario is over 25 years old then Mario has no discount" and "Mario is over 25 years old" are respectively formalized as follows:

$$over\_25 \supset \neg discount, \tag{7.5}$$

$$over\_25. \tag{7.6}$$

Following the above truth tables, one can infer that the proposition $\neg discount$ is true. We have hence infer that both *discount* and $\neg discount$ are true, that is, an inconsistency.

A fourth criticism which is connected to the previous one lies in the observation that in classical propositional logic, from a contradiction anything can be deduced (this is sometimes expressed in Latin by “Ex Falso Sequitur Quodlibet”). Indeed, given a contradiction α (i.e. α is always false), we can build the formula $\alpha \vee \beta$ where β is any formula. Since α is false then β is true. For example, suppose that we have as premise the contradiction $student \wedge \neg student$, the formula $(student \wedge \neg student) \vee mario\_is\_obliged\_to\_fly$ is valid and consequently we can derive that the proposition $mario\_is\_obliged\_to\_fly$ is true. The fact that from a contradiction, anything can be deduced is very important because, though conflicts (i.e. contradictions) are central to the law, anything cannot derived from them. Hence, classical propositional logic does not behave appropriately on the handling of contradictions for our purposes.

A fifth criticism dwells on the observation that many sentences are not truth-functional by nature. For instance, consider the sentence “Guido believes that Mario is not a student”. The truth-value of the sentence “Mario is not a student” does not determine the truth-value of “Guido believes that Mario is not a student”. Similarly, the truth-value of the statement “it is obligatory that Mario cooperate” does not only depend on the truth-value of the statement “Mario cooperate”. If we want a logic that accounts for such sentences, a remedy is that the truth-value of such modal statements does not only depend of the constituting statements but also to other statements. We shall see in Section 7.3 that modal logic aims at overcoming such issue.

Finally, though propositional logic can deal with many forms of statement and inference, some others cannot be captured by propositional logic. For instance, the well-known syllogism which from “All men are mortal” and “Socrate is a man” derives “Socrate is mortal”, cannot be accounted for in propositional logic. More generally, those forms of reasoning involving such words as “For all” or “There exists”, cannot be formalized in propositional logic. Such limitation really matters for normative reasoning since norms are usually addressed to multiple agents, each agent instantiating the norms to its own condition. To capture these forms of reasoning, propositional logic can be extended into a logic called predicate logic which we will see next.

7.2.2 First-order predicate logic

Propositional logic cannot capture forms of reasoning involving such words as “For all” or “There exists” to make general statements, for example about elements of a set. To capture these forms of reasoning, propositional logic can be extended into a logic called predicate logic. Predicate logic can be of the first-order or of higher order, but for our purposes, the presentation is limited to the first-order predicate logic, and shall be brief. More complete presentation of first-order predicate logic can found in many academic books (see e.g. [145]).

In predicate logic, a domain of discourse is modeled as individuals which can have some properties (these properties can be relations with other individuals). For example, we can consider that the person Mario is an individual having the property of being a student. An individual is denoted by a constant symbol and a property by a predicate symbol. For instance, the individual Mario can be denoted by the constant symbol m or $ds\_v3va5df$ and the property student by the predicate symbol s or $faff f$. For the sake of ergonomy, a symbol is usually chosen to suggest what it is intented to denote. Hence, in our case, the individual Mario and the property student will be denoted by the constant symbols "mario" and "*student*" respectively. Using predicate and constant symbols, we can symbolize the statement "Mario is a student" as $student(\text{mario})$. Note that while predicates can be used to denote a property of a single individual, they can also be used to express a relation between several individuals. For instance, the statement "Mario pays Guido" can be symbolized by $pay(\text{mario}, \text{guido})$.

When some individuals have the same properties or are related by the same relations, then they can be grouped to form sets of individuals or sets of n-tuple. If our intention is to reason about sets of individuals then the introduction of variables and quantifiers are invaluable. A variable refer to any individual within the domain of discourse, and is usually denoted by x, y, ... Using predicate and variable symbols, statements about undetermined individuals can be symbolized. For instance, the formula $student(x)$ expresses that an undetermined individual, referred as x, has the property of being a student. Quantifiers are used to quantify the variables over sets of individuals. In predicate logic, two quantifiers are usually considered: the universal quantifier symbolized by $\forall$ and the existential quantifier symbolized by $\exists$. The universal quantifier is used to bound a variable to any individual of a set. For example, the formula $\forall x\, student(x)$ expresses that for any individual x, x is a student. The existential quantifier is used to bound a variable to at least one individual of a set. For example, the formula $\exists x\, student(x)$ expresses that there exists an individual x such that x is a student.

So far, elements have been represented only by constant symbols. However, in domain of discourse, we would like to designate elements which are composed of other elements. *Functors* are intented to capture such constructions. For example, the family consisting of the parents Mario and Maria and children Carlo and Carla can be represented by the construction $family(\text{mario}, \text{maria}, child(\text{carlo}, \text{carla}))$. when the functors are unspecified they shall be symbolized by f, g, h etc.

Predicate logic is an extension of propositional logic and, uses the same connectives symbols ($\neg$, $\vee$, $\wedge$, $\supset$ and $\equiv$) to build compound statements. Finally, auxiliary symbols such as parentheses are used to limit the scope of other symbols.

To sum up, the alphabet of predicate logic consists of the following class of symbols: (i) variable symbols such as x, y, $\ldots$, (ii) constant symbols such as a, b, $\ldots$, (iii) functor symbols such as f, g, $\ldots$, (iv) predicate symbols such as p, q, r, $\ldots$, (v) logical connectives such as $\wedge$ (conjunction), $\neg$ (negation), $\equiv$ (logical equivalence), $\supset$ (implication) and $\vee$ (disjunction), (vi) quantifiers $\exists$ (existential) and $\forall$ (universal), and finally (vii) auxiliary symbols like parentheses and commas. To emphasize the arity n of a functor f, it is written in the form f/n. Similarly, to emphasize the arity n of a predicate p, it is written in the form p/n.

The objects of a domain are called terms whose syntax is defined as below:

Definition 7.1. *The set T of terms over a given alphabet $\mathscr{A}$ is the smallest set such that:*

- *Any constant in $\mathscr{A}$ is in T;*
- *Any variable in $\mathscr{A}$ is in T;*
- *If f/n is a functor in $\mathscr{A}$ and $t_1,\dots,t_n \in T$ then $f(t_1,\dots,t_n) \in T$*

Terms will be denoted by s and t. The syntax of well-formed formulas (wff) is defined as:

Definition 7.2. *Let T be the set of terms over the alphabet $\mathscr{A}$. The set $\mathscr{F}$ of wff (with respect to $\mathscr{A}$) is the smallest set such that:*

- *If p/n is a predicate symbol in $\mathscr{A}$ and $t1, tn \in T$ then $p(t1,\dots,tn) \in \mathscr{F}$*
- *If F and $G \in \mathscr{F}$, then so are $(F \wedge G)$, $(F \vee G)$, $(\neg F)$, $(F \supset G)$, $(F \equiv G)$*
- *If $F \in \mathscr{F}$ and x is a variable in $\mathscr{A}$ then $(\exists x F)$ and $(\forall x F) \in \mathscr{F}$*

Formulas of the form $p(t_1,\dots,t_n)$ are called *atomic formulas* (or simply *atoms*). Let F be a formula, an occurrence of the variable x in F is said to be *bound* if the occurrence is in the scope a quantifier $\forall$ or $\exists$, otherwise the occurrence is said to be *free*. A formula or a term which contains no variables is called a *ground*.

The previous paragraphs introduced the language of formulas as a formalization of a class of declarative statements of natural language. Such sentences refer to some “world” and may true or false in this world. Declarative statements refer to objects and relations on objects. The meaning of logic formula is defined relative to the abstraction of this world. The abstraction of this world is set of objects, and this set is called the *domain*. A relation between a domain (i.e. the abstraction of the world) and the set of bounded formulas is called an *interpretation*.

Definition 7.3 (Interpretation). *An interpretation I of an alphabet $\mathscr{A}$ is a domain $\mathscr{D}$ and a mapping that associates :*

- *Each constant $c \in \mathscr{A}$ with an element $c_I \in \mathscr{D}$*
- *Each n-ary functor $f \in \mathscr{A}$ with a function $f_I : \mathscr{D}_n \rightarrow \mathscr{D}$*
- *Each n-ary predicate symbol $p \in \mathscr{A}$ with a relation $p_I \subseteq \mathscr{D}^n$*

The domain $\mathscr{D}$ of an interpretation I will be sometimes written $|I|$. Remark that an interpretation does not deal with variable: a *valuation* is a mapping from variables of the alphabet to the domain of an interpretation.

Definition 7.4 (Semantics of terms). *Let I be an interpretation, υ a valuation and t a term. The meaning υ_I of t is an element of the domain $\mathscr{D}$ of interpret defined as follows:*

- *if t is a constant c, then $\upsilon_I(t) := c_I$,*
- *if t is a variable x, then $\upsilon_I(t) := \upsilon(x)$,*
- *if t is of the form $f(t_1,\dots,t_n)$, then $\upsilon_I(t) := f_I(\upsilon_I(t_1),\dots,\upsilon_I(t_n))$,*

The meaning of a formula is a truth-value (i.e. true or false) and, as in propositional logic, the truth-value of a formula depends on its components. However, given the refinement in the syntax due to the introduction of constants, variables, predicates, functors, quantifies, the semantics is also refined. In the following, the notation $I \models_{\upsilon} P$ stands for the statement "P is true with respect to the interpretation I and valuation υ", and $I \not\models_{\upsilon} P$ for "P is false with respect to the interpretation I and valuation υ"

Definition 7.5 (Semantics of wff's). *Let I be an interpretation, υ a valuation, P and Q well-formed formulas. The meaning of P, Q or combinations of P and Q w.r.t. I and υ is defined as follows:*

- $I \models_{\upsilon} p(t_1,\ldots,t_n)$ *iff* $\langle \upsilon_I(t_1),\ldots,\upsilon_I(t_n)\rangle \in p_I$,
- $I \models_{\upsilon} \neg P$ *iff* $I \not\models_{\upsilon} P$,
- $I \models_{\upsilon} P \wedge Q$ *iff* $I \models_{\upsilon} P$ *and* $I \models_{\upsilon} Q$,
- $I \models_{\upsilon} P \vee Q$ *iff* $I \models_{\upsilon} P$ *or* $I \models_{\upsilon} Q$ *(or both)*,
- $I \models_{\upsilon} P \supset Q$ *iff* $I \models_{\upsilon} Q$ *whenever* $I \models_{\upsilon} P$,
- $I \models_{\upsilon} P \equiv Q$ *iff* $I \models_{\upsilon} P \supset Q$ *and* $I \models_{\upsilon} Q \supset P$,
- $I \models_{\upsilon} \forall x P$ *iff* $I \models_{\upsilon} F$ *for every* $t \in |I|$,
- $I \models_{\upsilon} \exists x P$ *iff* $I \models_{\upsilon} F$ *for some* $t \in |I|$,

Given a set P of closed formulas and a domain $\mathscr{D}$ (i.e. an abstraction of a world), we are interested in finding out the interpretation(s) I in which the formulas of P can give a proper account of the domain, that is, the interpretation(s) I in which all formulas of P are true. Such interpretations are called models of P.

Definition 7.6. *An interpretation I is a* model *of a set P of closed formulas iff every formula of P is true in I.*

Given some explicit information about a system, we are interesting in deducing some others pieces of information about that system. In other words, given a model I of a set P of closed formula, we are interested in deducing other formulas Q from P which are also true in the model I. This leads to the notion of logical consequence.

Definition 7.7. *Let P be a set of closed formulas. A closed formula Q is a* logical consequence *of P (denoted $P \models Q$) iff Q is true in every model of P.*

As for propositional logic, the truth-value of a formula in some interpretations can be compiled using some rules of inferences. We do not provide them here and we will just notice that, in the general case, predicate logic is undecidable, i.e., it is undecidable whether a first-order logic formula is valid (or true under all possible interpretations). In other words, there is no decision procedure that can correctly determine whether a formula is valid. Eventually, first-order predicate logic is semi-decidable. If a formula is true given a set of axioms, there is a procedure that will determine this. However, if the formula is false, then there is no guarantee that a procedure will ever determine this and the procedure may never halt in this case. As we will see in Section (7.4.3) on logic programming, the issue of undecidability can be overcome by introducing restrictions on the language of formulas.

Furthermore, though first-order predicate logic is a response to the lack of expressiveness of propositional logic, since predicate logic is an extension of propositional logic, it inherits of the drawbacks of propositional logic concerning (i) the formalization of conditionals as material conditionals, (ii) the possibility of Modus Tollens, (ii) its monotonicity, (iii) its incapacity to formalize modal statement, (iv) that from a contradiction, anything can be deduced. Next, in order to overcome the incapacity of classical logic to formalize modal statements, modal logics is briefly presented.

7.3 Modal logics

Classical logic is truth-functional: the truth-value of complex statements depends only of the truth-value of its constitutive statements. However, this approach seems inadequate for many types of sentences. For instance, consider the modal sentence "Guido believes that Mario is not a student". The truth-value of the sentence "Mario is not a student" does not determine the truth-value of "Guido believes that Mario is not a student". Similarly, the truth-value of the modal statement "It is obligatory that Mario cooperates" does not simply depend on the truth-value of the statement "Mario cooperates". The truth-functional nature of classic propositional logic does not seem to be appropriate for such modal statements.

If we want a logic that accounts for such sentences, a remedy is that the truth-value of modal statements does not only depend of the constituting statements but also to some other statements. Modal logics aim at studying modal propositions and the logical relationships that they can bear to one another. Though modal logic was initially developed to deal with modes of necessity and possibility to capture modal sentence of the form "It is necessary that ..." or "It is possible that ...", it was extended to other modes as epistemic, deontic or temporal modes. Hence, modal logics cover a family of logics with similar reasoning schemata and different symbols aiming at capturing such or such modes. Table 7.1 lists the most common modal logics.

In the following, we present briefly the typical semantics used in modal logic systems, namely the possible worlds semantics. Bearing in mind that modal logic can handle predication, we limit our presentation to propositional modal logic for the sake of simplicity.

7.3.1 Possible world semantics

Modal logic was first developed to deal with modes of necessity and possibility. Accordingly, to facilitate the discussion, we consider first alethic logic (the logic of the modes regarding necessity and possibility), and then how the reasoning schemata can be adapted to other kind of modalities. Alethic logic begins by augmenting the classical propositional logic with two unary operators, $\Box$ denoting 'necessity' and $\Diamond$ denoting 'possibility'. Assuming this notation, "necessarily γ" shall be formalized as $\Box\gamma$, while "possibly γ" as $\Diamond\gamma$.

The idea is that the truth-value of modal statements does not only depend of the constituting statements but also to some other statements. To do so, an interpretation of modal propositions, i.e. how modal propositions shall be attributed truth-values,

Modal logic	Symbols	Expressions symbolized
Alethic logic	□	It is necessary that ..
	◇	It is possible that ..
Deontic logic	O	It is obligatory that ..
	P	It is permitted that ..
	F	It is forbidden that ..
Temporal logic	G	It will always be the case that ..
	F	It will be the case that ..
	H	It has always been the case that ..
	P	It was the case that..
Doxastic logic	B	It is believed that ..
Epistemic logic	B	It is believed that ..
	K	It is known that ..

Table 7.1. Some well-known modal logics.

is based on the idea of *possible worlds* that are in relations by means of so-called *accessibility relations*. We shall see later that the properties of these accessibility relations permit to determine the truth-value of propositions and hence, the properties of the studied logic.

Given a set W of possible worlds, a valuation v is introduced to give a truth-value to each proposition for each of the possible worlds in W. This means that the value assigned to γ at world w may differ from the value assigned to γ for another world w'. The truth-value of a sentence γ at world w given by the valuation v shall be written $v(\gamma, w)$.

In this setting, the truth-values true and false of complex sentences at world $w \in W$ given a valuation v is defined by the following clauses:

$$v(\sim\gamma, w) = \text{true} \quad iff \quad v(\gamma, w) = \text{true}, \tag{7.7}$$

$$v(\gamma \supset \beta, w) = \text{true} \quad iff \quad v(\gamma, w) = F \ or \ v(\beta, w) = \text{true}, \tag{7.8}$$

$$v(\Box\gamma, w) = \text{true} \quad iff \quad \forall w' \in W,\ if\ wRw',\ then\ v(\gamma, w') = \text{true}, \tag{7.9}$$

$$v(\Diamond\gamma, w) = \text{true} \quad iff \quad \exists w' \in W, wRw',\ v(\gamma, w') = \text{true}. \tag{7.10}$$

The clause in (7.9) is at the heart of alethic logic and captures the notion of necessity. From it, we can derive the following schema named (K):

$$\Box(\gamma \supset \beta) \supset (\Box\gamma \supset \Box\beta). \tag{7.11}$$

The clauses in (7.10) and (7.9) involves that the operator $\Box$ can be defined from $\Diamond$ by noting $\Box\gamma \equiv \neg\Diamond\neg\gamma$.

As an illustration of the use of the clauses given above, conisder the model made of:

- the set of possible worlds $W = \{w_1, w_2, w_3\}$, and
- the set of accessibility relations $R = \{w_1Rw_2,\ w_1Rw_3\}$, and
- the set of valuation $V = \{v(student \supset discount,\ w_1) = \text{true},$
  $v(student,\ w_2) = \text{true},$
  $v(student \supset discount,\ w_2) = \text{true},$
  $v(student,\ w_3) = \text{true},$
  $v(student \supset discount,\ w_3) = \text{true}, \}$

We derive $v(\Box student, w_1) = \text{true}$ and $v(\Box(student \supset discount), w_1) = \text{true}$, that is *student* is necessarily true. From the clause (7.8), we conclude $v(discount,\ w_2) = \text{true}$ and $v(discount,\ w_3) = \text{true}$, and by means of the clause (7.9), $v(\Box discount,\ w_1) = \text{true}$. From this last result, we have $v(\neg\Box student \vee \Box discount,\ w_1) = \text{true}$, that is, $v(\Box student \supset \Box discount,\ w_1) = \text{true}$. By consequence, we obtain $v(\neg\Box(student \supset discount) \vee (\Box student \supset \Box discount),\ w_1) = \text{true}$, that is $v(\Box(student \supset discount) \supset (\Box student \supset \Box discount),\ w_1) = \text{true}$ which verifies well the schema K (7.11).

Some intuitions are not captured by the clauses given in (7.7)-(7.10). For example, in some systems we would like to express that if γ is necessary then γ is 'necessarily necessary', corresponding to the schema $\Box\gamma \supset \Box\Box\gamma$. To account for it, the accessibility relation can be provided with some properties, and for each property of the accessibility relation corresponds some valid and invalid schemata. Some well-known accessibility relations with their corresponding interpretations and axioms are presented in Table 7.2.

Accessibility Relation	Interpretation	Axiom	Axiom Name
	Distributivity	$\Box(\gamma \supset \beta) \supset (\Box\gamma \supset \Box\beta)$	(K)
$\forall u \exists v$, uRv	*Serial*	$\Box\gamma \supset \Diamond\gamma$	(D)
$\forall u$, uRu	*Reflexive*	$\Box\gamma \supset \gamma$	(M)
$\forall u \forall v \forall w$, if uRv and vRw then uRw	*Transitive*	$\Box\gamma \supset \Box\Box\gamma$	(4)
$\forall u \forall v$, if uRv then vRu	*Symmetric*	$\gamma \supset \Box\Diamond\gamma$	(B)
$\forall u \forall v \forall w$, if wRv and wRu then vRu	*Euclidean*	$\Diamond\gamma \supset \Box\Diamond\gamma$	(5)

Table 7.2. Some well-known accessibility relations with their corresponding interpretations and axioms.

For example, in any model such that the accessibility relation is reflexive then the schema $\Box\gamma \supset \gamma$ is valid, that is, in any world of any model $\Box\gamma \supset \gamma$ is true.

The properties of the accessibility allows a systemic study of modal logics. Accordingly, each modal logic is characterized by the properties of the accessibility relation. If a modal logic needs such and such property then the relation property needs such and such property. What axioms and rules must be added to the propositional logic to create a 'correct' system of modal logic is often a matter of philosophical opinion, usually driven by the theorems one wishes to prove. Many modal logics, known collectively as *normal modal logics*, include classically the distribution axiom (K): $\Box(\gamma \supset \beta) \supset (\Box\gamma \rightarrow \Box\beta)$, and the necessitation axiom (N): $\gamma \supset \Box\gamma$. The weakest normal modal logic, named K in honor of S. Kripke, is simply propositional logic augmented by $\Box$, the axioms (K) and (N).

7.3.2 Epistemic logic

J. Hintikka proposed a formal account of reasoning on knowledge and beliefs based on the possible worlds semantics [103] which strongly influenced successive works on the field. In this view, an agent knows γ in w if γ is true in all the worlds w' accessible from w.

If the operator $\Box$ is interpreted as the belief operator (in this case, the symbol $\Box$ is rewritten B), then the corresponding modal logic is the so-called doxastic logic, the logic of belief and disbelief. What axioms must be considered to create a 'correct' doxastic logic is a long on-going debate amongst logicians, philosophers and computer scientists. The most common account for doxastic logic is the so-called KD45, that is the modal logic limited to the axioms (K), (D), (4) and (5). The axiom (D): $\mathrm{B}\gamma \supset \neg\mathrm{B}\neg\gamma$ (bearing in mind that $\Box\gamma \equiv \neg\Diamond\neg\gamma$) indicates that the relation of accessibility is serial, and retains consistency among beliefs. The axiom (4): $\mathrm{B}\gamma \supset \mathrm{BB}\gamma$ indicates that the relation of accessibility is transitive. For example, if we have $v(\mathrm{B}\mathit{student}, w_1) =$ true, then we have also $v(\mathrm{BB}\mathit{student}, w_1) =$ true. Transitivity can be thus interpreted as a mean to capture introspection. The axiom (5): $\neg\mathrm{B}\gamma \supset \mathrm{B}\neg\mathrm{B}\gamma$ is interpreted as negative introspection.

By adding the axiom (M), the accessibility relation becomes reflexive and we obtain the so-called epistemic logic, the modal logic of knowledge and belief. Knowledge is indicated by K. The reflexive property of the accessibility relation implies the schema $\mathrm{K}\gamma \supset \gamma$ which is commonly related to veracity: if I know γ then γ is the case.

Beside the fact that the modeling of knowledge and belief in terms of accessible worlds is far from being intuitive, modal epistemic logic is undermined by some paradoxes. To give a flavor of them, we can indicate the so-called *logical omniscience* problem. Briefly, if an agent knows γ, and if γ entails β then the agent also knows β. A simple example taken from [214] illustrates well the problem here: following J. Hintikka's setting, from the facts that (i) Lois Lane believes that Superman can fly, and since (ii) Clark Kent is Superman, then we can conclude that (iii) Lois Lane believes that Clark Kent can fly. This conclusion is not expected because in the story of Superman, Lois Lane does *not* believe that Clark Kent can fly. Many works (see e.g. [214]) which we cannot address here aim at overcoming this issue.

7.3.3 Deontic logic

If the operator $\Box$ is interpreted as the obligation operator (in this case, the symbol $\Box$ is rewritten O), then we are studying deontic logic, the modal logic of obligation and other related notions as permission, prohibition etc.... Since deontic notions are central to the law, deontic logic is often argued to take a major role as a formal language in legal knowledge representation and reasoning.

Many other properties of the accessibility relation can be analyzed with respect to a deontic setting. The most cited and studied of deontic logic is the so called Standard Deontic Logic (SDL) which is usually axiomatized as a normal modal logic, that is propositional logic augmented with the distribution axiom (K): $\mathrm{O}(\gamma \supset \beta) \supset (\mathrm{O}\gamma \supset \mathrm{O}\beta)$, the necessitation axiom (N): $\gamma \supset \mathrm{O}\gamma$ and the axiom (D): $\mathrm{O}\gamma \supset \neg\mathrm{O}\neg\gamma$ (or equivalently $\mathrm{O}\gamma \supset \mathrm{P}\gamma$). In this view, the axiom (K) considers that if a material conditional is obligatory, and its antecedent is obligatory, then so is its consequent, and justifies the so-called deontic detachment inference pattern: given $\mathrm{O}\gamma$ and $\mathrm{O}(\gamma \supset \beta)$ we derive $\mathrm{O}\beta$. The axiom (N) tells us that if something is a theorem, then it is obligatory. The axiom (D) indicates that what is obligatory is also permitted, and can be rewritten as $\mathrm{O}\gamma \supset \mathrm{P}\gamma$ (the operator $\Diamond$ can be interpreted as the permission operator noted here P). In Standard Deontic Logic, the relation of accessibility is not reflexive, otherwise we would have the undesirable schema $\mathrm{O}\gamma \supset \gamma$, that is, if γ ought to be, then γ is the case. Hence, deontic logic discards the reflexibility relation.

Beside its discutable intuition, Standard Deontic Logic is unfortunately undermined by many paradoxes, or in other words, is not correct. An impressive lists of paradoxes can be found in [138]. To give a flavour of the encountered issues, we can consider one of the most studied, namely the Chilshom's paradox. It consists of the derivation of an inconsistency from four sentences as the followings:

- It ought to be that John goes to assist his neighbors. $\mathrm{O}go$.
- It ought to be that if John goes, then he tells them he is coming. $\mathrm{O}(go \supset tell)$.
- If john doesn't go, then he ought not tell them he is coming. $\neg go \supset \mathrm{O}\neg tell$.
- John does not go. $\neg go$

By deontic detachment, the formulas $\mathrm{O}go$ and $\mathrm{O}(go \supset tell)$ implies $\mathrm{O}tell$ whereas, by factual detachment, the formulas $\neg go$ and $\neg go \supset \mathrm{O}\neg tell$ implies $\mathrm{O}\neg tell$. We have hence deduced $\mathrm{O}tell$ and $\mathrm{O}\neg tell$, that is, an inconsistency. As modal epistemic logic, many scholars proposed some (complex) variants of deontic logic in order to solve the paradoxes (see [206] for a review).

7.4 Non-monotonic logics

In Section 7.1 on the appropriateness of logic to formalize legal reasoning, it has been argued that since law has for object a world which is inherently not fully observable in space and in time, which is dynamic and unpredictable, then the deductive logical approach is deemed to fail. Furthermore, on the assumption that conflicts, which are central to the law, are modeled as a contradiction, classical logics are not appropriate for our purposes since classical logics aim at avoiding inconsistencies and does not permit reasoning with such inconsistencies.

In logical terms, these criticisms on the appropriateness of logic to formalize legal reasoning are related to the *monotonicity*[1] of classical logics, that is, the addition of new premises never invalidates formerly derivable consequences. Monotonicity can characterized formally using the relation $\vdash$ of consequence between a set of premises and a single sentence:

$$\text{If } A \vdash \alpha \text{ and } B \supseteq A \text{ then } B \vdash \alpha$$

where A and B are the sets and α the sentence. While these criticisms on monotonicity are acceptable for classical logics which are monotonic, they are not acceptable on the light of non-monotonic systems in which the addition of new premises can invalidate formerly derivable consequences and in which the reasoning with potential inconsistent information is possible. Non-monotonic logics abandon the monotonicity property of the relation $\vdash$, and adopt a non-monotonic consequence relation denoted $\mid\!\sim$. Non-monotonic logics covers a family of formalisms devised to capture a non-monotonic consequence relation, and some properties, which $\mid\!\sim$ should/can/has to respect, have been provided in the literature. For example, D. Gabbay in [80] argues that any well-behaved non-monotonic consequence relation should respect reflexivity, cut and cautious monotony properties:

- reflexivity: if $\alpha \in A$ then $A \mid\!\sim \alpha$,
- cut: If $A, \beta \mid\!\sim \alpha$ and $A \mid\!\sim \beta$ then $A \mid\!\sim \alpha$,
- cautious monotony: If $A \mid\!\sim \alpha$ and $A \mid\!\sim \beta$, then $A, \beta \mid\!\sim \alpha$.

Other properties can be proposed to characterize the non-monotonic consequence relation, and we can use them to guide the design of a non-monotonic logic satisfying such or such properties of non-monotonicity (see e.g. Kraus et al. 's [120]). This 'top-down' method contrasts with the 'bottom-up' method with which initial non-monotonic logics were designed by proposing first non-monotonic mechanisms and then by studying their non-monotonicity properties. The 'top-down' method cannot be addressed here, and in the following, we present briefly the most studied non-monotonic logics and their relevances for our purposes. The reader can find in [174] another overview of the uses of non-monotonic logics in artificial intelligence and law.

[1] In mathematics, monotonicity refers to a process which from a larger input you can get only a larger output.

7.4.1 Circumscription

Circumscription was proposed by J. McCarthy in [134] and mainly superseded by the same author in [135], to overcome the qualification problem (i.e. which conditions must be fulfilled to get an effective action) by formalizing the idea that things are as expected *unless* otherwise specified, e.g. an action is effective unless something prevents it.

Though circumscription was initially defined in first-order logic, we present first its adaptation to propositional logic in order to help the reader to grasp the basic idea. The idea is to evaluate the truth-value of propositions on the light of a restricted set of interpretations. Suppose that we have the following axioms:

$$student \wedge \neg abnormal \supset discount \tag{7.12}$$

$$student \tag{7.13}$$

Our goal is to find out a mechanism to derive *discount*. Classical propositional logic does not achieve this goal as *discount* is not true in all the interpretations. This is apparent in the last row of Table 7.3.

student	*student* ∧ ¬*abnormal* ⊃ *discount*	*abnormal*	*discount*
1	1	0	1
1	1	1	0

Table 7.3. Truth table.

When we compute the truth-value of the proposition *discount*, we consider the interpretation in which the proposition *abnormal* is true, whereas it is not contained in the axioms. The idea is then to remove any interpretation in which propositions that are not in the axioms takes the value true. In the remaining interpretation, the proposition *discount* is true. Now, suppose that the new information that *abnormal* is true. In this case, using the same mechanism as previously done, then we can draw that *discount* is false. This example illustrates the non-monotonicity of (propositional) circumscription: the formerly derived consequence has changed by adding a premise.

In its original first-order logic formulation, circumscription consists in minimizing the extension of some predicates (the extension of a predicate is the set of tuples of values the predicate is true on). This minimization is similar to the closed world assumption that what is not known to be true is false.

The following presents circumscription as proposed by J. McCarthy in [134]: Let A be a sentence of first-order logic (e.g. the conjunction of the premises of a theory) containing a predicate $P(x_1, \ldots, x_n)$ which we will write $P(\bar{x})$. We write $A(\Phi)$ for the result of replacing all occurrences of P in A by the predicate expression Φ.

Definition 7.8. *The circumscription of P in $A(P)$ is the sentence schema*

$$A(\Phi) \wedge \forall \bar{x}.(\Phi(\bar{x}) \supset P(\bar{x})) \supset \forall \bar{x}.(P(\bar{x}) \supset \Phi(\bar{x}))$$

Circumscription consists in the above sentence schema which can be used along the rules of inference of first-order logic. The schema assumes a predicate parameter Φ that may substituted by an arbitrary predicate expression. The conjunct $A(\Phi)$ asserts that Φ has to satisfy the conditions satisfied by P while the second conjunct $\forall\bar{x}.(\Phi(\bar{x}) \supset P(\bar{x}))$ asserts that the entities satisfying Φ are a subset of those that satisfy P. Since the schema is an implication, if these two previous conjuncts are assumed then we can conclude the sentence on the right $\forall\bar{x}.(P(\bar{x}) \supset \Phi(\bar{x}))$. The conclusion asserts the converse of the second conjunct to indicate that Φ and P have to coincide.

Interestingly, circumscription can be used by minimizing predicates expressing the abnormalities of situations. Minimizing the extension of these predicates permits to reason under the implicit assumption that things are as expected (that is, they are not abnormal), and that this assumption is made only if possible (abnormality can be assumed false only if this is consistent with the facts.)

For example, suppose the statements "a student have a discount if he is not abnormal", "students over 25 are abnormal" and "Mario is a student":

$$\begin{aligned}
&\forall x, \mathit{student}(x) \wedge \neg \mathit{abnormal}(x) \supset \mathit{discount}(x),\\
&\forall x, \mathit{over\_25}(x) \supset \mathit{abnormal}(x),\\
&\mathit{student}(\text{mario}).
\end{aligned} \tag{7.14}$$

In first-order logic, assuming the fact $\mathit{student}(\text{mario})$ we cannot conclude that $\mathit{discount}(\mathit{mario})$ is true since we cannot prove $\neg\mathit{abnormal}(\text{mario})$. To prove $\mathit{discount}(\text{mario})$, the circumscription technique can be used by circumscribing the predicate $\mathit{abnormal}$. For the sake of the example, we use a slight generalization of the previous circumscription schema which permits to circumscribe two predicates simultaneously (say P and Q):

$$\begin{aligned}
&A(\Phi,\Psi) \wedge \forall\bar{x}(\Phi(\bar{x}) \supset P(\bar{x})) \wedge \forall\bar{y}(\Psi(\bar{y}) \supset Q(\bar{y}))\\
&\supset \forall\bar{x}(P(\bar{x}) \supset \Phi(\bar{x})) \wedge \forall\bar{y}(Q(\bar{y}) \supset (\bar{y})).
\end{aligned} \tag{7.15}$$

Accordingly, the circumscription formula $A(\mathit{discount}, \mathit{abnormal})$ is:

$$\begin{aligned}
&A(\Phi,\Psi) \wedge \forall x(\Phi(x) \supset \mathit{discount}(x)) \wedge \forall y(\Psi(y) \supset \mathit{abnormal}(y))\\
&\qquad\qquad \supset \forall x(\mathit{discount}(x) \supset \Phi(x) \wedge \forall y(\mathit{abnormal}(x) \supset \Psi(y)),
\end{aligned} \tag{7.16}$$

which spelled out is:

$$\begin{aligned}
&\forall x \mathit{student}(x) \wedge \neg\Psi(x) \supset \Phi(x)\\
&\wedge \forall x \mathit{over\_25}(x) \supset \Psi(x)\\
&\wedge \mathit{student}(\text{mario})\\
&\forall x(\Phi(x) \supset \mathit{discount}(x)) \wedge \forall y(\Phi(y) \supset \mathit{abnormal}(y))\\
&\qquad\qquad \supset \forall x(\mathit{discount}(x) \supset \Phi(x) \wedge \forall y(\mathit{abnormal}(x) \supset \Phi(y)).
\end{aligned} \tag{7.17}$$

If we now substitute $\Phi(x) \equiv (x = \text{mario})$ and $\Psi(y) \equiv \bot$ into (7.17) then the left side of the implication is true, and we obtain:

$$\forall x(\mathit{discount}(x) \supset (x = \text{mario})). \tag{7.18}$$

which asserts that Mario has a discount. Such treatment of abnormalities can be useful in legal reasoning to capture exceptions.

Beside the fact that the circumscription is not particularly intuitive, many issues undermined the original circumscription and triggered many proposals to dealt with them (e.g. see [79], p. 16). A major issue concerns its non-computability and thus the incompleteness of any proof theory. Accordingly, some 'reductions' have been proposed to provide similar but computable frameworks (e.g. see [79], p. 18).

7.4.2 Default logic

Default logic is a non-monotonic logic proposed by R. Reiter in [176] to formalize reasoning with default assumptions.

A theory in default logic is a pair $\langle D, W \rangle$ where W is a set of formulas to express indisputable facts, and D is a set of default rules of the following form:

$$\frac{\alpha : \beta_1, \beta_2, ..., \beta_n}{\omega}.$$

If the so-called prerequisite α is derived, and each of the so-called justifications β_i is consistent with the derived statements, then the consequence ω is derived. In the original default logic, formulas are initially assumed to be expressed in the language of first-order logic. Defaults with free variables represent the sets of their ground instances over the Herbrand universe. Remark that contraposition cannot be applied to defaults (in classical logics, contraposition implies that the material conditional $student \supset discount$ is equivalent to $\neg discount \supset \neg student$). As we have seen in Section 7.2.1 that contraposition is undesired for defaults, default logic overcomes this issue of contraposition in a radical way: they are not allowed for defaults.

Let us illustrate default logic. The following is a default formalizing that any student, who is assumed to not be over 25, has a discount:

$$D = \{\frac{student(x) : \neg over\_25(x)}{discount(x)}\}.$$

This default is accompanied by the facts that Mario and Guido are students and that Guido is over 25:

$$W = \{student(\text{mario}), student(\text{guido}), over\_25(\text{guido})\}.$$

According to the default rule, Mario has a discount because the precondition $student(\text{mario})$ holds and the justification $over\_25(\text{mario})$ cannot be inconsistent with any already proved statements. On the contrary, we cannot prove that Guido has a discount because the justification $over\_25(\text{guido})$ is inconsistent with the fact $over\_25(\text{guido})$ though the precondition $student(\text{guido})$ holds.

A default rule can be applied to a theory in order to add its conclusion to the theory. Other default rules may then be applied to the resulting theory. When the theory is such that no other default can be applied, the theory is called an *extension* of the initial default theory. R. Reiter defines an extension of a theory using a fixed-point semantics as provided in the following. Given a set S of closed well-formed formulas ω (a well-formed formula is closed if it does not contain any free variable), the set $Th(S)$ is defined by $Th(S) = \{\omega | \omega$ is a closed well-formed formulas and $S \vdash w\}$

Definition 7.9. *Let (D,W) be a closed default theory, S be a set of closed well-formed formulas, and $\Gamma(S)$ be the smallest set satisfying the following three conditions:*

- $W \subseteq \Gamma(S)$,
- $Th(\Gamma(S)) = \Gamma(S)$,
- *if* $\frac{\alpha:\beta_1,\ldots,\beta_n}{\omega} \in D$, $\alpha \in \Gamma(S)$, $\neg\beta_1,\ldots,\neg\beta_n \notin S$ *then* $\omega \in \Gamma(S)$.

A set E satisfying $\Gamma(E) = E$ is an extension of (D,W).

The first condition indicates that facts are contained in the extension. The second condition expresses that the extension has to be deductively closed. Roughly, the third condition specifies that defaults have to be applied as much as possible. R. Reiter gives a somewhat more intuitive characterization of an extension:

Definition 7.10. *Let (D,W) be a closed default theory, E be a set of closed well-formed formulas, and E_i $(i \geq 0)$ be sets of closed wffs defined as follows:*

- $E_0 = W$,
- $E_{i+1} = Th(E_i) \cup \{\omega \,|\, \frac{\alpha:\beta_1,\ldots,\beta_n}{\omega} \in D,\ \alpha \in E_i,\ \neg\beta_i \notin E\}$.

E is an extension of (D,W) iff $E = \bigcup_{i=0}^{\infty} E_i$.

This definition is non-constructive because the extension E appears in the definition of E_{i+1}, and thus makes the derivation of an extension a non trivial task. Importantly, since the default rules may be applied in different order, then different extensions can be built. A famous illustration is the Nixon diamond example which is a default theory with two extensions:

$$\begin{aligned} D &= \{\tfrac{republican(x):\neg pacifist(x)}{\neg pacifist(x)}, \tfrac{quaker(x):Pacifist(x)}{Pacifist(x)}\}, \\ W &= \{republican(\text{nixon}), quaker(\text{nixon})\}. \end{aligned} \tag{7.19}$$

Since Nixon is both a Republican and a Quaker, both defaults can be applied. Applying the first default, we obtain that Nixon is not a pacifist, which makes the second default not applicable. Applying the second default leads to the conclusion that Nixon is a pacifist, which makes the first default not applicable. The Nixon diamond default theory has therefore two extensions, one in which $pacifist(\text{nixon})$ is true, and another in which $pacifist(\text{nixon})$ is false.

This example illustrates well that since several ways to entail a formula exist, some default theories can have several extensions, i.e. different semantics can be formulated. The two most well-known are the skeptical and the credulous semantics.

In the skeptical semantics, a formula is entailed by a default theory if it is entailed by all its extensions. In credulous semantics, a formula is entailed by a default theory if it is entailed by at least one of its extensions. For example, the Nixon diamond example theory has two extensions, one in which Nixon is a pacifist and one in which he is not a pacifist. Consequently, neither $pacifist(\text{nixon})$ nor $\neg pacifist(\text{nixon})$ are skeptically entailed, while both of them are credulously entailed.

Though default logic deals elegantly with non-monotonicity and removes the issue on contraposition, it is undermined by some well-known problems. We present some of them (see [79] p. 45-47 for a more complete list). Firstly, a major drawback of default logic is its computational complexity (see e.g. [50]). Given a default theory $\langle D, W \rangle$ and a sentence β, the amounts of resources required for the execution of an algorithm computing whether for example there is an extension which contains β, do not permit an efficient implementation.

Secondly, the existence of extensions is not guaranteed in default logic. For example, consider a theory consisting of the following default:

$$\frac{\text{true}{:}\neg A}{A}. \tag{7.20}$$

The statement A cannot belong to any extension because the default is inapplicable. However, if a candidate extension E does not contain A then E is not an extension since no default has been applied. In other words, the empty extension is not even guaranteed.

Thirdly, default logic may not account for some intuitively expected results. For example, consider the following theory:

$$\begin{aligned} D &= \{\tfrac{student(x):\neg discount(x)}{discount(x)}, \tfrac{member(x):\neg discount(x)}{discount(x)}\}, \\ W &= \{student(\text{mario}) \vee member(\text{mario})\} \end{aligned} \tag{7.21}$$

intuitively, since Mario is a student or a member, one of the defaults should apply and, hence we would expect to derive $discount(\text{mario})$. In default logic, however, a default is applicable only if its prerequisite is derived, and thus we cannot derive $discount(\text{mario})$.

Fourthly, defaults are provided under a form of inference rules which does not allow for reasoning about them.

Notwithstanding with these problems, default logic is one of the most studied among non-monotonic logics, and many variants or extensions have been proposed. For example, prioritized versions of default logic have been proposed in order to account for preferences between defaults (see e.g. [43]).

7.4.3 Logic programming

Logic programming is the interpretation of first-order predicate logic as a programming language [114]. The idea is to use logic to represent knowledge and the use of deduction to solve problems by deriving logical consequences. R. Kowalski explains in [115]:

> A consequence of using logic to represent knowledge is that such knowledge can be understood *declaratively*. A consequence of using deduction to derive consequences in a computational manner is that the same knowledge can also be understood *procedurally*. Thus logic programming allows us to view the same knowledge both declaratively *and* procedurally.

In the legal domain, the use of logic programming to formalize pieces of legislation was investigated originally by researchers at Imperial College (see e.g. [198, 199, 117]).

Since first-order predicate logic is not decidable in the general case, its use as a programming language requires some constraints on the efficiency of computability. This is achieved by introducing restrictions on the language of formulas. In the original logic programming, a program is a set of restricted formulas called *Horn clauses*. A Horn clause is a clause (a disjunction of literals) with *at most* one positive literal. A literal is an atomic formula (atom) or its negation (referred to as positive and negative literals, respectively). The following is an example of a Horn clause:

$$\neg A_1 \vee \ldots \vee \neg A_m \vee A_0$$

where $A_0, \ldots, A_m$ are *positive* atomic formula (atom). Any variable is assumed to be universally quantified, that is, the universal quantifier is left implicit. A *definite clause* is a clause which contains exactly one positive literal. A definite program is a finite set of definite clauses which are usually written equivalently in the form of an implication:

$$A_0 \leftarrow A_1 \wedge \ldots \wedge A_m \,.$$

This restriction on formulas makes it possible to use a special inference rule called the *SLD-resolution inference* introduced by J. Robinson [180], and deduction is performed by backwards reasoning using this inference rule.

The most well-known implemented logic programming system is probably Prolog (standing originally for *Pro*grammation en *log*igue), and was first implemented in 1972 by Colmerauer and Roussel. As different logic programming formalisms exist, different Prolog languages exist (see [41] for a comprehensive book about Prolog).

In the original logic programming, a program is a set of Horn clauses and consequently express 'positive information'[2]. However, in many common situations, we use 'negative information' to express ourselves. This usually the case when we express that something does not hold, for example, as the statement "Mario does not have a discount". Though it is arguable that in every day scenarios pieces of negative

[2] A Horn clause of the form $\leftarrow A_1 \wedge \ldots \wedge A_m$ is interpreted as a goal statement which asserts the goal of successfully executing the procedures calls A_i.

information are seldom stated explicitly, law is domain in which negation has an important role. Indeed, if conflicts are central to the law and if conflicts are modeled by the contraction between positive and negative information, then the use of negation is particularly relevant for the formalization of legal reasoning. The introduction of negation in logic programming is a field of research of its own right (for a review see e.g. [18]) and only essential points for our purposes will be provided.

Initially, in order to capture a notion of negation without losing the efficiency in implementation, logic programming was extended with so-called *negation by failure* (see e.g. [56]). In this setting, a clause is written as:

$$A_0 \leftarrow A_1 \wedge \ldots \wedge A_m \wedge \sim A_{m+1} \wedge \ldots \wedge \sim A_n$$

where $m \leq n$ and $A_0, \ldots, A_n$ are *positive* atomic formula (atom). The notion of negation as failure can be related to the so-called *closed world assumption* according which something is false if it is not currently proved to be true. To deal with negation by failure, a Horn clause theorem prover is augmented with a special inference rule: this is the negation as failure inference rule whereby $\sim A$ can be inferred from failure to derive A (i.e. if every possible proof of A fails). This can be formalized as:

$$\frac{P \nvdash A}{\sim A} \tag{7.22}$$

where P is definite program and A an atom. In such setting, deduction is usually performed using SLDNF-resolution (i.e. SLD-resolution augmented with negation as failure).

Interestingly for our purpose, when logic programming is extended with negation as failure, it can support non-monotonic reasoning. For example, R. Kowalski in [116] investigates several ways in which negations and exceptions arising in legislation can be represented by negation as failure. For instance, a statement of the form “A student has a discount unless he or she is over 25 years hold” can be expressed by the clause:

$$discount(x) \leftarrow student(x), \sim over\_25(x). \tag{7.23}$$

If our set of facts includes $student$(mario) (but not $over\_25$(mario)) then we can conclude $discount$(mario), whereas if the set of facts includes both $student$(mario) and $over\_25$(mario)) then we cannot conclude anymore $discount$(mario).

Beside the fact that negation as failure may result in looping programs (depending on the inference procedure), a serious problem is that they are unsound: indeed, it can be proved that any proof procedure deriving a negative literal from a definite program is unsound. In fact, the point is that negation as failure is not properly a classical negation $\neg$. Logic programming with classical negation is usually called *extended logic programming* in which programs are sets of clauses written as:

$$L_0 \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \sim L_{m+1} \wedge \ldots \wedge \sim L_n \tag{7.24}$$

where $m \leq n$ and $L_1, \ldots, L_n$ are *literals* (i.e. an atomic formula (atom) or its negation).

M. Gelfond and V. Lifschitz in [83] proposed a so-called stable model semantics: classical negation is eliminated from extended programs by a simple transformation which replace any negated atom $\neg p$ by a new positive atom p^*. A full presentation of extended logic programming is beyond our scope, however, the main result is that extensions of logic programming capturing classical negation do exist.

Another issue of negation as failure concerns the maintenance of knowledge bases. Suppose that a new exception $\sim below\_60(x)$ is added to the clause 7.23 which has to be changed into:

$$discount(x) \leftarrow student(x), \sim over\_25(x), \sim below\_60(x). \tag{7.25}$$

This illustrates the criticism based on the observation that negation as failure does not offer the possibility of adding new *specific* exceptions without having to change the original clauses (see e.g. [170]). A solution is to consider *general* exceptions: for example, the statement of the form “A student has a discount unless he or she is over 25 years hold” can be expressed by the clauses (7.26) and (7.27) :

$$discount(x) \leftarrow student(x), \sim exception_{7.23}(x), \tag{7.26}$$

$$exception_{7.23}(x) \leftarrow over\_25(x), \sim exception_{7.27}(x). \tag{7.27}$$

If a new exception $\sim below\_60(x)$ is added to the statement formalized by the clause 7.26, then it is sufficient to add the following clause;

$$exception_{7.23}(x) \leftarrow below\_60(x), \sim exception_{7.28}(x). \tag{7.28}$$

Though the maintenance of the knowledge base is facilitated using general exceptions, this solution is not totally satisfactory because the ‘network’ of exceptions may be difficult to grasp in large sets of rules. To avoid such issue, a solution is modeling defeasible reasoning by means of rules without negation as failure: instead, defeasible reasoning is modeled by rules and a priority relation among them. Rules are usually labeled in order to ease the expression of the priority expression. Doing so, instead of changing original clauses, we can add new clauses and appropriate the priority relation. For example, to cater for the new exception $\sim below\_60(x)$, instead of replacing the clause (7.23) by the clause (7.25), we can add to the former the clause:

$$r_{7.29}: \quad \neg discount(x) \leftarrow below\_60(x) \tag{7.29}$$

(where $r_{7.29}$ is the label), and the following priority relation:

$$r_{7.23} > r_{7.29} \tag{7.30}$$

where $r_{7.23} > r_{7.29}$ means that the rule labeled by $r_{7.23}$ is superior to the rule labeled by $r_{7.29}$. Some logic programming formalisms with priority relation are compared by Antoniou et al. in [16].

7.4.4 Argumentation

Argumentation is an intuitive paradigm to construct non-monotonic logics. It permits to account for both the defeasibilty of arguments and the handling of conflicts. Over the years, many formal argumentation systems have been proposed (see e.g. [68, 38, 209]) but all share substantially the same basic idea consisting in building arguments in favor or against a statement, selecting "acceptable" ones, and evaluating whether the original statement is accepted or not. Formal argumentation systems seem particularly well adapted to model dialectical processes in legal reasoning and, as consequence legal reasoning is the privileged domain of application and inspiration for argumentation (see e.g. [86, 87, 170, 172]).

An advantage of argumentation is that the reasoning process is modular and made of intuitive steps, thus avoiding the monolithic approach of many traditional non-monotonic logics. In [170], H. Prakken proposes a four-layered view on argumentation:

- The first layer, called the logical layer, defines how single arguments can be built.
- The second layer, called the dialectical layer, focuses on conflicting arguments and defines dialectical status of arguments.
- The third layer, called the procedural layer, regulates the conduct of argumentative dialogues.
- The fourth layer, called the heuristic layer, deals with the strategies in dialogues (within the bounds of the procedural layer).

The reasoning process of the logical and dialectical layer consists usually of the following four steps:

- build arguments in favor or against a statement,
- identify the conflicts among the arguments,
- determine which arguments are acceptable, and
- define the justified statements.

Some arguments systems leave the internal structure of arguments unspecified (see e.g. [68, 209]) whereas some others use a particular logical language to build arguments (see e.g. [172]). Usually, such languages are defined over a set of literals and two kinds of rules: strict and defeasible rules.

Of course, some arguments may conflict if they support contradictory conclusions. A conflict relation is usually analyzed in terms defeating relation. A first argument may defeat a second argument which may not defeat the first. For example, strict arguments (i.e. arguments constructed with facts and strict rules) can defeat defeasible arguments, but the inverse is not true. Since, literals and rules are two building blocks of arguments, it is also common to distinguish rebutting and undercutting. Informally, a first argument rebuts a second argument if the conclusion of the first is contradictory with the conclusion of the second. A first argument undercuts a second if the second uses a rule whose applicability is disputed by the first. In [68], P.M. Dung proposes the following definition of an argumentation framework (slightly adapted for our purpose):

Definition 7.11. *An argumentation framework is a pair $AF = \langle AR, defeat \rangle$ where AR is a set of arguments and $defeat$ is a binary relation on AR, i.e. $defeat \subseteq AR \times AR$.*

We say that an argument A defeats an argument B if and only if $defeat(A,B) \in Def$ (or A defeats B). Comparing arguments by pairs is not enough since a defeating argument can in turn be defeated by other arguments. The next step is about defining some criteria, or a semantics, to determine the acceptability of arguments. Usually, a set of acceptable arguments is called an extension. In [68], P.M. Dung proposed different semantics (i.e. different notion of acceptability). Some semantics determine a unique set of acceptable arguments (i.e. an extension) while some others return several extensions and allowing different status for arguments. The following recalls the semantics proposed by P.M. Dung in [68]:

Definition 7.12. *A set S of arguments is said to be* conflict-free *if there are no arguments $A, B \in S$ such that A defeats B.*

Definition 7.13. *An argument $A \in AR$ is* acceptable *w.r.t. a set S of arguments iff for each argument $B \in AR$: if B defeats A then B is defeated by S.*

- *A conflict-free set of arguments S is admissible iff each argument in S is acceptable w.r.t. S.*
- *An admissible set S of arguments is called a complete extension iff each argument which is acceptable w.r.t. S, belongs to S.*
- *A preferred extension of an argumentation framework AF is a maximal (w.r.t. set inclusion) admissible set of AF.*
- *A conflict-free set of arguments S is called a stable extension iff S attacks each argument which does not belong to S.*
- *The grounded extension is the least (w.r.t. set inclusion) complete extension.*

Informally, the notion of preferred extension corresponds to a credulous semantics of an argumentation framework while the notion of grounded extension corresponds to a skeptical semantics. Justified statements are usually those which are supported by at least one argument in each extension.

An attractive aspect of argumentation is that justified statements can be compiled using dialectical proof procedures similar to human argumentative dialogues about an issue. Such dialectical proof procedures are usually formalized under the form of dialogue-games which take the form of interactions between two or more agents, where each agent makes a move by making some utterance in a common communication language, and according to some pre-defined rules. Moreover, since argumentation can be used for diverse goals, diverse types of dialogues games can be designed (for example for persuasion dialogues, negotiation, deliberation dialogues, inquiry, etc.)

Such dialectical proof procedures are often embedded into a multi-agent context: the procedure involves agents putting forward arguments for and against propositions, together with justifications for the acceptability of these arguments. Argumentation as such can be used to model and guide multi-agent interactions by specifying protocols.

In a procedural setting in which argumentation can be modeled as a game, the exchange of arguments can be analysed with game-theoretical tools, and agents can build argumentative strategies (see e.g. [184, 179]).

Finally, it has been shown that other well-known non-monotonic logics as circumscription, default logic, auto-epistemic Logic and logic programming can be captured by argumentation logics (see e.g. [110, 68, 38]). Doing so, argumentation provides a common intuitive framework to compare and evaluate different non-monotonic logics.

7.5 Time and non-monotonic logics

The following presents briefly two well-studied non-monotonic logics, namely the situation calculus and the event calculus, aiming at capturing dynamic aspects as time and change.

7.5.1 Situation calculus

The originators of the situation calculus, J. McCarty and P. Hayes, provide in [136] the simple idea on which it is based:

> A situation *s* is the complete state of the universe at an instant of time. [...] Since the universe is too large for complete description, we shall never completely describe a situation; we shall only give facts about situations. These facts will be used to deduce further facts about that situation, about future situations and about situations that persons can bring about from that situation.

Situations are constructed by actions from an initial situation and actions are represented syntactically as functions from one situation to the resulting situation.

The situation calculus is based on circumscription as used by J. McCarthy to formalize the implicit assumption of inertia: things do not change unless otherwise specified (resolving thus the so-called frame problem [136]). Unfortunately, S. Hanks and D. McDermott in [100] demonstrated that the solution proposed by J.McCarthy was leading to wrong results in some cases, like in the Yale shooting problem scenario. Many proposals followed to capture correctly the Yale shooting problem and similar issues. The following presents a version amongst them which was introduced by R. Reiter in [177]. The basic elements of R. Reiter's version of situation calculus are:

- the actions which can be performed in the world. A special predicate *Poss* is used to indicate when an action is executable,

- the fluents which describe the state of the world (i.e. the 'properties of the world'), and
- the situations, each being a finite sequence of actions (i.e. a history of action occurrences).

Remark that in R. Reiter's version of the situation calculus described here, a situation does not represent a state or snapshot, but a history. In this view, a dynamic world is modeled as a series of situations resulting of various actions being performed. Statements whose truth-value may vary from situation to situation are represented by fluents, i.e. predicates taking a situation term as their final argument. For example, $member(\text{mario}, s)$ means that Mario is member in situation s. The situation which exists before the performance of any action is called the initial situation and is typically denoted $s0$. The new situation resulting from the performance of an action is denoted using the function symbol *do* (some other references also use *result*). The function symbol *do* has a situation and an action as arguments, and a new situation as a result. Accordingly, $do(a,s)$ denotes the successor state to s resulting from performing the action a. The description of a dynamic world is formalized by the following axioms:

- axioms about the situations (situation axioms),
- action precondition axioms,
- successor state axioms, and
- foundational axioms.

Each of these axioms are presented below.

Situation axioms. Any situation are specified by means of situation axioms. For example, the following formalizes the fact that Mario is not member in the initial situation s0:

$$\neg member(\text{mario}, \text{s0}).$$

Action precondition axioms. As already pointed out by the qualification problem, some actions may not be executable in a given situation. The possibility of performing an action is expressed by literals of the form $poss(a,s)$, where a is the action, s a situation, and *poss* is a predicate denoting the performance of the action a in the situation s. In the example, the condition to join the association is only possible when one is not member is formulated by:

$$\neg member(x,s) \supset poss(join(x),s).$$

Successor state axioms. Given that an action is possible in a situation, we must specify the effects of that action on the fluents. To do so, effect axioms specify the effect of a given action on the truth-value of a given fluent. These axioms indicating the truth-value of a fluent $F(\overrightarrow{x},s)$ are classified into positive and negative effect axioms:

$$poss(a,s) \wedge \gamma_F^+(\overrightarrow{x},a,s) \supset F(\overrightarrow{x},do(a,s)),$$

$$poss(a,s) \wedge \gamma_F^-(\overrightarrow{x},a,s) \supset \neg F(\overrightarrow{x},do(a,s)).$$

The formula γ_F^+ is intented to express the conditions under which action a in situation s makes the fluent F become true in the successor situation $do(a,s)$, whereas

γ_F^- expresses the conditions under which performing action a in situation s makes fluent F false. For example, if an agent x has paid ($paid(x,s)$) and joins the association ($poss(join(x),s)$) then the agent x is member in the successive situation $do(join(x),s)$:

$$poss(join(x),s) \wedge paid(x,s) \supset member(x,do(join(x),s)).$$

While the above axioms are about the effects of actions, they cannot be used to formalize all the things that remain unchanged when an action is made (i.e. frame problem). In order to solve the frame problem, R. Reiter in [177] proposes the following successor state axiom:

$$poss(a,s) \supset \left[F(\overrightarrow{x},do(a,s)) \equiv \gamma_F^+(\overrightarrow{x},a,s) \vee \left(F(\overrightarrow{x},s) \wedge \neg\gamma_F^-(\overrightarrow{x},a,s)\right)\right].$$

This formula tells us that if it is possible to perform action a in situation s, then the fluent F is true in the resulting situation $do(a,s)$ if and only if performing a in s makes it true, or it is true in situation s and performing a in s does not make it false. In other words, something holds at a situation s if it was initiated by an action a in the last situation or if it held in the last situation and was not terminated by the last actions.

Foundational axioms. The foundational axioms formalize the unique names of actions and situations. R. Reiter proposes (i) the "unique names axioms for actions" according to which for distinct action $a(\overrightarrow{x})$ and $a'(\overrightarrow{y})$ have distinct arguments $a(\overrightarrow{x}) \neq a'(\overrightarrow{y})$; and identical actions have identical arguments $a(\overrightarrow{x}) = a'(\overrightarrow{y}) \supset \overrightarrow{x} = \overrightarrow{y}$, (ii) the "unique names axioms for states" according to which the initial state $s0$ is the result of any action $s0 \neq do(a,s)$, and situations are histories $do(a,s) = do(a',s') \supset a = a' \wedge s = s'$.

Interestingly, it is possible (see e.g. [118]) to write the situation calculus as a logic program using the following clauses:

$$holds(f,results(a,s)) \leftarrow happens(a,s) \wedge initiates(a,f,s),$$

$$holds(f,results(a,s)) \leftarrow happens(a,s) \wedge holds(f,s) \wedge \neg terminates(a,f,s).$$

The predicates *happens*, *initiates* and *terminates* correspond to the predicates *poss*, $\gamma_F^+(\overrightarrow{x},a,s)$, and $\gamma_F^-(\overrightarrow{x},a,s)$ respectively. The first clause defines what sentences hold because they are initiated by an action. The second clause is the frame axiom and defines what properties hold because they held before and no actions terminated it. The negation $\neg$ is interpreted as negation as failure.

The situation calculus relies on the concept of situation instead of an explicit concept of time (as the event calculus). A direct consequence is that actions do no occur at an explicit time instants, but occur in a situation. This is a serious drawback in legal reasoning since statements often refer to an explicit time. For example, there is no original mean to express a statement as "The present law enters in force the 1^{st} November 1999". Many other issues exist. Among them, there is no original mean:

- for representing information about the duration of actions,
- for representing situation that do not result from an action,
- for representing preconditions that must hold beyond the start time of the action, and
- for dealing with simultaneous actions.

Many extensions to the original situation calculus exist to handle these problems, but their integration into one homogeneous formalism is an issue of its own. The difficulty of such integration is augmented by the fact that, for many people, the situation calculus is not particularly intuitive.

7.5.2 Event calculus

The event calculus is a treatment of time initially proposed by R. Kowalski and M. Sergot in [119] within the framework of logic programming so that it results in a formalism which is computable.

The event calculus uses reification techniques to account for times. Reifying a logic involves moving into the meta-language where a formula in the initial language, i.e. the object language, becomes a term in the new language. Usually, the embedding logic is the first-order logic while the reified language is many sorted. For example, one can reason about the particular aspects of the truth of statements of the object language through the use of a truth predicate like $true$ which takes as arguments a formula in the object language and a temporal expression. Doing so, we have formula of the form:

$$true(statement, temporal\,argument)$$

whose intented meaning is that the first argument is true at the time denoted by the temporal argument. For instance, the sentence “Mario is asleep between 2 o’clock and 4 o’clock” can be formalized as $true(asleep(\text{mario}), (\text{2pm}, \text{4pm}))$.

In the event calculus, the primitive concepts are events, properties, instants and intervals to account for scenarios in which events happens at some instants, and initiates or terminates some properties holding on some temporal intervals. Event calculus axioms permit to derive when some properties hold. The predication $happens(e,t)$ is used to express that an event e happens at time t. The predication $initiates(e,p)$ and $terminates(e,p)$ define the effects of an event e on a property p.

To derive when a property holds, we use the predicate $holdsAt$ which is defined on the idea of inertia: a property holds a time $t2$ if an event initiated this property at $t2$, or in a previous time $t1$ if this property has not been terminated between $t1$ and $t2$ (in the event calculus jargon, clipped). M. Shanahan in [200] proposes the following axioms:

$$holdsAt(p,t2) \leftarrow initially(f) \wedge \sim clipped(p,t1,t2). \qquad (7.31)$$

$$holdsAt(p,t2) \leftarrow happens(e,t1) \wedge t1 < t2 \wedge initiates(e,p) \wedge \sim clipped(p,t1,t2). \tag{7.32}$$

$$clipped(p,t1,t2) \equiv \exists e,t\,[happens(e,t) \wedge t1 < t < t2 \wedge terminates(e,p)]. \tag{7.33}$$

For example, the statements “If an agent x joins the association a then the agent x is member” and “If an agent x leaves the association a then the agent x is not member” can be formalized as:

$$initiates(join(x,a), member(x,a)),$$

$$terminates(leave(x,a), member(x,a)).$$

Suppose a scenario in which Mario joined the association Alliance in 2000, and leaved it in 2005. We formalize it by the following statements:

$$happens(join(\text{mario}, alliance), 2000)$$

$$happens(leave(\text{mario}, alliance), 2005)$$

Using the axioms (7.33) and (7.32) of the event calculus, then we can derive, for example, that Mario is a member of Alliance in 2003 (i.e. $holdsAt(member(\text{mario}, alliance), 2003)$ is true) but not in 2006 (i.e. $holdsAt(member(\text{mario}, alliance), 2003)$ is false).

As the event calculus is proposed in the framework of logic programming, it results that it can be implemented in Prolog. For example, the basic event calculus can be implemented by the following domain independent clauses:

```
holdsAt(Fluent, T) :-
initially(Fluent),
not(broken(Fluent, 0, T)).

holdsAt(Fluent, T) :-
initiates(Event, Fluent),
happens(Event, EarlyTime),
EarlyTime < T,
not(broken(Fluent, EarlyTime, T)).

broken(Fluent, T1, T3) :-
happens(Event, T2),
T1 =< T2,
T2 < T3,
terminates(Event, Fluent).
```

Due to its relative simplicity, the event calculus is often appreciated by computer scientists. Unfortunately, the axioms (7.31)-(7.33) are not sufficient for their intended roles because they can lead to unexpected results in some scenarios concerned with the frame problem [200]. Many others issues undermined such axiomatics of event calculus and many extensions exist to overcome these issues (see e.g. [140]). For example, in [200], M. Shanahan proposes a variant accounting for indirect effects, actions with non-deterministic effects, concurrent actions, continuous change, etc.... In the legal domain, R. Hernández Marín and G. Sartor proposed in [133] an extension of event calculus to account for some temporal aspects of the legal knowledge.

8

Defeasible logic

In the previous Chapters, we have pointed out the necessity of a non-monotonic logic to formalize interactions between normative systems and cognitive agents. Among the studied non-monotonic formalisms, it turns out that circumscription and default logic have both a high computational complexity, and thus the amounts of resources required to compute the conclusions of some theories do not permit an efficient implementation. Argumentation formalisms provided intuitive account of defeasible reasoning but the associated complexities are still a possible issue. Logic programming with negation as failure seemed to be fulfilled the requirement, but logic programming with priorities was argued to be more appropriate for our purposes.

In the remainder, we commit to a particular non-monotonic logic formalism, namely defeasible logic. Defeasible logic was initially designed by D. Nute [148, 147] and belongs to a family of approaches based on the idea of logic programming allowing non-monotonicity without negation as failure or logic. Other logics in this family include courteous logic programs [97] and LPwNF (Logic Programming without Negation as Failure) [66], but Antoniou et al. in [17, 16] has accredited to defeasible logic a higher expressiveness than these two systems with respect to skeptical reasoning. In [10], its flexibility was advocated by proposing simple ways to tune it, and thus, to generate variants with ambiguity propagation or incapable of team defeat for example. Formal properties of defeasible logic were analyzed [11], a denotational semantics was provided in [129], a model theoretic semantics in [131] as well as an argumentation semantics in [91] which can ease its use in argumentation systems. In [12], close links are established amongst known semantics of logic programs as stable model semantics and Kunen semantics with respect to defeasible logic.

Defeasible logic was designed to be easily implementable right from the beginning, unlike most other approaches, and M. Maher in [130] proved that propositional defeasible logic has a linear complexity. In [132], M. Maher et al. report a query answering system employing a backward-chaining approach (called Deimos) and a forward-chaining implementation which computes all conclusion (called Delores). A prolog meta-interpreter for defeasible logic also exists [12]. More recent implementated systems make possible interactions with the Semantic Web (see e.g.

[22, 9, 23, 21]). Defeasible logic has been advocated for the modeling of regulations [14], business rules [8] and business contracts [88].

We present its language in Section 8.1, provide its proof conditions in 8.2, and an argumentation semantics in 8.3.

8.1 Language

In the following we the use D. Billington' s formulation of defeasible logic [32]. A defeasible logic theory is a structure $D = (F, R, \mathscr{S})$ where F is a finite set of facts, R a finite set of rules, and $\mathscr{S}$ a superiority relation on R.
Facts are indisputable statements, for example, "Mario is a minor," formally written as:

$$minor(\text{mario}).$$

Rules can be strict, defeasible, or defeaters. Strict rules are rules in the classical sense; whenever the premises are indisputable, so is the conclusion. An example of a strict rule is "Minors are persons," formally written as:

$$r_1: \quad minor(x) \rightarrow person(x).$$

Defeasible rules are rules that can be defeated by contrary evidence. An example of a defeasible rule is "Persons have legal capacity"; formally:

$$r_2: \quad person(x) \Rightarrow has\_legal\_capacity(x).$$

Defeaters are rules that cannot be used to draw any conclusion. Their only use is to prevent some conclusions by defeating some defeasible rules. An example of this kind of rule is "Minors might not have legal capacity," formally represented as:

$$r_3: \quad minor(x) \leadsto \neg has\_legal\_capacity(x).$$

The idea here is that even if we know that someone is a minor, this is not sufficient evidence for the conclusion that he or she does not have legal capacity. The superiority relation between rules indicates the relative strength of each rule. That is, stronger rules override the conclusions of weaker rules. For example, if the rule r_3 overrides r_2, formally written:

$$r_3 \succ r_2$$

then we can derive neither the conclusion that Mario has legal capacity nor the conclusion that he does have legal capacity. Note that the superiority relation between certain rules can be left unspecified because in many cases there is no natural way of assigning preferences.

Definition 8.1 (Language). *Let* Prop *be a set of propositional atoms, and* Lab *be a set of labels. The sets below are defined as the smallest sets closed under the following rules:*

Literals

$$\text{Lit} = \text{Prop} \cup \{\neg p | p \in \text{Prop}\}$$

Rules

$$\begin{aligned}
&\text{Rule}_s = \{r{:}\phi_1, \ldots, \phi_n \to \psi | r \in \text{Lab}, \phi_1, \ldots, \phi_n \in \text{Lit}, \psi \in \text{Lit}\}\\
&\text{Rule}_d = \{r{:}\phi_1, \ldots, \phi_n \Rightarrow \psi | r \in \text{Lab}, \phi_1, \ldots, \phi_n \in \text{Lit}, \psi \in \text{Lit}\}\\
&\text{Rule}_{dft} = \{r{:}\phi_1, \ldots, \phi_n \leadsto \psi | r \in \text{Lab}, \phi_1, \ldots, \phi_n \in \text{Lit}, \psi \in \text{Lit}\}\\
&\text{Rule} = \text{Rule}_s \cup \text{Rule}_d \cup \text{Rule}_{dft}
\end{aligned}$$

Superiority relations

$$\text{Sup} = \{s \succ r | s, r \in \text{Lab}\}$$

Definition 8.2 (Defeasible Theory). A defeasible theory is a structure $D = (F, R, \mathscr{S})$ where

- $F \subseteq$ Lit is a finite set of facts,
- $R \subseteq$ Rule is a finite set of rules such that each rule has an unique label,
- $\mathscr{S} \subseteq$ Sup is a set of acyclic superiority relations.

In the following, Greek letters (α, β, etc) shall be used as meta-variables ranging over literals and the letters r, s and w as meta-variables ranging over rule labels. We use some abbreviations: If γ is a literal, $\sim\gamma$ denotes the complementary literal (if γ is a positive literal p then $\sim\gamma$ is $\neg p$; and if γ is $\neg p$, then $\sim\gamma$ is p). $A(r)$ denotes the set $\{\phi_1, \ldots, \phi_n\}$ of *antecedents* of a rule r, and $C(r)$ to denote its *consequent*. The set of rules whose consequent is γ is denoted $R[\gamma]$:

$$R[\gamma] = \{r | r \in R, C(r) = \gamma\}.$$

The set of strict rules whose consequent is γ is denoted $R_s[\gamma]$. The set of defeasible and strict rules whose consequent is γ is denoted $R_{sd}[\gamma]$.

8.2 Proof theory

A conclusion of a theory D is a tagged literal having one of the following forms:

$+\Delta\gamma$ meaning that γ is definitely provable in D.
$-\Delta\gamma$ meaning that γ is not definitely provable in D.
$+\partial\gamma$ meaning that γ is defeasible provable in D.
$-\partial\gamma$ meaning that γ is not defeasible provable in D.

Provability is based on the concept of a derivation (or proof) in D. A derivation is a finite sequence $P = (P(1), .., P(n))$ of tagged literals. The initial part of the sequence P of length i is denoted $P(1..i)$. Each tagged literal or rule satisfies some proof conditions.

A literal γ is definitely derivable (i.e. $+\Delta\gamma$) if (1) it is contained in the set of facts, or (2) it exists a definitely applicable strict rule with consequent γ. Formally:

If $P(i+1) = +\Delta\gamma$ then
(1) $\gamma \in F$, or
(2) $\exists r \in R_s[\gamma], \forall \alpha \in A(r) + \Delta\alpha \in P(1..i)$.

To prove that a literal γ is not definitely provable (i.e. $-\Delta\gamma$), we have to show that any attempt to provide a definite proof fails.

If $P(i+1) = -\Delta\gamma$ then
(1) $\gamma \notin F$, and
(2) $\forall r \in R_s[\gamma], \exists \alpha \in A(r) - \Delta\alpha \in P(1..i)$.

Example 1 *Let us illustrate the definite provability by considering the following theory D_2:*

$F = \{high\_income,\ disaster\ \}$,

$$\begin{aligned} R = \{\ & r_0\text{:} \quad high\_income \Rightarrow tax, \\ & r_1\text{:} \quad high\_income,\ disaster \rightarrow \neg tax, \\ & r_2\text{:} \quad \neg tax \Rightarrow invest\}, \end{aligned}$$

$\mathscr{S} = \{r_0 \succ r_1\}$.

The set of facts contains high_income and disaster, hence we derive:

$$+\Delta high\_income,$$
$$+\Delta disaster.$$

From the strict rule r_1 and the conclusion $+\Delta high\_income$, $+\Delta disaster$, we obtain:

$$+\Delta\neg tax.$$

There is no strict rules for tax, hence we conclude:

$$-\Delta tax.$$

Defeasible provability ($+\partial$) for literals consists of three phases. In the first phase, we put forward a supported reason for the literal that we want to prove. Then in the second phase, we consider all possible attacks against the desired conclusion. Finally in the last phase, we have to counter-attack the attacks considered in the second phase.

If $P(i+1) = +\partial\gamma$ then
(1) $+\Delta\gamma \in P(1..i)$, or
(2) (2.1) $-\Delta{\sim}\gamma \in P(1..i)$, and
(2.2) $\exists r \in R_{sd}[\gamma]$, $\forall \alpha \in A(r), +\partial\alpha \in P(1..i)$, and
(2.3) $\forall s \in R[{\sim}\gamma]$,
(2.3.1) $\exists \alpha \in A(s), -\partial\alpha \in P(1..i)$, or
(2.3.2) $\exists w \in R_{sd}[\gamma]$,
$\forall \alpha \in A(w), +\partial\alpha \in P(1..i)$, and $w \succ s$.

Let us examine the proof condition of the defeasible provability of γ. We have two cases: we show that γ is already definitely provable (1), or we need to argue using the defeasible part of the theory (2). In this second case, to prove γ defeasibly we must show that $\sim\gamma$ is not definitely provable (2.1). We require then there must be a strict or defeasible rule $r \in R$ which can be applied and with head γ (2.2). But now we need to consider possible attacks, i.e., reasoning chains in support of $\sim\gamma$, that is, any rule s which has head $\sim\gamma$ (2.3). Note that here we consider defeaters, too, whereas they could not be used to support the conclusion γ; this is in line with the motivation of defeaters given earlier. These attacking rules s have to be discarded (2.3.1), or must be counterattacked by a stronger and applicable rule w which has a head γ (2.3.2).

Remark that the derivation of $D \vdash +\partial\gamma$ from $D \vdash +\Delta\gamma$ can be somewhat misleading because if we interpret literals tagged by $+\partial$ as defeasible conclusions, then we may also conclude that $+\partial\gamma$ may be latter withdraw on light of further information, which is not the case here.

Example 2 *Let us illustrate the defeasible provability by considering the theory D_2. The set of facts contains high_income and disaster, hence we derive:*

$$+\Delta high\_income,$$
$$+\Delta disaster.$$

From the rule r_1 and $+\Delta high\_income$, $+\Delta disaster$, we obtain:

$$+\Delta\neg tax.$$

From $+\Delta\neg tax$, we can derive:

$$+\partial\neg tax.$$

From the rule r_2 and $+\partial\neg tax$, we conclude:

$$+\partial invest.$$

To prove that a literal is not defeasibly provable we have to show that any attempt to give a proof fails.

If $P(i+1) = -\partial\gamma$ then
(1) $-\Delta\gamma \in P(1..i)$, and
(2) (2.1) $+\Delta\sim\gamma \in P(1..i)$, or
(2.2) $\forall r \in R_{sd}[\gamma]$, $\exists\alpha \in A(r), +\partial\alpha \in P(1..i)$, or
(2.3) $\exists s \in R[\sim\gamma]$,
(2.3.1) $\forall\alpha \in A(s), +\partial\alpha \in P(1..i)$, or
(2.3.2) $\forall w \in R_{sd}[\gamma]$,
$\exists\alpha \in A(w), +\partial\alpha \in P(1..i)$, or $w \not> s$

The inference conditions for negative proof tags are derived from the inference conditions for the corresponding positive proof tag by applying the so-called Principle of Strong Negation (see [13]): the strong negation of a formula is closely related to the function that simplifies a formula by moving all negations to an innermost position in the resulting formula and replace the positive tags with the respective negative tags and vice versa.

Notice that the clause (2.3.2) of the proof condition for $+\partial$ allows to consider rules w that counter-attacks any rule s attacking a supporting rules r for the desired conclusion. This allows to deal with the notion of so-called *team defeat*: if there is a team A consisting of applicable rules with consequent γ, and a team B consisting of applicable rules with consequent $\sim\gamma$, then these teams compete with one another. Team A wins if and only if any rule in team B is overruled by a rule in team A. If the team A wins then we can prove $+\partial\gamma$.

Example 3 *Let us illustrate the notion of team defeat by the following theory D_3.*

$F = \{b\}$,

$$\begin{aligned} R = \{\ r_1 &: \quad b \Rightarrow a, \\ r_2 &: \quad b \Rightarrow a, \\ r_3 &: \quad b \Rightarrow \neg a, \\ r_4 &: \quad b \Rightarrow \neg a\}, \end{aligned}$$

$\mathscr{S} = \{r_1 \succ r_3, r_2 \succ r_4\}$.

The set of facts contains only b, hence we obtain:

$$+\Delta b,$$
$$+\partial b.$$

Let us try to derive $+\partial a$. Consider the proof condition for $+\partial$. The clause (1) is not fulfilled since we cannot derive $+\Delta a$. The clause (2.1) is fulfilled since we derived $-\Delta\neg a$. From the conclusion $+\partial b$ and the rule r_1 or r_2, the clause (2.2) is fulfilled. The rules r_3 and r_4 are also applicable but for each there is a stronger rule (r_1 or r_2) which is applicable and is stronger, hence the clause (2.3) is fulfilled. As a consequence, we derive:

$$+\partial a$$

In [13], G. Antoniou et al. shows that it is easy to define a variant of defeasible that does not include team defeat. To do so, the clause (2.3.2) in the inference condition $+\partial$ as to be changed by the clause "(2.3.2) $r \succ s$". In other words, an attack on rule r by rule s can only be defended by r itself (if r is stronger than s.)

Strict rules are used in two different ways. In the proof conditions for definite provability, strict rules are used as in classical logic: if they can fire they are applied, regardless of any reasoning chains with the opposite conclusion. In the proof condition for defeasible provability, strict rules are used like defeasible rules: a strict rule may be applicable, yet it may not fire because there is a rule with the opposite conclusion that is not weaker. Also, strict rules are not automatically superior to defeasible rules. This treatment of strict rules may look a bit confusing and counterintuitive.

We conclude by the management of cyclic theories in which the derivation of γ participates to trigger a rule with consequent $\sim\gamma$. In the present version of defeasible logic, there is no such idea that deriving $+\partial\gamma$ block the derivation of $+\partial\sim\gamma$.

Example 4 *Let us illustrate the notion of cycle by the following theory D_4.*

$F = \{b\}$,

$$R = \{r_0\colon\ b \Rightarrow a,\\ \quad r_1\colon\ a \Rightarrow \neg a\},$$

$\mathscr{S} = \varnothing$.

The set of facts contain only b, hence we derive:

$$+\Delta b,\\ +\partial b.$$

There is not strict rules with consequent a and ¬a, hence we obtain:

$$-\Delta a,\\ -\Delta\neg a.$$

Let us try to derive $+\partial a$. Consider the proof condition for $+\partial$. The clause (1) is not fulfilled since we cannot derive $+\Delta a$. The clause (2.1) is fulfilled since we derived $-\Delta\neg a$. From the conclusion $+\partial b$ and the rule r_0, the clause (2.2) is fulfilled. However, the r_1 is also applicable, hence the clause (2.3) is not fulfilled. As a consequence, we cannot derive $+\partial a$. If the clauses of the proof condition for $-\partial$, then we derive:

$$-\partial a.$$

Let us try to derive $+\partial\neg a$. Consider the proof condition for $+\partial$. The clasue (1) is not fulfilled since we did not derive $+\Delta\neg a$. The clause (2.1) is fulfilled since we derived $-\Delta a$. From the conclusion $+\partial a$ and the rule r_1, the clause (2.2) is fulfilled. However, the r_0 is also applicable, hence the clause (2.3) is not fulfilled. As a consequence, we cannot derive $+\partial\neg a$. If the clauses of the proof condition for $-\partial$, then we obtain:

$$-\partial\neg a.$$

Now consider the slightly different theory in which the superiority relation is not empty:

$F = \{b\}$,

$$R = \{r_0\colon\ b \Rightarrow a,\\ \quad r_1\colon\ a \Rightarrow \neg a\},$$

$\mathscr{S} = \{r_1 \succ r_0\}$.

The set of facts contains only b, hence we derive:

$$+\Delta b,\\ +\partial b.$$

There is not strict rules with consequent a and $\neg a$, hence we conclude:

$$-\Delta a,$$
$$-\Delta \neg a.$$

Let us try to derive $+\partial a$. Consider the proof condition for $+\partial$. The clause (1) is not fulfilled since we cannot derive $+\Delta a$. The clause (2.1) is fulfilled since we derived $-\Delta\neg a$. From the conclusion $+\partial b$ and the rule r_0, the clause (2.2) is fulfilled. The r_1 is also applicable but r_0 is also applicable and is stronger than r_1, hence the clause (2.3) is fulfilled. As a consequence, we can derive:

$$+\partial a.$$

Let us try to derive $+\partial\neg a$. Consider the proof condition for $+\partial$. The clause (1) is not fulfilled since we did not derive $+\Delta\neg a$. The clause (2.1) is fulfilled since we derived $-\Delta a$. From the conclusion $+\partial a$ and the rule r_1, the clause (2.2) is fulfilled. However, the r_0 is also applicable and is stronger than r_1, hence the clause (2.3) is not fulfilled. As a consequence, we cannot derive $+\partial\neg a$. If the clauses of the proof condition for $-\partial$, then we can derive:

$$-\partial\neg a.$$

Interestingly, G. Antoniou et al. in [11] shed light on the relations between the types of provability $\vdash +\Delta$, $\vdash -\Delta$, $\vdash +\partial$, $\vdash -\partial$ and the complements $\nvdash +\Delta$, $\nvdash -\Delta$, $\nvdash +\partial$, $\nvdash -\partial$ for literals γ and $\neg\gamma$. From the definitions of the proof conditions, it is straightforward to show the following theorem.

Theorem 8.3. *Let D be a defeasible theory. For any literal γ:*

- *if $D \vdash +\Delta\gamma$, then $D \vdash +\partial\gamma$,*
- *if $D \vdash -\partial\gamma$, then $D \vdash -\Delta\gamma$,*
- *we cannot have $D \vdash +\Delta\gamma$ and $D \vdash -\Delta\gamma$,*
- *we cannot have $D \vdash +\partial\gamma$ and $D \vdash -\partial\gamma$,*

For any literal, [11] shows that there are exactly six different possible outcomes of the proof theory, and presents for each outcome a simple theory which achieves this outcome:

- $D \nvdash -\Delta\gamma$ and $D \nvdash +\partial\gamma$, $(p \rightarrow p)$,
- $D \vdash +\partial\gamma$ and $D \nvdash +\Delta\gamma$ and $D \nvdash -\Delta\gamma$, $(\Rightarrow p\ ,\ p \rightarrow p)$,
- $D \vdash +\Delta\gamma$ and $D \vdash +\partial\gamma$, $(\rightarrow p)$,
- $D \vdash +\partial\gamma$ and $D \vdash -\Delta\gamma$, $(\Rightarrow p)$,
- $D \vdash -\Delta\gamma$ and $D \nvdash +\partial\gamma$ and $D \nvdash -\partial\gamma$, $(\ p \Rightarrow p)$,
- $D \vdash -\partial\gamma$ and $D \nvdash +\Delta\gamma$, $(\ \emptyset)$.

It follows that a theory is consistent if we cannot conclude that both a literal γ and its complement $\sim\gamma$ are defeasibly true unless they are both definitely true:

Theorem 8.4. *Let D be an acyclic defeasible theory. For any literal γ, $D \vdash +\Delta\gamma$ and $D \vdash +\Delta{\sim}\gamma$ iff $D \vdash +\partial\gamma$ and $D \vdash +\partial{\sim}\gamma$.*

Further results can be found in [11], among them, a modular transformation of any defeasible theory D enables to build an equivalent theory D' without facts, defeaters, and superiority relations. This transformation is used in to provide an argumentation semantics of defeasible logic. Another argumentation semantics which does not need the transformation is provided in [90]. In the remainder, we propose a slightly adapted argumentation semantics which does not require the transformation as in [11], but which is arguably simpler and reflects better the structure of proof theories than [90].

8.3 Argument semantics

Defeasible logic has been defined by its proof theory so far, but non-monotonic reasoning can be analyzed in terms of arguments: non-monotonicity arises when an argument for a conclusion is defeated by a counter-argument. So a non-monotonic logic can be interpreted in terms of interacting arguments, giving for it an argumentation semantics. We present the argumentation semantics in three steps: firstly we define how arguments can be built, secondly we focus on conflicting arguments and determine dialectical status of arguments, and thirdly we define the justified arguments.

8.3.1 Arguments

The argument layer defines what arguments are. An argument for a conclusion (a literal or a rule) is a proof tree (or monotonic derivation) of that conclusion in defeasible logic. Nodes are labeled by either literals or rules which are tagged by $+\Delta'$ and $+\partial'$. Nodes are connected by compound arrows that correspond to grounded inferences rules. In the remainder, Greek letters (α, β, etc) shall be used as meta-variables ranging over literals or rules according to the context.

Definition 8.5. *An argument is a proof tree such that:*

- *each node is labeled by an assertion γ tagged by ∂' or Δ', and*
- *each leaf node is labeled by $+\Delta'\gamma$ where $\gamma \in F \cup R$, and*
- *each compound arrow connecting nodes corresponds to a grounded inference rule of the following types:*

$$\frac{r \in R_s[\gamma], r \quad is \quad \Delta' - applicable}{+\Delta'\gamma} \tag{8.1}$$

$$\frac{r \in R[\gamma], r \quad is \quad \partial' - applicable}{+\partial'\gamma} \tag{8.2}$$

$$\frac{+\Delta'\gamma}{+\partial'\gamma} \tag{8.3}$$

where

If r is Δ'-applicable *then*
(1) $+\Delta' r$, and
(2) $\forall \alpha \in A(r)$, $+\Delta' \alpha$.

If r is ∂'-applicable *then*
(1) $+\partial' r$, and
(2) $\forall \alpha \in A(r)$, $+\partial' \alpha$.

- *the consequent of a defeater, which is ∂'*-applicable*, is the root node.*

The last condition specifies that a defeater rule may only be used at the top of an argument: no chaining of defeaters is allowed.

Example 5 *To illustrate the definition of argument, consider the theory D_2:*

$F = \{high\_income,\ disaster\ \}$,

$$\begin{array}{lll} R = \{ & r_0: & high\_income \Rightarrow tax, \\ & r_1: & high\_income, disaster \rightarrow \neg tax, \\ & r_2: & \neg tax \Rightarrow invest\}, \end{array}$$

$\mathscr{S} = \{r_0 \succ r_1\}$.

We can build the following arguments:

- *F1*: $[+\Delta' high\_income]$,
- *F2*: $[+\Delta' disaster]$,
- *R0*: $[+\Delta' r_0]$,
- *R1*: $[+\Delta' r_1]$,
- *R2*: $[+\Delta' r_2]$,
- *F1'*: $[[+\Delta' high\_income] + \partial' high\_income]$,
- *F2'*: $[[+\Delta' disaster] + \partial' disaster]$,
- *R0'*: $[[+\Delta' r_0] + \partial' r_0]$,
- *R1'*: $[[+\Delta' r_1] + \partial' r_1]$,
- *R2'*: $[[+\Delta' r_2] + \partial' r_2]$,
- *A*: $[[F1', R0'] + \partial' tax]$,
- *B*: $[[F1, F2, R1] + \Delta' \neg tax]$,
- *B'*: $[[B] + \partial' \neg tax]$,
- *C*: $[[B', R2'] + \partial' invest]$.

Definition 8.6. *A (proper) sub-argument of an argument A is a (proper) sub-tree of the tree associated to A.*

Example 6 *The argument B is a proper sub-argument of argument B'. The argument B' is not a proper sub-argument of B'.*

Definition 8.7. *A tagged assertion ($+\Delta' \gamma$ or $+\partial' \gamma$) is a conclusion of an argument if it labels a node of the argument.*

A more usual alternative would be to regard only the root of an argument as its unique conclusion, but this choice would make the other definitions more complicated. Since conclusions can be differently qualified depending on the rules used, arguments are differentiated as follows:

Definition 8.8. *A supportive argument is a finite argument in which no defeater is applied.*

Definition 8.9. *A strict argument is an argument in which any node is tagged by* $+\Delta'$.

Definition 8.10. *An argument that is not strict is defeasible.*

Example 7 *The argument* B: $[[F1, F2, R1] + \Delta' \neg tax]$ *is a strict and supportive argument.*

8.3.2 Acceptable arguments

The previous Section defined the argument layer and isolated the concept of argument. This Section presents the dialectical layer which is concerned with relations standing amongst arguments. It defines the notion of support, attack and counter-attack, and focuses on the interaction amongst arguments. Firstly, we introduce the notion of support:

Definition 8.11. *A set of arguments S supports a defeasible argument A if any proper sub-argument of A is in S.*

Note that, in our setting, the atomic arguments, constituted of a fact or a rule of the theory, are supported by the empty set.

The conditions determining which argument can attack or counter-attack another argument are defined in the following. In the Section presenting the proof theory, a defeasible conclusion is shown to have a proof condition consisting of three phases. In the first phase, a supporting rule r for the desired conclusion is provided. In the second phase, all possible attacks provided by a rule s against the conclusion are considered. In the third phase, counter-attacks are proposed, that is, the counter-attack consists of a rule w such that w is stronger than s.
So in the proof condition, the relation of attack between the first and second phase is somewhat different of the relation of attack between the second and third phase. To reflect this, we provide the notion of attack and defeat between arguments in the following.

Definition 8.12. *An argument S attacks a defeasible argument R if and only if* $+\#{\sim}\gamma$ *and* $+\partial'\gamma$ *are conclusions of the arguments S and R respectively, where* $\# \in \{\Delta', \partial'\}$.

Example 8 *The argument* B: $[[F1, F2, R1] + \Delta' \neg tax]$ *attacks the argument* A: $[[F1', R0'] + \partial' tax]$.

Definition 8.13. *An argument W defeats a defeasible argument S if and only if*

- *$+\#\gamma$ and $+\partial'{\sim}\gamma$ are conclusions of a rule $w \in R[\gamma]$ and a rule $s \in R[{\sim}\gamma]$ in the arguments W and S respectively, where $\# \in \{\Delta', \partial'\}$, and*
- *if $\# = \partial'$, $w \succ s$.*

Example 9 *The argument A: $[[F1', R0'] + \partial' tax]$ does not defeat the argument B: $[[F1, F2, R1] + \Delta' \neg tax]$ whereas the argument B attacks and defeats the argument A.*

Defeasible reasoning differentiates traditionally between rebuttal and undercutting. We stick to the tradition and define the notion of undercutting as follows:

Definition 8.14. *An argument A undercuts a defeasible argument B if A attacks a proper sub-argument of B.*

In this setting, an argument that is attacked but not undercut is said to be rebutted.

Definition 8.15. *A set of arguments S undercuts a defeasible argument B if there is an argument A supported by S that attacks a proper sub-argument of B.*

Definition 8.16. *A set of arguments S defeats a defeasible argument B if there is an argument A supported by S that defeats B.*

Comparing arguments by pairs is not enough since an attacking argument can in turn be attacked by other arguments. In the following, we will define justified arguments, i.e. arguments that have no viable attacking argument in the discourse, and rejected arguments that are attacked by justified argument. As in many argumentation systems, we base the status justified or rejected of arguments on the concept of acceptability of an argument w.r.t. to set of arguments S. That an argument A is acceptable w.r.t. to set of arguments S means that any attacker against A is defeated by an argument supported by S. We present next a slightly adapted version of P.M. Dung's definition of acceptability [68].

Definition 8.17. *An argument A is acceptable w.r.t. a set of arguments S if and only if either*

(1) A is strict, or
(2) for any argument B attacking A
(2.1) B is undercut by S, or
(2.2) B is defeated by S.

That any argument B, which attacks an acceptable argument A and which is not defeated by S, must be undercut (i.e. a proper sub-argument of B must be attacked, not the conclusion) by a counter-argument C supported by S (i.e. the counter-argument C is possibly not a member of S) aims to provide an ambiguity blocking semantics of the system.

8.3.3 Justified arguments

Based on the concept of acceptability we proceed to define justified arguments and justified literals. That an argument A is justified means that it resists any refutation.

Definition 8.18. *The set of justified arguments in a theory D is $JArgs_D = \bigcup_{i=0}^{\infty} J_{D,i}$ with*

- $J_{D,0} = \varnothing$,
- $J_{D,i+1} = \{A \in Args_D | A \quad is \quad acceptable \quad w.r.t. \quad J_{D,i}\}$,

where $Args_D$ is the set of arguments that can be generated from D.

So, an argument A is acceptable w.r.t. $J_{D,i+1}$ if either A is strict, or any argument B attacking A is undercut by $J_{D,i}$ (i.e. there is an argument C supported by $J_{D,i}$ that attacks a proper sub-argument of B) or counter-attacked by an argument supported by $J_{D,i}$.

Definition 8.19. *A tagged assertion $+\partial'\gamma$ is justified if it is the conclusion of a supportive argument in $JArgs_D$.*

Example 10 *Let us illustrate these definitions by a step-by-step construction of the set of justified arguments of the theory D_2.*

We need to find out any argument that is acceptable w.r.t. $J_{D_2,0}$ which is by definition the empty set. Let us consider the strict arguments. The arguments $F1$, $F2$, $R0$, $R1$, $R2$ and B are strict and thus are acceptable w.r.t $J_{D,0}$. Let us now consider non-strict arguments. The argument $F1'$, $F2'$, $R0'$, $R1'$ and $R2'$ are not attacked by any argument. Hence they are acceptable w.r.t. $J_{D_2,0}$.

The argument A is attacked by the argument B which is neither undercut nor counter-attacked by $J_{D_2,0}$. Hence the argument A is not acceptable w.r.t. $J_{D_2,0}$. Similarly, the arguments B' and C are attacked by the argument A which is neither undercut nor counter-attacked by $J_{D_2,0}$. Hence the arguments B' and C are not acceptable w.r.t. $J_{D_2,0}$.

We have parse any tentative to make acceptable arguments w.r.t. $J_{D_2,0}$, therefore

$$J_{D_2,1} = \{F1, F2, R0, R1, R2, F1', F2', R0', R1', R2', B\}.$$

The next step is to find out any argument that is acceptable w.r.t. $J_{D_2,1}$. The argument A is attacked by the argument B which is neither undercut nor counter-attacked by $J_{D_2,1}$. Hence the argument A is not acceptable w.r.t. $J_{D_2,1}$. The arguments B' and C are attacked by the argument A which is defeated by $J_{D_2,1}$ (consider the argument B). Hence the argument B' and C are acceptable w.r.t. $J_{D_2,1}$. We have parsed any tentative to make acceptable arguments w.r.t. $J_{D_2,1}$, therefore

$$J_{D_2,2} = \{B, C\}$$

It is easy to see that $\forall i > 2$, $J_{D_2,i} = \varnothing$. By definition, the set of justified arguments $JArgs_{D_2}$ is $\bigcup_{i=0}^{+\infty} J_{D_2,i}$, that is,

$$JArgs_{D_2} = \{F1, F2, R0, R1, R2, F1', F2', R0', R1', R2', B, B', C\}.$$

Example 11 *This example illustrates how the argumentation semantics deals with team defeat. Consider the theory D_3 from the example 3. Possible arguments include:*

- *B*: $[+\Delta' b]$,
- *B'*: $[[+\Delta' b] + \partial_{B'} b]$,
- *A1*: $[+\Delta' r_1]$,
- *A2*: $[+\Delta' b]$,
- *A3*: $[+\Delta' b]$,
- *A4*: $[+\Delta' b]$,
- *A1*: $[[+\Delta' r_1]\partial' r_1]$,
- *A2*: $[[+\Delta' r_2]\partial' r_2]$,
- *A3*: $[[+\Delta' r_3]\partial' r_3]$,
- *A4*: $[[+\Delta' r_4]\partial' r_4]$,
- *A1”*: $[[[+\Delta' b]\partial' b], [[\Delta' r_1]\partial' r_1]\partial' a]$,
- *A2”*: $[[[+\Delta' b]\partial' b], [[\Delta' r_2]\partial' r_2]\partial' a]$,
- *A3”*: $[[[+\Delta' b]\partial' b], [[\Delta' r_3]\partial' r_3]\partial' \neg a]$,
- *A4”*: $[[[+\Delta' b]\partial' b], [[\Delta' r_4]\partial' r_4]\partial' \neg a]$.

The strict arguments B, A1, A2, A3, A4 are acceptable w.r.t. $J_{D_3,0}$. The arguments B’, A1’, A2’, A3’, A4’ are not attacked or defeated by any argument and hence are also acceptable with respect to $J_{D_3,0}$. As a consequence we can build the set:

$$J_{D_3,1} = \{B, A1, A2, A3, A4, B', A1', A2', A3', A4'\}.$$

The argument A1” is attacked by the arguments A3” and A4” which are attacked by the arguments A1” and A2” respectively, and the arguments A1” and A2” are supported by the set $J_{D_3,1}$. Hence the argument A1” is acceptable w.r.t. $J_{D_3,1}$. Similarly for A2”.

The argument A3” is attacked by the arguments A1” and A2” which are attacked by the arguments A3” and A4” respectively, but the arguments A1” and A2” are neither undercut nor defeated by J_{D_3}. Hence, the argument A3” is not acceptable w.r.t. $J_{D_3,1}$. Similarly for A4”.

We have parsed any tentative to make acceptable arguments w.r.t. $J_{D_3,1}$, therefore we have:

$$J_{D_3,2} = \{A3'', A4''\}.$$

$\forall i > 2$, $J_{D_3,i} = \varnothing$ *and thus*

$$JArgs_{D_3} = \{B, A1, A2, A3, A4, B', A1', A2', A3', A4', A3'', A4''\}.$$

If a tagged assertion $+\partial'\gamma$ is justified means then it is provable (+∂). However, defeasible logic permits to express when a conclusion is not provable (-∂). Briefly, that a conclusion is not provable means that every possible argument for that conclusion has been refuted. In the following, this notion is captured by assigning the status rejected to arguments that are refuted. Roughly speaking, an argument is rejected if it has a rejected sub-argument or it cannot overcome an attack from a justified argument. Given an argument A, a set S of arguments (to be thought of as arguments that have already been rejected), and a set J of arguments (to be thought of as justified arguments that may be used to support attacks on A), we assume the following definition of the argument A being rejected by S and J:

Definition 8.20. *An argument A is rejected by the sets of arguments S and J when A is not strict and if (i) a proper sub-argument of A is in S or (ii) it is attacked by an argument supported by J.*

Definition 8.21. *The set of rejected arguments in a theory D w.r.t. J is $RArgs_D(J) = \bigcup_{i=0}^{\infty} R_{D,i}$ with*

- $R_{D,0}(J) = \varnothing$,
- $R_{D,i+1}(J) = \{a \in Args_D | a \quad is \quad rejected \quad by \quad R_{D,i}(J) \quad and \quad J\}$.

Definition 8.22. *A tagged assertion $+\partial'\gamma$ is rejected by J if there is no argument in $Args_D - RArgs_D(J)$ that ends with as a supportive rule for $+\partial'\gamma$.*

As shortcut, we say that an argument is rejected if it is rejected w.r.t. $JArgs_D$ and a literal is rejected if it is rejected by $JArgs_D$.

An argumentation semantics with ambiguity blocking can now be provided by characterizing conclusions of defeasible logic in argumentation terms:

Definition 8.23. *Let D be a defeasible theory and γ a literal,*

- *$D \vdash +\Delta\gamma$ iff there is a strict argument supporting $+\Delta'\gamma$ in $Args_D$.*
- *$D \vdash -\Delta\gamma$ iff there is no strict argument supporting $+\Delta'\gamma$ in $Args_D$.*
- *$D \vdash +\partial\gamma$ iff $+\partial'\gamma$ is justified.*
- *$D \vdash -\partial\gamma$ iff $+\partial'\gamma$ is rejected by $Jargs_D$.*

This argumentation semantics is consistent with the proof theories of the presented defeasible logic in the sense that conclusions get similarly tagged. The proof not provided here is similar to the one in [91]. It follows that for any defeasible theory, no argument is both justified and rejected, and thus no literal is both justified and rejected. Eventually, if the set $JArgs_D$ of justified arguments contains two arguments with conflicting conclusions then both arguments are strict. That is, inconsistent conclusions can be reached only when the strict part of the theory is inconsistent.

9

Modal defeasible logic

A simple cognitive model of agents was informally presented in Chapter 5. In this Section, we provide a formalization of the cognitive model in modal defeasible logic.

Modal defeasible logic is an umbrella expression for extensions of defeasible logic with modal operators. D. Nute proposed in [150] a deontic defeasible logic. More recently, G. Governatori and A. Rotolo proposed extensions of defeasible logic to capture combinations of mental attitudes and deontic concepts (e.g. see [93, 60, 59]). An implementation as Prolog meta-program of a variant of modal defeasible logic exits (see [15, 65]).

In the remainder of the Section, we propose a version of defeasible logic accounting for sequences of modalities. In order to ease the comparison with basic defeasible and to highlight similarities and differences, we provide the proof theory and the argumentation semantics in similar way as the previous Chapter on basic defeasible logic.

9.1 Language

As in basic defeasible logic, we keep the distinction between *strict rules*, *defeasible rules*, and *defeaters*. Accordingly, a strict rule is an expression of the form $\phi_1,\ldots,\phi_n \rightarrow \psi$ such that whenever the premises are indisputable so is the conclusion. A defeasible rule is an expression of the form $\phi_1,\ldots,\phi_n \Rightarrow \psi$ whose conclusion can be defeated by contrary evidence. An expression $\phi_1,\ldots,\phi_n \rightsquigarrow \psi$ is a *defeater* used to defeat some defeasible rules by producing evidence to the contrary.

Modal literals can occur in the antecedent of rules or in its consequent. This is a shift from the [93, 60, 59]'s formalism in which, in order to extend defeasible logic with modal operators, new types of rules relative to modal operator were introduced by labeling arrows of the rules by a modal operator. This latter solution leads to distinguishing different modes through which the literals can be derived using rules. Such formalism is most relevant when no sequence of modalities is needed: however, since we do intend to capture sequence of modalities, the formalism, in which new types of rules relative to modal operator are introduced, is discarded in favor of a formalism in which modal literals can occur in the antecedent of rules or in its consequent.

The types of agent are linked to the relations conflict and defeat between modal statements (see Section 5.5). The binary relations conflict and defeat defines which types of statements are in conflict and which are the stronger ones. For example, if we write $\text{conflict}(\text{Obl}_{ag}\gamma, \text{Des}_{ag}{\sim}\gamma)$ this means that obligations holding for the agent *ag* conflict with desires of the agent *ag*, and if we write $\text{defeat}(\text{Obl}_{ag}\gamma, \text{Des}_{ag}{\sim}\gamma)$ this means that by default obligations overrides desires. In this perspective, agent types are meaningful within a non-monotonic setting and are nothing but general strategies to detect and solve conflicts between the different components of the cognitive profiles of agent's deliberation.

Definition 9.1 (Language). *Let* Prop *be a set of propositional atoms,* Ag *a finite set of agents,* Lab *be a set of labels and* $m, k \in \mathbb{N}$*. The sets below are defined as the smallest sets closed under the following rules:*

Modal operators

$$\text{Mod} = \{\text{Hold}_{ag}, \text{Des}_{ag}, \text{Bring}_{ag}, \text{Obl}_{ag}, \text{Perm}_{ag}, \text{Forb}_{ag}, \text{Fac}_{ag} \,|\, ag \in \text{Ag}\}$$

Modal literals

$$\text{MLit} = \{(X_i)_{1..n}\neg^m\gamma \,|\, \gamma \in \text{Prop}, X_i = \neg^k X, X \in \text{Mod}\}$$

Rules

$$\begin{aligned}
&\text{Rule}_s = \{(r\colon \quad \phi_1, \ldots, \phi_n \rightarrow \psi) \,|\, r \in \text{Lab}, \phi_1, \ldots, \phi_n, \psi \in \text{MLit} \cup \text{MRule}\}\\
&\text{Rule}_d = \{(r\colon \quad \phi_1, \ldots, \phi_n \Rightarrow \psi) \,|\, r \in \text{Lab}, \phi_1, \ldots, \phi_n, \psi \in \text{MLit} \cup \text{MRule}\}\\
&\text{Rule}_{dft} = \{(r\colon \quad \phi_1, \ldots, \phi_n \rightsquigarrow \psi) \,|\, r \in \text{Lab}, \phi_1, \ldots, \phi_n, \psi \in \text{MLit} \cup \text{MRule}\}\\
&\text{Rule} = \text{Rule}_s \cup \text{Rule}_d \cup \text{Rule}_{dft}
\end{aligned}$$

Modal rules

$$\begin{aligned}
&\text{MRule}_s = \{(X_i)_{1..n}\neg^m r \,|\, X_i \in \{\neg^k\text{Hold}_{ag} | ag \in \text{Ag}\}, r \in \text{Rule}_s\}\\
&\text{MRule}_d = \{(X_i)_{1..n}\neg^m r \,|\, X_i \in \{\neg^k\text{Hold}_{ag} | ag \in \text{Ag}\}, r \in \text{Rule}_d\}\\
&\text{MRule}_{dft} = \{(X_i)_{1..n}\neg^m r \,|\, X_i \in \{\neg^k\text{Hold}_{ag} | ag \in \text{Ag}\}, r \in \text{Rule}_{dft}\}\\
&\text{MRule} = \text{MRule}_s \cup \text{MRule}_d \cup \text{MRule}_{dft}
\end{aligned}$$

Conflict relations

$$\text{Conflict} = \{\text{conflict}(\gamma, \beta) \,|\, \gamma, \beta \in \text{MLit} \cup \text{MRule}\}$$

Defeat relations

$$\text{Defeat} = \{\text{defeat}(\gamma, \beta) \,|\, \gamma, \beta \in \text{MLit} \cup \text{MRule}\}$$

Superiority relations

$$\text{Sup} = \{(X_i)_{1..n}(s \succ r) \,|\, X_i \in \{\text{Hold}_{ag} | ag \in \text{Ag}\}, s, r \in \text{Lab}\}$$

Definition 9.2 (Defeasible theory). A defeasible theory is a structure $D = (F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$ where

- $F \subseteq \text{Lit} \cup \text{MLit}$ is a finite set of facts,
- $R = R_1 \cup R_2$ is a finite set of rules such that each rule has an unique label, and where:
  - $R_1 \subseteq \text{MRule}$ and,
  - $R_2 = \{r_{\gamma \rightarrow \beta}\colon \quad \gamma \rightarrow \beta | \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\} \cup \{r_{\gamma \Rightarrow \beta}\colon \quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$
- $\mathscr{S} \subseteq \text{Sup}$ is a set of acyclic superiority relations,
- $Ag = Ag_1 \cup \{\text{obj}\}$ is a set of agents such that $\forall a \in Ag_1, a \neq \text{obj}$,
- $\mathscr{C} \subseteq \text{Conflict}$ is a set of conflict relations,
- $\mathscr{D} \subseteq \text{Defeat}$ is a set of defeat relations.

We use some abbreviations. The set of sequences of operators $(X_i)_{1..n}$ which does not contain any negated operator is denoted Seq^+:

$$Seq^+ = \{(X_i)_{1..n} \,|\, X_i \in \text{Mod}\}$$

The set of antecedents $\{\phi_1, \ldots, \phi_n\}$ of a rule is denoted $A(r)$, and its *consequent* is denoted $C(r)$. The set of rules whose consequent is γ is denoted $R[\gamma]$:

$$R[\gamma] = \{r | r \in R, C(r) = \gamma\}.$$

The set of strict rules whose consequent is γ is denoted $R_s[\gamma]$. The set of defeasible and strict rules whose consequent is γ is denoted $R_{sd}[\gamma]$.

9.2 Proof theory

A conclusion of a theory D is a tagged (modal)literal or (modal) rule having one of the following forms:

$+\Delta\gamma$ meaning that γ is definitely provable in D.
$-\Delta\gamma$ meaning that γ is not definitely provable in D.
$+\partial\gamma$ meaning that γ is defeasibly provable in D.
$-\partial\gamma$ meaning that γ is not defeasibly provable in D.

Provability is based on the concept of a derivation (or proof) in D. A derivation is a finite sequence $P = (P(1), .., P(n))$ of tagged (modal) literals or (modal) rules. Each tagged (modal) literal or (modal) rule satisfies some proof conditions, which correspond to inference rules for the four kinds of conclusions we have mentioned above.

Before moving to the conditions governing provability of conclusions, we need to introduce some preliminary notions. Firstly, it is crucial to establish criteria for detecting and solving conflicts between the different components which characterize the cognitive profiles of agent's deliberation, and, above all between mental states and normative provisions. For example, consider the conflicts arising from an obligation and a prohibition:

$$r\colon\quad \text{Hold}_{\text{mario}}a \Rightarrow \text{Obl}_{\text{mario}}c$$
$$s\colon\quad \text{Hold}_{\text{mario}}b \Rightarrow \text{Forb}_{\text{mario}}c$$

If the two rules apply, then they conflict. Accordingly, for any γ we introduce the sets $\mathscr{C}_{\text{onflict}}(\gamma)$ that contains all the expression in conflict with γ. We define two sets of conflict: the set $\mathscr{C}^1_{\text{onflict}}(\gamma)$ which denotes the set of assertions in conflict with γ irrespective to any agent theory, and the set $\mathscr{C}^*_{\text{onflict}}(\gamma)$ which denotes the set of assertions in conflict with γ with respect to an agent theory.

Definition 9.3 (Conflicts[1]). *The set $\mathscr{C}^1_{\text{onflict}}(\gamma)$, which denotes the set of assertions in conflict with γ irrespective to any agent theory, is defined as follows. Let $\sim\gamma \in \mathscr{C}^1_{\text{onflict}}(\gamma)$, we have:*

- $\mathscr{C}^1_{\text{onflict}}(\gamma) \supseteq \{\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\beta'),\, \beta' \in \mathscr{C}^1_{\text{onflict}}(\gamma)\}$,
- $\mathscr{C}^1_{\text{onflict}}(\gamma) \supseteq \{\beta \,|\, \gamma \in \mathscr{C}^1_{\text{onflict}}(\beta)\}$.

For any $\gamma \in \text{MLit} \cup \text{MRule}$,

- $\mathscr{C}^1_{\text{onflict}}(\gamma) = \{\neg\gamma\}$,

For any $X \in \{\text{Hold}_{ag}, \text{Des}_{ag}, \text{Bring}_{ag}\}$, $\gamma \in \text{MLit} \cup \text{MRule}$,

- $\mathscr{C}^1_{\text{onflict}}(X\gamma) \supseteq \{X \sim\gamma\}$,

For any $\gamma \in \text{MLit}$,

- $\mathscr{C}^1_{\text{onflict}}(\text{Obl}_{ag}\gamma) \supseteq \{\neg\text{Perm}_{ag}\gamma,\ \text{Perm}_{ag}{\sim}\gamma\}$,
- $\mathscr{C}^1_{\text{onflict}}(\text{Fac}_{ag}\gamma) \supseteq \{\text{Obl}_{ag}{\sim}\gamma,\ \text{Obl}_{ag}\gamma\}$.

Definition 9.4 (Conflicts*). Let $\mathscr{C}^*_{\text{onflict}}$ a set of conflict relations $\text{conflict}^*(\gamma,\beta)$ defined as follows. Let $\gamma \in \text{MLit} \cup \text{MRule}$, and let a theory $D = (F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$, we have:

- $\text{conflict}^*(\gamma,\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\text{conflict}(\beta,\gamma)$ or $\text{conflict}(\gamma,\beta) \in \mathscr{C}$ or $\text{conflict}^*(\beta,\gamma) \in \mathscr{C}^*_{\text{onflict}}$,
- $\text{conflict}^*(\gamma,\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\exists\beta \in \mathscr{E}_{\text{quivalent}}(\beta')$, $\text{conflict}^*(\gamma,\beta') \in \mathscr{C}^*_{\text{onflict}}$,
- $\text{conflict}^*(\text{Hold}_{\text{obj}}\gamma, \text{Hold}_{\text{obj}}\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\text{conflict}^*(\gamma,\beta) \in \mathscr{C}^*_{\text{onflict}}$,
- $\text{conflict}^*(\gamma,\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\text{conflict}^*(\text{Hold}_{\text{obj}}\gamma, \text{Hold}_{\text{obj}}\beta) \in \mathscr{C}^*_{\text{onflict}}$,
- $\text{conflict}^*(\text{Hold}_{ag}\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\text{Hold}_{ag}\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\text{conflict}^*(\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\beta) \in \mathscr{C}^*_{\text{onflict}}$,
- $\text{conflict}^*(\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\text{conflict}^*(\text{Hold}_{ag}\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\text{Hold}_{ag}\beta) \in \mathscr{C}^*_{\text{onflict}}$.

As abbreviation, we have $\mathscr{C}^*_{\text{onflict}}(\gamma) = \{\beta \,|\, \text{conflict}^*(\gamma,\beta) \in \mathscr{C}^*_{\text{onflict}}\}$.

We define the set $\mathscr{C}_{\text{onflict}}(\gamma) = \mathscr{C}^1_{\text{onflict}}(\gamma) \cup \mathscr{C}^*_{\text{onflict}}(\gamma)$ as the set of assertions in conflict with γ. An illustration of the notion of conflict is provided in the example 12. Next, we define the equivalent assertions.

Definition 9.5 (Equivalences). The set $\mathscr{E}_{\text{quivalent}}(\gamma)$, which denotes the set of assertions equivalent to γ, is defined as follows. For any $\gamma \in \text{MLit} \cup \text{MRule}$, $X \in \text{Mod}$, $n \in \mathbb{N}$ and $\sim\gamma \in \mathscr{C}_{\text{onflict}}(\gamma)$, we have:

- $\mathscr{E}_{\text{quivalent}}(\gamma) = \{\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{E}_{\text{quivalent}}(\gamma) = \{\neg^{2n}\gamma\}$,
- $\mathscr{E}_{\text{quivalent}}(\neg^{n}\gamma) = \{\neg^{n}\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{E}_{\text{quivalent}}(\neg^{n}X\,\gamma) = \{\neg^{n}X\,\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{E}_{\text{quivalent}}(\text{Obl}_{ag}\gamma) = \{\neg\text{Perm}_{ag}{\sim}\gamma, \text{Forb}_{ag}{\sim}\gamma\}$.

An illustration of the notion of conflict is provided in the example 12. Next, we provide by means of the conversion relations which modal expressions can be converted into which modal expressions. Note that we do *not* cater for the notion of *rule conversion* which permits to use rules for a modality X as they were for another modality Y provided certain conditions on the modes of the literals in antecedent of the rules (see Section 5.4.3.) However, mere conversions between modalities exist also abstraction done of the conditions on the modes of the literals in antecedent of the rules. For example, if we derive the obligation to pay taxes, then we can also derive the permission to pay taxes. The set of assertions which can result by conversion from an assertion γ is denoted $\mathscr{C}_{\text{onvert}}(\gamma)$.

Definition 9.6 (Conversions). The set $\mathscr{C}_{\text{onvert}}(\gamma)$, which denotes the set of assertions which can result by conversion from an assertion γ, is defined as follows. For any $X \in \text{Mod}$, $\gamma \in \text{MLit} \cup \text{MRule}$ and $n \in \mathbb{N}$, we have:

- $\mathscr{C}_{\text{onvert}}(\gamma) \supseteq \{\beta \,|\, \beta \in \mathscr{C}_{\text{onvert}}(\beta'), \beta' \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{C}_{\text{onvert}}(\gamma) \supseteq \{\text{Hold}_{\text{obj}}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}((\neg)^{n}\text{Hold}_{ag}\gamma) \supseteq \{\text{Hold}_{ag}(\neg)^{n}\text{Hold}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Hold}_{ag}(\neg)^{n}\text{Hold}_{ag}\gamma) \supseteq \{(\neg)^{n}\text{Hold}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Bring}_{ag}\gamma) \supseteq \{\text{Hold}_{\text{obj}}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Bring}_{ag}\gamma) \supseteq \{\text{Hold}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Obl}_{ag}\gamma) \supseteq \{\text{Perm}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Fac}_{ag}\gamma) \supseteq \{\text{Perm}_{ag}\gamma, \text{Perm}_{ag}{\sim}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\neg^{n}\gamma) \supseteq \{\neg^{n}\beta \,|\, \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$,
- $\mathscr{C}_{\text{onvert}}(\neg^{n}X\gamma) \supseteq \{\neg^{n}X\beta \,|\, \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$.

Finally, based on the sets of defeating relations provided by a theory, we assume the following sets of defeating assertions.

Definition 9.7 (Defeats). Let $\mathscr{D}_{\text{efeat}}$ a set of defeat relations $\text{defeat}^*(\gamma,\beta)$ defined as follows. Let $\gamma \in \text{MLit} \cup \text{MRule}$, and let a theory $D = (F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$ we have:

- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}(\beta,\gamma) \in \mathscr{D}$,
- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\exists\beta \in \mathscr{E}_{\text{quivalent}}(\beta')$, $\text{defeat}^*(\gamma,\beta') \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\exists\gamma \in \mathscr{E}_{\text{quivalent}}(\gamma')$, $\text{defeat}^*(\gamma',\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\text{Hold}_{\text{obj}}\gamma, \text{Hold}_{\text{obj}}\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\text{Hold}_{\text{obj}}\gamma, \text{Hold}_{\text{obj}}\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\text{Hold}_{ag}\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\text{Hold}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\text{Hold}_{ag}\text{Hold}_{ag}\gamma, \text{Hold}_{ag}\text{Hold}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$.

As abbreviation, $\mathscr{D}_{\text{efeat}}(\gamma) = \{\beta \,|\, \text{defeat}^*(\beta,\gamma) \in \mathscr{D}_{\text{efeat}}\}$.

We illustrate the notion given above in the example 12.

Example 12 *Given* defeat($\mathrm{Obl}_{\mathrm{guido}}\gamma$, $\mathrm{Des}_{\mathrm{guido}}{\sim}\gamma$) *in a theory, we have:*

$$\begin{aligned}\mathscr{C}_{\mathrm{onflict}}(\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{guido}}a) \supseteq \{&\neg\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Obl}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{guido}}\neg a,\\ &\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Perm}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{mario}}\mathrm{Perm}_{\mathrm{guido}}\neg a,\\ &\mathrm{Hold}_{\mathrm{mario}}\mathrm{Forb}_{\mathrm{guido}}a\ \}\end{aligned}$$

$$\begin{aligned}\mathscr{C}_{\mathrm{onflict}}(\mathrm{Hold}_{\mathrm{guido}}\mathrm{Obl}_{\mathrm{guido}}a) \supseteq \{&\neg\mathrm{Hold}_{\mathrm{guido}}\mathrm{Obl}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{guido}}\neg\mathrm{Obl}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{guido}}\mathrm{Obl}_{\mathrm{guido}}\neg a,\\ &\mathrm{Hold}_{\mathrm{guido}}\neg\mathrm{Perm}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{guido}}\mathrm{Perm}_{\mathrm{guido}}\neg a,\\ &\mathrm{Hold}_{\mathrm{guido}}\mathrm{Forb}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}\neg a\ \}\end{aligned}$$

$$\begin{aligned}\mathscr{E}_{\mathrm{quivalent}}(\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{guido}}a) \supseteq \{&\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Perm}_{\mathrm{guido}}\neg a,\\ &\mathrm{Hold}_{\mathrm{mario}}\mathrm{Forb}_{\mathrm{guido}}\neg a\}\end{aligned}$$

$$\begin{aligned}\mathscr{C}_{\mathrm{onvert}}(\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{guido}}a) \supseteq \{&\mathrm{Hold}_{\mathrm{mario}}\mathrm{Perm}_{\mathrm{guido}}a,\\ &\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Obl}_{\mathrm{guido}}\neg a,\\ &\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Forb}_{\mathrm{guido}}a\ \}\end{aligned}$$

$$\begin{aligned}\mathscr{D}_{\mathrm{efeat}} \supseteq \{&\mathrm{defeat}^*(\mathrm{Obl}_{\mathrm{guido}}\gamma,\ \mathrm{Des}_{\mathrm{guido}}{\sim}\gamma),\\ &\mathrm{defeat}^*(\mathrm{Hold}_{\mathrm{guido}}\mathrm{Obl}_{\mathrm{guido}}\gamma,\ \mathrm{Hold}_{\mathrm{guido}}\mathrm{Des}_{\mathrm{guido}}{\sim}\gamma),\\ &\mathrm{defeat}^*(\mathrm{Hold}_{\mathrm{obj}}\mathrm{Obl}_{\mathrm{guido}}\gamma,\ \mathrm{Hold}_{\mathrm{obj}}\mathrm{Des}_{\mathrm{guido}}{\sim}\gamma)\ \}\end{aligned}$$

Definition 9.8 (Superiority $\mathscr{S}^*$). Let $\mathscr{S}^*$ a set of superiority relations $(X_i)_{1..n}(w \succ s)$ defined as follows. Let a theory $D = (F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$ we have:

- $(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $(X_i)_{1..n}(w \succ s) \in \mathscr{S}$,
- $\mathrm{Hold}_{\mathrm{obj}}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$,
- $(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $\mathrm{Hold}_{\mathrm{obj}}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$,
- $\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $\mathrm{Hold}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$,
- $\mathrm{Hold}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$.

As abbreviation, $\mathscr{S}^*((X_i)_{1..n}(w \succ s)) = \{(X_i)_{1..n}(w \succ s) \mid (X_i)_{1..n}(w \succ s) \in \mathscr{S}^*\}$.

Definition 9.9 (Superiority $\mathscr{S}_{\mathrm{up}}$). Let $\mathscr{S}_{\mathrm{up}}$ denote the set of tuples $\langle (X_i)_{1..n}\gamma, (Y_i)_{1..n}(w \succ s)\rangle$ where $(X_i)_{1..n}\gamma$ is an assertion, and $(Y_i)_{1..n}(w \succ s) \in \mathscr{S}^*$. We have $\langle (X_i)_{1..n}\gamma, (Y_i)_{1..n}(w \succ s)\rangle \in \mathscr{S}_{\mathrm{up}}$ if and only if $(f^{\succ}(X_i))_{1..n} = (Y_i)_{1..n}$ where $f^{\succ}(\neg^k X_{ag}) = \mathrm{Hold}_{ag}$.

As abbreviation, we have $\langle (X_i)_{1..n}\gamma, (w \succ s)\rangle \in \mathscr{S}_{\mathrm{up}}$ if and only if $\langle (X_i)_{1..n}\gamma, (Y_i)_{1..n}(w \succ s)\rangle \in \mathscr{S}_{\mathrm{up}}$.

In order to lighten the presentation of the proof conditions, we present separately the condition for applicability of rules. If a modal rule $(X_i)_{1..n}r$ is applicable then (1) the rule r has for consequent γ, (2) the (possible empty) sequence of operators $(X_i)_{1..n}$ does not contain a negated operator (i.e.) $(X_i)_{1..n} \in Seq^+$, (3) the modal rule $(X_i)_{1..n}r$ is provable, and (4) all the antecedents are proved appropriately:

If $(X_i)_{1..n}r$ is $\Delta(X_i)_{1..n}\gamma$-applicable then
(1) $r \in R_s[\gamma]$, and
(2) $(X_i)_{1..n} \in Seq^+$, and
(3) $+\Delta(X_i)_{1..n}r \in P(1..i)$, and
(4) $\forall\alpha \in A(r)$, $+\Delta(X_i)_{1..n}\alpha \in P(1..i)$.

If $(X_i)_{1..n}r$ is $\Delta(X_i)_{1..n}\gamma$-discarded then
(1) $r \notin R_s[\gamma]$, or
(2) $(X_i)_{1..n} \notin Seq^+$, or
(3) $-\Delta(X_i)_{1..n}r \in P(1..i)$, or
(4) $\exists\alpha \in A(r)$, $-\Delta(X_i)_{1..n}\alpha \in P(1..i)$.

The conditions for a rule to be ∂-applicable are similar as those for Δ-applicable:

If $(X_i)_{1..n}r$ is $\partial(X_i)_{1..n}\gamma$-applicable then
(1) $r \in R[\gamma]$, and
(2) $(X_i)_{1..n} \in Seq^+$, and
(3) $+\partial(X_i)_{1..n}r \in P(1..i)$, and
(4) $\forall\alpha \in A(r)$, $+\partial(X_i)_{1..n}\alpha \in P(1..i)$.

If $(X_i)_{1..n}r$ is $\partial(X_i)_{1..n}\gamma$-discarded then
(1) $r \notin R[\gamma]$, or
(2) $(X_i)_{1..n} \notin Seq^+$, or
(3) $-\partial(X_i)_{1..n}r \in P(1..i)$, or
(4) $\exists\alpha \in A(r)$, $-\partial(X_i)_{1..n}\alpha \in P(1..i)$.

We are now ready to define the proof theory that is, the inference conditions to derive tagged conclusions from a given theory D. Note that the formalism we have introduced allows us to modal rules, thus we have to admit the possibility that rules are not only given but can be derived. Accordingly we have to give conditions that allow us to derive rules. We begin with the proof conditions to determine whether a modal literal or rule is a definite conclusion of a theory D. Let $\gamma \in \text{MLit} \cup \text{MRule}$,

If $P(i+1) = +\Delta\gamma$ then
(1) $\gamma \in F \cup R$, or
(2) $\exists r \in R_s$, r is $\Delta\gamma$-applicable.

To prove that a modal literal is not definitely provable we have to show that any attempt to give a definite proof fails.

If $P(i+1) = -\Delta\gamma$ then
(1) $\gamma \notin F \cup R$, or
(2) $\forall r \in R_s$, r is $\Delta\gamma-$ discarded.

We now turn our attention to defeasible derivations. Defeasible provability ($+\partial$) for modal literals (or modal rules) consists of three phases. In the first phase, we put forward a supported reason for the conclusion that we want to prove. Then in the second phase, we consider all possible attacks against the desired conclusion. Finally in the last phase, we have to defeat the attacks considered in the second phase.

If $P(i+1) = +\partial\gamma$ then
(1) $+\Delta\gamma \in P(1..i)$, or
(2) (2.1) $\forall\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $-\Delta\beta \in P(1..i)$, and
    (2.2) $\exists r \in R_{sd}$, r is $\partial\gamma -$ applicable, and
    (2.3) $\forall\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $\forall s \in R$,
        (2.3.1) s is $\partial\beta -$ discarded, or
        (2.3.2) $\exists w \in R_{sd}$, w is $\partial\gamma -$ applicable, and
            (2.3.2.1) $\gamma \in \mathscr{D}_{\text{efeat}}(\beta)$, $\langle\gamma, (s \succ w)\rangle \notin \mathscr{S}_{\text{up}}$, or
            (2.3.2.2) $\langle\gamma, (w \succ s)\rangle \in \mathscr{S}_{\text{up}}$.

Let us illustrate the proof condition of the defeasible provability of γ. We have two cases: 1) We show that γ is already definitely provable; or 2) we need to argue using the defeasible part of D. In this second case, to prove γ defeasibly we must show that $\sim\gamma$ is not definitely provable (2). We require then there must be a strict or defeasible rule $r \in R$ which can be applied and with consequent γ (2.1). But now we need to consider possible attacks, i.e., reasoning chains in support of $\sim\gamma$, that is, any rule s from which one can derive $\sim\gamma$. Note that here we consider defeaters, too, whereas they could not be used to support the conclusion $\mathrm{X}\gamma$; this is in line with the motivation of defeaters given earlier. These attacking rules s have to be discarded (2.3.1), or must be defeated by a stronger rule w which has a consequent γ (2.3.2).
To prove that a modal literal is not defeasibly provable we have to show that any attempt to give a proof fails.

If $P(i+1) = -\partial\gamma$ then
(1) $-\Delta\gamma \in P(1..i)$, and
(2) (2.1) $\exists\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $+\Delta\beta \in P(1..i)$, or
    (2.2) $\forall r \in R_{sd}$, r is $\partial\gamma -$ discarded, or
    (2.3) $\exists s \in R$, $\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$,
        (2.3.1) s is $\partial\beta$-applicable, or
        (2.3.2) $\exists w \in R_{sd}$, w is $\partial\gamma -$ applicable, and
            (2.3.2.1) $\gamma \notin \mathscr{D}_{\text{efeat}}(\beta)$, $\langle\gamma or(s \succ w)\rangle \in \mathscr{S}_{\text{up}}$, and
            (2.3.2.2) $\langle\gamma, (w \succ s)\rangle \notin \mathscr{S}_{\text{up}}$.

Example 13 *Let us illustrate the above proof conditions by the following theory* $D_{13} = \{F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D}\}$ *where:*

$F = \{\mathrm{Hold}_{\mathrm{guido}} member(\mathrm{mario}), \mathrm{Hold}_{\mathrm{guido}} distrust(\mathrm{mario})\ \}$,

$R = R_1 \cup R_2$, *where*
$R_1 = \{\mathrm{Hold}_{ag}(r_0:\quad \mathrm{Hold}_{ag} member(x) \Rightarrow \mathrm{Obl}_x cooperate(x)),$
$\quad \mathrm{Hold}_{ag}(r_1:\quad \mathrm{Hold}_{ag} distrust(x) \Rightarrow \mathrm{Des}_x \neg cooperate(x))\},$
$R_2 = \{r_{\gamma\to\beta}:\quad \gamma \to \beta | \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}:\quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\},$

$\mathscr{S} = \emptyset$,

$Ag = \{\text{mario}, \text{guido}, \text{obj}\}$,

$\mathscr{C} = \{\text{conflict}(\text{Hold}_{\text{guido}}\text{Obl}_{\text{mario}}\gamma,\ \text{Hold}_{\text{guido}}\text{Des}_{\text{mario}}{\sim}\gamma)\}$,

$\mathscr{D} = \{\text{defeat}(\text{Hold}_{\text{guido}}\text{Obl}_{\text{mario}}\gamma,\ \text{Hold}_{\text{guido}}\text{Des}_{\text{mario}}{\sim}\gamma)\}$.

This agent mario *is social because, by default, obligations defeats desires. The facts* $\text{Hold}_{\text{mario}}\mathit{member}(\text{mario})$ *and* $\text{Hold}_{\text{mario}}\mathit{distrust}(\text{mario})$ *belong to the theory, hence we derive:*

$$+\Delta\text{Hold}_{\text{guido}}\mathit{member}(\text{mario}),\ +\partial\text{Hold}_{\text{guido}}\mathit{member}(\text{mario}),$$
$$+\Delta\text{Hold}_{\text{guido}}\mathit{distrust}(\text{mario}),\ +\partial\text{Hold}_{\text{guido}}\mathit{distrust}(\text{mario}).$$

The set of rules R_2 *contains the following rule:*

$r_{\text{Hold}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\text{Hold}_{ag}\gamma}$: $\text{Hold}_{ag}\gamma \Rightarrow \text{Hold}_{ag}\text{Hold}_{ag}\gamma$,

which is applicable, hence we obtain:

$$+\Delta\text{Hold}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{member}(\text{mario}),\ +\partial\text{Hold}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{member}(\text{mario}),$$
$$+\Delta\text{Hold}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{distrust}(\text{mario}),\ +\partial\text{Hold}_{\text{guido}}\text{Hold}_{\text{guido}}\mathit{distrust}(\text{mario}).$$

The rules r_0 *and* r_1 *belong to the theory, hence we derive:*

$$+\Delta\text{Hold}_{ag}(r_0\text{:}\quad \text{Hold}_{ag}\mathit{member}(x) \Rightarrow \text{Obl}_x\mathit{cooperate}(x)),$$
$$+\partial\text{Hold}_{ag}(r_0\text{:}\quad \text{Hold}_{ag}\mathit{member}(x) \Rightarrow \text{Obl}_x\mathit{cooperate}(x)),$$
$$+\Delta\text{Hold}_{ag}(r_1\text{:}\quad \text{Hold}_{ag}\mathit{distrust}(x) \Rightarrow \text{Des}_x\neg\mathit{cooperate}(x)),$$
$$+\partial\text{Hold}_{ag}(r_1\text{:}\quad \text{Hold}_{ag}\mathit{distrust}(x) \Rightarrow \text{Des}_x\neg\mathit{cooperate}(x)).$$

The rule r_0 *is* $+\partial\text{Hold}_{\text{guido}}\text{Obl}_{\text{mario}}\mathit{cooperate}(\text{mario})$-applicable.
The rule r_1 *is* $+\partial\text{Hold}_{\text{guido}}\text{Des}_{\text{mario}}\neg\mathit{cooperate}(\text{mario})$-applicable.
The rule r_0 *conflicts with* r_1 *because we have:*

$\text{Hold}_{\text{guido}}\text{Des}_{\text{mario}}\neg\mathit{cooperate}(\text{mario}) \in \mathscr{C}_{\text{onflict}}(\text{Hold}_{\text{guido}}\text{Obl}_{\text{mario}}\mathit{cooperate}(\text{mario}))$.

However, the rule r_0 *defeats* r_1 *because we have:*

$\text{Hold}_{\text{guido}}\text{Obl}_{\text{mario}}\mathit{cooperate}(\text{mario}) \in \mathscr{D}_{\text{efeat}}(\text{Hold}_{\text{guido}}\text{Des}_{\text{mario}}\neg\mathit{cooperate}(\text{mario}))$.

Hence, we conclude:

$$-\partial\text{Hold}_{\text{guido}}\text{Des}_{\text{mario}}\neg\mathit{cooperate}(\text{mario}),$$
$$+\partial\text{Hold}_{\text{guido}}\text{Obl}_{\text{mario}}\mathit{cooperate}(\text{mario}).$$

Furthermore, the set of rules R_2 *contains the following rules:*

$r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\text{Forb}_{ag}\sim\gamma}$: $\text{Hold}_{ag}\text{Obl}_{ag}\gamma \rightarrow \text{Hold}_{ag}\text{Forb}_{ag}{\sim}\gamma$,
$r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\text{Perm}_{ag}\gamma}$: $\text{Hold}_{ag}\text{Obl}_{ag}\gamma \Rightarrow \text{Hold}_{ag}\text{Perm}_{ag}\gamma$.

The above rules are both applicable, hence we can conclude among others:

$$+\partial\text{Hold}_{\text{guido}}\text{Forb}_{\text{mario}}\neg\mathit{cooperate}(\text{mario}),$$
$$+\partial\text{Hold}_{\text{guido}}\text{Perm}_{\text{mario}}\mathit{cooperate}(\text{mario}).$$

Example 14 *Suppose the following theory* $D_{14} = \{F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D}\}$ *where:*

$F = \{\mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}(\mathrm{mario}),\ \mathrm{Hold}_{\mathrm{mario}}\mathit{strike\_by\_disaster}(\mathrm{mario})\ \}$,

$R = R_1 \cup R_2$, *where*
$R_1 = \{\mathrm{Hold}_{\mathrm{mario}}(r_0\colon\quad \mathrm{Hold}_{ag}\mathit{has\_high\_income}(x) \Rightarrow \mathrm{Obl}_x\mathit{pay\_tax}(x)),$
$\quad\quad \mathrm{Hold}_{\mathrm{mario}}(r_1\colon\quad \mathrm{Hold}_{ag}\mathit{has\_high\_income}(x), \mathrm{Hold}_{ag}\mathit{strike\_by\_disaster}(x) \Rightarrow \mathrm{Perm}_x\neg\mathit{pay\_tax}(x)),$
$\quad\quad \mathrm{Hold}_{\mathrm{mario}}(r_2\colon\quad \mathrm{Perm}_{\mathrm{mario}}\neg\mathit{pay\_tax}(x) \Rightarrow \mathrm{Des}_{\mathrm{mario}}\mathit{invest}(x)),$
$\quad\quad \mathrm{Hold}_{\mathrm{mario}}(r_3\colon\quad \mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}(x), \mathrm{Des}_{\mathrm{mario}}\mathit{invest}(x) \Rightarrow \mathrm{Bring}_{\mathrm{mario}}\mathit{invest}(x))\},$
$R_2 = \{r_{\gamma\to\beta}\colon\quad \gamma \to \beta | \beta \in \mathscr{E}_{\mathrm{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}\colon\quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\mathrm{onvert}}(\gamma)\}$

$\mathscr{S} = \{\mathrm{Hold}_{\mathrm{mario}}(r_1 \succ r_0)\}$,

$Ag = \{\mathrm{mario}, \mathrm{obj}\}$,

$\mathscr{C} = \{\mathrm{conflict}(\mathrm{Obl}_{\mathrm{mario}}\gamma, \mathrm{Des}_{\mathrm{mario}}{\sim}\gamma),\ \mathrm{conflict}(\mathrm{Obl}_{\mathrm{mario}}\gamma, \mathrm{Bring}_{\mathrm{mario}}{\sim}\gamma)\}$,

$\mathscr{D} = \{\mathrm{defeat}(\mathrm{Obl}_{\mathrm{mario}}\gamma, \mathrm{Des}_{\mathrm{mario}}{\sim}\gamma),\ \mathrm{defeat}(\mathrm{Obl}_{\mathrm{mario}}\gamma, \mathrm{Bring}_{\mathrm{mario}}{\sim}\gamma)\}$.

From the fact $\mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}$*, the rule* r_0 *is applicable, but the putative conclusion* $\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{mario}}\mathit{pay\_tax}$ *is defeated by the application of the rule* r_1 *which is stronger. Hence we can derive:*

$$+\partial\mathrm{Hold}_{\mathrm{mario}}\mathrm{Perm}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario}).$$

From the conclusion $+\partial\mathrm{Hold}_{\mathrm{mario}}\mathrm{Perm}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario})$ *and the rule* r_2*, we obtain:*

$$+\partial\mathrm{Hold}_{\mathrm{mario}}\mathrm{Des}_{\mathrm{mario}}\mathit{invest}(\mathrm{mario}).$$

From the previous result, the fact $\mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}$ *and the rule* r_3*, we derive:*

$$+\partial\mathrm{Hold}_{\mathrm{mario}}\mathrm{Bring}_{\mathrm{mario}}\mathit{invest}(\mathrm{mario}).$$

Finally, the set R_2 *includes the following rule:*

$$R_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma}\colon\quad \mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma \Rightarrow \mathrm{Hold}_{ag}\mathrm{Hold}_{ag}\gamma$$

which is applicable and not attacked by any other rule, so we have:

$$+\partial\mathrm{Hold}_{\mathrm{mario}}\mathrm{Hold}_{\mathrm{mario}}\mathit{invest}(\mathrm{mario}).$$

In words, Mario believes that his action is successful.

Example 15 *Suppose the following theory* $D_{15} = \{F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D}\}$ *where:*

$F = \{\mathrm{Hold}_{\mathrm{obj}}\mathrm{Hold}_{\mathrm{mario}}a,\ \mathrm{Hold}_{\mathrm{obj}}\mathrm{Hold}_{\mathrm{mario}}b\ \}$,

$R = R_1 \cup R_2$, *where*
$R_1 = \{\mathrm{Hold}_{\mathrm{obj}}(r_0\colon\quad \mathrm{Hold}_{ag}a \Rightarrow \mathrm{Obl}_{ag}c),$
$\quad\quad \mathrm{Hold}_{\mathrm{obj}}(r_1\colon\quad \mathrm{Hold}_{ag}b \Rightarrow \mathrm{Forb}_{ag}c),$
$R_2 = \{r_{\gamma\to\beta}\colon\quad \gamma \to \beta | \beta \in \mathscr{E}_{\mathrm{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}\colon\quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\mathrm{onvert}}(\gamma)\}$

$>= \{\mathrm{Hold}_{\mathrm{mario}}(r_0 \succ r_1)\}$,

$Ag = \{\mathrm{mario}, \mathrm{obj}\}$,

$\mathscr{C} = \emptyset$,

$\mathscr{D} = \emptyset$.

The rules r_0 and r_1 belongs to the theory, hence we can derive:

$$+\partial\mathrm{Hold}_{\mathrm{obj}}(r_0\colon\quad \mathrm{Hold}_{ag}a \Rightarrow \mathrm{Obl}_{ag}c),$$
$$+\partial\mathrm{Hold}_{\mathrm{obj}}(r_1\colon\quad \mathrm{Hold}_{ag}b \Rightarrow \mathrm{Forb}_{ag}c),$$

The rules r_0 and r_1 are both applicable but conflict (because $\mathrm{Hold}_{\mathrm{obj}}\mathrm{Forb}_{\mathrm{mario}}c \in \mathscr{C}_{\mathrm{onflict}}(\mathrm{Hold}_{\mathrm{obj}}\mathrm{Obl}_{\mathrm{mario}}c)$). The rule r_0 is stronger than the rule r_1, hence we obtain:

$$+\partial\mathrm{Hold}_{\mathrm{obj}}\mathrm{Obl}_{\mathrm{mario}}c,\ +\partial\mathrm{Hold}_{\mathrm{obj}}\mathrm{Perm}_{\mathrm{mario}}c \textit{ and } -\partial\mathrm{Hold}_{\mathrm{obj}}\mathrm{Forb}_{\mathrm{mario}}c.$$

Example 16 *Suppose the following theory $D_{16} = \{F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D}\}$:*

$F = \{\mathrm{Hold}_{ag}a,\ \mathrm{Hold}_{\mathrm{mario}}b\ \}$,

$R = R_1 \cup R_2$, *where*
$R_1 = \{(r_0\colon\quad \mathrm{Hold}_{ag}a \Rightarrow \mathrm{Hold}_{ag}\neg c,$
$\quad (r_1\colon\quad \mathrm{Hold}_{ag}b \Rightarrow \mathrm{Bring}_{ag}c),$
$R_2 = \{r_{\gamma\to\beta}\colon\quad \gamma \to \beta | \beta \in \mathscr{E}_{\mathrm{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}\colon\quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\mathrm{onvert}}(\gamma)\}$

$>= \{\mathrm{Hold}_{\mathrm{mario}}(r_0 \succ r_{\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma})\}$,

$Ag = \{\mathrm{mario}, \mathrm{obj}\}$,

$\mathscr{C} = \emptyset$,

$\mathscr{D} = \emptyset$.

The rules r_0 and r_1 belongs to the theory, hence we derive:

$$+\partial(r_0\colon\quad \mathrm{Hold}_{ag}a \Rightarrow \mathrm{Hold}_{ag}\neg c),$$
$$+\partial(r_1\colon\quad \mathrm{Hold}_{ag}b \Rightarrow \mathrm{Bring}_{ag}c),$$

*The rule r_1 is $+\partial\mathrm{Bring}_{\mathrm{mario}}c-$*applicable*, and no counter-rule exists, hence we have:*

$$+\partial\mathrm{Bring}_{\mathrm{mario}}c$$

Furthermore, the set of rules R_2 contains the rule:

$$r_{\mathrm{Bring}_{ag}\gamma\Rightarrow\gamma}\colon\quad \mathrm{Bring}_{ag}\gamma \to \mathrm{Hold}_{ag}\gamma$$

*which is $+\partial\mathrm{Hold}_{\mathrm{mario}}c-$*applicable*. However, this latter rule conflicts with the rule r_0 which is $+\partial\mathrm{Hold}_{\mathrm{mario}}\neg c-$*applicable*. The rule r_0 defeats $r_{\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}$, hence we obtain:*

$$+\partial\mathrm{Bring}_{\mathrm{mario}}c,\ +\partial\mathrm{Hold}_{\mathrm{mario}}\neg c \textit{ and } -\partial\mathrm{Hold}_{\mathrm{mario}}c.$$

that is, the attempt $\mathrm{Bring}_{\mathrm{mario}}c$ is unsuccessful. If the superiority relation $>= \{r_0 \succ r_{\mathrm{Bring}_{ag}\gamma\Rightarrow\gamma}\}$ is replaced by $>= \{r_{\mathrm{Bring}_{ag}\gamma\Rightarrow\gamma} \succ r_0\}$ then we conclude:

$$+\partial\mathrm{Bring}_{\mathrm{mario}}c,\ -\partial\mathrm{Hold}_{\mathrm{mario}}\neg c \textit{ and } +\partial\mathrm{Hold}_{\mathrm{mario}}c,$$

and in this case, the attempt $\mathrm{Bring}_{\mathrm{mario}}c$ is successful.

9.3 Argumentation semantics

As for basic non-modal defeasible logic, modal defeasible logic can be analyzed in terms of interacting arguments, giving for it an argumentation semantics. As a matter of fact, the argumentation semantics provided in Section 8.3 for non-modal defeasible logic needs to be only slightly adapted to fit modal defeasible logic: in order to help the reader, the remainder of this Section is thus a slight adaption of Section 8.3.

9.3.1 Arguments

The argument layer defines what arguments are. An argument for a conclusion (a literal or a rule) is a proof tree (or monotonic derivation) of that conclusion in modal defeasible logic. Nodes are labeled by either modal literals or rules which are tagged by Δ' or ∂' in order to keep trace of the strength of assertions. Nodes are connected by arrows that correspond to grounded inferences rules.

Definition 9.10. *An argument is a proof tree such that:*

- *each node is labeled by a modal literal or rule tagged by Δ' or ∂', and*
- *each leaf mode is labeled by $+\Delta'\gamma$ where $\gamma \in F \cup R$, and*
- *each compound arrow connecting nodes corresponds to a grounded inference rule of the following types:*

$$\frac{r \in R,\ r \quad is \quad \Delta'\gamma - \text{applicable}}{+\Delta'\gamma} \tag{9.1}$$

$$\frac{r \in R,\ r \quad is \quad \partial'\gamma - \text{applicable}}{+\partial'\gamma} \tag{9.2}$$

$$\frac{+\Delta'\gamma}{+\partial'\gamma} \tag{9.3}$$

where

*If $(X_i)_{1..n}r$ is $\Delta'(X_i)_{1..n}\gamma$-*applicable *then*
(1) $r \in R_s[\gamma]$, and
(2) $(X_i)_{1..n} \in Seq^+$, and
(3) $+\Delta'(X_i)_{1..n}r$, and
(4) $\forall\alpha \in A(r)$, $+\Delta'(X_i)_{1..n}\alpha$.

*If $(X_i)_{1..n}r$ is $\partial'(X_i)_{1..n}\gamma$-*applicable *then*
(1) $r \in R[\gamma]$, and
(2) $(X_i)_{1..n} \in Seq^+$, and
(3) $+\partial'(X_i)_{1..n}r$, and
(4) $\forall\alpha \in A(r)$, $+\partial'(X_i)_{1..n}\alpha$.

- *If the rule r in the inference (9.2) is a defeater then the post-condition $+\partial'\gamma$ is the root node.*

The last condition specifies that a defeater rule may only be used at the top of an argument; in particular, no chaining of defeaters is allowed.

Example 17 *Suppose the following theory $D_{14} = \{F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D}\}$:*

$F = \{\text{Hold}_{\text{mario}}\mathit{has\_high\_income}(\text{mario}),\ \text{Hold}_{\text{mario}}\mathit{strike\_by\_disaster}(\text{mario})\ \}$,

$$\begin{aligned}
&R = R_1 \cup R_2, \textit{ where}\\
&R_1 = \{\text{Hold}_{\text{mario}}(r_0\colon\quad \mathit{has\_high\_income}(x) \Rightarrow \text{Obl}_x \mathit{pay\_tax}(x)),\\
&\qquad\ \ \text{Hold}_{\text{mario}}(r_1\colon\quad \mathit{has\_high\_income}(x), \mathit{strike\_by\_disaster} \Rightarrow \text{Perm}_x\neg\mathit{pay\_tax}(x)),\\
&\qquad\ \ \text{Hold}_{\text{mario}}(r_2\colon\quad \text{Perm}_{\text{mario}}\neg\mathit{pay\_tax}(x) \Rightarrow \text{Des}_{\text{mario}}\mathit{invest}(x)),\\
&\qquad\ \ \text{Hold}_{\text{mario}}(r_3\colon\quad \mathit{has\_high\_income}(x),\ \text{Des}_{\text{mario}}\mathit{invest}(x) \Rightarrow \text{Bring}_{\text{mario}}\mathit{invest}(x))\},\\
&R_2 = \{r_{\gamma\rightarrow\beta}\colon\quad \gamma \rightarrow \beta | \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}\colon\quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}
\end{aligned}$$

$\mathscr{S} = \{\text{Hold}_{\text{mario}}(r_1 \succ r_0)\}$,

$Ag = \{\text{mario}, \text{obj}\}$,

$\mathscr{C} = \{\text{conflict}(\text{Obl}_{\text{mario}}\gamma, \text{Des}_{\text{mario}}{\sim}\gamma),\ \text{conflict}(\text{Obl}_{\text{mario}}\gamma, \text{Bring}_{ag}{\sim}\gamma)\}$,

$\mathscr{D} = \{\text{defeat}(\text{Obl}_{\text{mario}}\gamma, \text{Des}_{\text{mario}}{\sim}\gamma),\ \text{defeat}(\text{Obl}_{\text{mario}}\gamma, \text{Bring}_{\text{mario}}{\sim}\gamma)\}$.

We can build the following arguments:

- *F1*: $[+\Delta'\text{Hold}_{\text{mario}}\mathit{has\_high\_income}(\text{mario})]$,
- *F2*: $[+\Delta'\text{Hold}_{\text{mario}}\mathit{strike\_by\_disaster}(\text{mario})]$,
- *R0*: $[+\Delta'\text{Hold}_{\text{mario}}r_0]$,
- *R1*: $[+\Delta'\text{Hold}_{\text{mario}}r_1]$,
- *R2*: $[+\Delta'\text{Hold}_{\text{mario}}r_2]$,
- *R3*: $[+\Delta'\text{Hold}_{\text{mario}}r_3]$,
- $R_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\text{Forb}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\text{Forb}_{ag}{\sim}\gamma}]$,
- $R_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Perm}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Perm}_{ag}{\sim}\gamma}]$,
- $R_{\text{Hold}_{ag}\text{Perm}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Obl}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Perm}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Obl}_{ag}{\sim}\gamma}]$,
- $R_{\text{Hold}_{ag}\text{Perm}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Forb}_{ag}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Perm}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Forb}_{ag}\gamma}]$,
- $R_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\neg\text{Obl}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\neg\text{Obl}_{ag}{\sim}\gamma}]$,
- $R_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\text{Perm}_{ag}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\text{Perm}_{ag}\gamma}]$,
- $R_{\text{Hold}_{ag}\text{Bring}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\gamma}$: $[+\Delta' r_{\text{Hold}_{ag}\text{Bring}_{ag}\gamma\Rightarrow\text{Hold}_{ag}\gamma}]$,
- *F1'*: $[[+\Delta'\text{Hold}_{\text{mario}}\mathit{has\_high\_income}(\text{mario})] + \partial'\text{Hold}_{\text{mario}}\mathit{has\_high\_income}(\text{mario})]$,
- *F2'*: $[[+\Delta'\text{Hold}_{\text{mario}}\mathit{strike\_by\_disaster}] + \partial'\text{Hold}_{\text{mario}}\mathit{strike\_by\_disaster}(\text{mario})]$,
- *R0'*: $[[+\Delta'\text{Hold}_{\text{mario}}r_0] + \partial'\text{Hold}_{\text{mario}}r_0]$,
- *R1'*: $[[+\Delta'\text{Hold}_{\text{mario}}r_1] + \partial'\text{Hold}_{\text{mario}}r_1]$,
- *R2'*: $[[+\Delta'\text{Hold}_{\text{mario}}r_2] + \partial'\text{Hold}_{\text{mario}}r_2]$,
- *R3'*: $[[+\Delta'\text{Hold}_{\text{mario}}r_3] + \partial'\text{Hold}_{\text{mario}}r_3]$,
- $R'_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\text{Forb}_{ag}{\sim}\gamma}$: $[+\partial' r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\text{Forb}_{ag}{\sim}\gamma}]$,
- $R'_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Perm}_{ag}{\sim}\gamma}$: $[+\partial' r_{\text{Hold}_{ag}\text{Obl}_{ag}\gamma\rightarrow\text{Hold}_{ag}\neg\text{Perm}_{ag}{\sim}\gamma}]$,

- $R'_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}\sim\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}\sim\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}_{ag}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}_{ag}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}\sim\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}\sim\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}]$,
- A_{Obl}: $[[F1', R0'] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathrm{Obl}_{\mathrm{mario}}\mathit{pay\_tax}(\mathrm{mario})]$,
- A_{Perm}: $[[A_{\mathrm{Obl}}, R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathrm{Perm}_{\mathrm{mario}}\mathit{pay\_tax}(\mathrm{mario})]$,
- A_{Forb}: $[[A_{\mathrm{Obl}}, R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\mathrm{Forb}_{ag}\sim\gamma}] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathrm{Forb}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario})]$,
- $A_{\neg\mathrm{Perm}\neg}$: $[[A_{\mathrm{Obl}}, R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Perm}i\sim\gamma}] + \partial'\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Perm}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario})]$,
- $A_{\neg\mathrm{Obl}\neg}$: $[[A_{\mathrm{Obl}}, R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}i\sim\gamma}] + \partial'\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Obl}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario})]$,
- $B_{\mathrm{Perm}\neg}$: $[[F1', F2', R1'] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathrm{Perm}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario})]$,
- $B_{\neg\mathrm{Obl}}$: $[[B_{\mathrm{Perm}\neg}, R'_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}i\sim\gamma}]+\partial'\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Obl}_{\mathrm{mario}}\mathit{pay\_tax}(\mathrm{mario})]$,
- $B_{\neg\mathrm{Forb}\neg}$: $[[B_{\mathrm{Perm}\neg}, R'_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}i\gamma}]+\partial'\mathrm{Hold}_{\mathrm{mario}}\neg\mathrm{Forb}_{\mathrm{mario}}\neg\mathit{pay\_tax}(\mathrm{mario})]$,
- C: $[[B_{\mathrm{Perm}\neg}, R2'] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathrm{Des}_{\mathrm{mario}}\mathit{invest}(\mathrm{mario})]$,
- D: $[[C, R3'] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathrm{Bring}_{\mathrm{mario}}\mathit{invest}(\mathrm{mario})]$,
- *D'*: $[[D, R'_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}] + \partial'\mathrm{Hold}_{\mathrm{mario}}\mathit{invest}(\mathrm{mario})]$.

Definition 9.11. *A (proper) sub-argument of an argument A is a (proper) sub-tree of the tree associated to A.*

Definition 9.12. *A tagged assertion ($+\Delta'\gamma$ or $+\partial'\gamma$) is a conclusion of an argument if it labels a node of the argument.*

A more usual alternative would be to regard only the root of an argument as its unique conclusion, but this choice would make the other definitions more complex. Since conclusions can be differently qualified depending on the rules used, arguments are differentiated as follows:

Definition 9.13. *A supportive argument is a finite argument in which no defeater is used.*

Definition 9.14. *A strict argument is an argument in which any node is tagged by $+\Delta'$.*

Definition 9.15. *An argument that is not strict is called defeasible.*

Example 18 *The argument A_{Obl} is a supportive argument for $+\partial\mathrm{Hold}_{mario}\mathrm{Obl}_{mario}\mathit{pay\_tax}$. It is not a strict argument and thus it is a defeasible argument.*

9.3.2 Acceptable arguments

The precedent Section defined the argument layer and isolated the concept of argument. This Section presents the dialectical layer which is concerned with relations standing amongst arguments. It defines the notion of support and attack, and focuses on the interaction amongst arguments. Firstly, we introduce the notion of support:

Definition 9.16. *A set of arguments S supports a defeasible argument A if every proper sub-argument of A is in S.*

Note that, in our setting, the atomic arguments, constituted of a fact or a rule of the theory, are supported by the empty set.

The conditions that determine which argument can attack or defeat another argument are defined in the following. In the Section presenting the proof theory, a defeasible conclusion is shown to have a proof condition consisting of three phases. In the first phase, a supporting rule r for the desired conclusion is provided. In the second phase, all possible attacks provided by a rule s against the conclusion are considered. In the third phase, counter-attacks are proposed, that is, the counter-attack consists of a rule w such that w is stronger than s.
So, in the proof condition, the relation of attack between the first and second phase is somewhat different of the relation of attack between the second and third phase. To reflect this, we provide the notion of attack and counter-attack between arguments in the following.

Definition 9.17. *An argument S attacks a defeasible argument R if and only if*

- $+\#'\gamma$ *and* $+\partial'\beta$ *are conclusions of the arguments S and R respectively, where* $\# \in \{\Delta', \partial'\}$*, and*
- $\gamma \in \mathscr{C}_{\text{onflict}}(\beta)$.

Example 19 *The argument* $B_{\text{Perm}\neg}$ *attacks the argument* A_{Obl} *and vice versa.*

Definition 9.18. *An argument W defeats a defeasible argument S if and only if*

(1) $+\#\gamma$ *and* $+\partial'\beta$ *are conclusions of a rule* $w \in R[\gamma]$ *and a rule* $s \in R[\beta]$ *respectively, where* $\# \in \{\Delta', \partial'\}$*, and*
(2) if $\# = \partial$*, (2.1)* $\gamma \in \mathscr{D}_{\text{efeat}}(\beta)$, $\langle \gamma, (s \succ w) \rangle \notin \mathscr{S}_{\text{up}}$*, or*
(2.2) $\langle \gamma, (w \succ s) \rangle \in \mathscr{S}_{\text{up}}$.

Example 20 *The argument* A_{Obl} *attacks and defeats the argument* $B_{\text{Perm}\neg}$ *whereas the argument* $B_{\text{Perm}\neg}$ *attacks, but does not defeat the argument* A_{Obl}.

Defeasible reasoning differentiates traditionally between rebuttal and undercutting. We stick to the tradition and define the notion of undercutting as follows:

Definition 9.19. *An argument A undercuts a defeasible argument B if A attacks a proper sub-argument of B.*

In this setting, an argument that is attacked but not undercut is said to be rebutted.

Definition 9.20. *A set of arguments S undercuts a defeasible argument B if there is an argument A supported by S that attacks a proper sub-argument of B.*

Definition 9.21. *A set of arguments S defeats a defeasible argument B if there is an argument A supported by S that defeats B.*

Comparing arguments by pairs is not enough since an attacking argument can in turn be attacked by other arguments. In the following, we will define justified arguments, i.e. arguments that have no viable attacking argument in the discourse, and rejected arguments that are attacked by justified argument. As in many argumentation systems, we base the status justified or rejected of arguments on the concept of acceptability of an argument w.r.t. to set of arguments S. That an argument A is acceptable w.r.t. to set of arguments S means that any attacker against A is defeated by an argument supported by S. In this line, we next present a slightly adapted version of P.M.Dung 's definition of acceptability [68].

Definition 9.22. *An argument A is acceptable w.r.t. a set of arguments S if and only if either*

(1) A is strict, or
(2) for any argument B attacking A
(2.1) B is undercut by S, or
(2.2) B is defeated by S.

That any argument B, which attacks an acceptable argument A and which is not defeated by S, must be undercut (i.e. a proper sub-argument of B must be attacked, not the conclusion) by a counter-argument C supported by S (i.e. the counter-argument C is possibly not a member of S) aims to provide an ambiguity blocking semantics of the system.

9.3.3 Justified arguments

Based on the concept of acceptability we proceed to define justified arguments and justified literals. That an argument A is justified means that it resists every refutation.

Definition 9.23. *The set of justified arguments in a theory D is* $JArgs_D = \bigcup_{i=0}^{\infty} J_{D,i}$ *with*

- $J_{D,0} = \varnothing$,
- $J_{D,i+1} = \{A \in Args_D | A \quad is \quad acceptable \quad w.r.t. \quad J_{D,i}\}$.

where $Args_D$ *is the set of arguments which can be generated from the heory D.*

So, an argument A is acceptable w.r.t. $J_{D,i+1}$ if either A is strict, or any argument B attacking A is undercut by $J_{D,i}$ (i.e. there is an argument C supported by $J_{D,i}$ that attacks a proper sub-argument of B) or defeated by an argument supported by $J_{D,i}$.

Definition 9.24. *A tagged assertion* $+\partial'\gamma$ *is justified if and only if it is the conclusion of a supportive argument in* $JArgs_D$.

Example 21 *Let us illustrate these definitions by a step-by-step construction of the set of justified arguments of the theory* D_{14}.
We need to find out any argument that is acceptable w.r.t. $J_{D_{14},0}$ *which is by definition the empty set. The following arguments are strict and thus are acceptable w.r.t* $J_{D,0}$.

- *F1*: $[+\Delta'\mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}(\mathrm{mario})]$,
- *F2*: $[+\Delta'\mathrm{Hold}_{\mathrm{mario}}\mathit{strike\_by\_disaster}(\mathrm{mario})]$,
- *R0*: $[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_0]$,
- *R1*: $[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_1]$,
- *R2*: $[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_2]$,
- *R3*: $[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_3]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\mathrm{Forb}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\mathrm{Forb}_{ag}{\sim}\gamma}]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Perm}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Perm}_{ag}{\sim}\gamma}]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}_{ag}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}_{ag}\gamma}]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}]$,
- $R_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}$: $[+\Delta' r_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}]$,

Let us now consider non-strict arguments. The following arguments are not attacked by any argument, and thus are acceptable w.r.t. $J_{D_{14},0}$.

- *F1'*: $[[+\Delta'\mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}]+\partial'\mathrm{Hold}_{\mathrm{mario}}\mathit{has\_high\_income}(\mathrm{mario})]$,
- *F2'*: $[[+\Delta'\mathrm{Hold}_{\mathrm{mario}}\mathit{strike\_by\_disaster}]+\partial'\mathrm{Hold}_{\mathrm{mario}}\mathit{strike\_by\_disaster}(\mathrm{mario})]$,
- *R0'*: $[[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_0]+\partial'\mathrm{Hold}_{\mathrm{mario}}r_0]$,
- *R1'*: $[[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_1]+\partial'\mathrm{Hold}_{\mathrm{mario}}r_1]$,
- *R2'*: $[[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_2]+\partial'\mathrm{Hold}_{\mathrm{mario}}r_2]$,
- *R3'*: $[[+\Delta'\mathrm{Hold}_{\mathrm{mario}}r_3]+\partial'\mathrm{Hold}_{\mathrm{mario}}r_3]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\mathrm{Forb}_{ag}{\sim}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\mathrm{Forb}_{ag}{\sim}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Perm}_{ag}{\sim}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Perm}_{ag}{\sim}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}_{ag}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma\rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Forb}_{ag}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\neg\mathrm{Obl}_{ag}{\sim}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Obl}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\mathrm{Perm}_{ag}\gamma}]$,
- $R'_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}$: $[+\partial' r_{\mathrm{Hold}_{ag}\mathrm{Bring}_{ag}\gamma\Rightarrow\mathrm{Hold}_{ag}\gamma}]$,

Any argument A_x is attacked by an argument B_x which is neither undercut nor counter-attacked by $J_{D_{14},0}$. Hence the arguments A_x is not acceptable w.r.t. $J_{D_{14},0}$. Similarly, the arguments B_x, C, D and D' are attacked by the argument A_x which are neither undercut nor counter-attacked by $J_{D_{14},0}$. Hence the arguments B_x, C, D and D' are not acceptable w.r.t. $J_{D_{14},0}$.

We have parse any tentative to make acceptable arguments w.r.t. $J_{D_{14},0}$.The next step is to find out any argument that is acceptable w.r.t. $J_{D_{14},1}$. Any argument A_x is attacked by an argument B_x which is neither undercut nor counter-attacked by $J_{D_{14},1}$. Hence the argument A_x are not acceptable w.r.t. $J_{D_{14},1}$. The arguments B_x, C, D and D' are attacked by an argument A_x which is defeated by $J_{D_{14},1}$ (consider the argument $B_{\mathrm{Perm}\neg}$). Hence the arguments B_x, C, D and D' are acceptable w.r.t. $J_{D_{14},1}$. We have parse any tentative to make acceptable arguments w.r.t. $J_{D_{14},1}$, therefore

$$J_{D_{14},2} = \{B_{\mathrm{Perm}\neg}, B_{\neg\mathrm{Obl}}, B_{\neg\mathrm{Forb}\neg}, C, D, D'\}$$

It is easy to see that $\forall i > 2$, $J_{D_{14},i} = \varnothing$. *By definition, the set of justified arguments* $JArgs_{D_{14}}$ *is* $\bigcup_{i=0}^{+\infty} J_{D_{14},i}$.

That a tagged assertion $+\partial'\gamma$ is justified means that it is provable (+∂). However, Defeasible Logic permits to express when a conclusion is not provable (-∂). Briefly, that a conclusion is not provable means that every possible argument for that conclusion has been refuted. In the following, this notion is captured by assigning the status rejected to arguments that are refuted. Roughly speaking, an argument is rejected if it has a rejected sub-argument or it cannot overcome an attack from a justified argument. Given an argument A, a set S of arguments (to be thought of as arguments that have already been rejected), and a set J of arguments (to be thought of as justified arguments that may be used to support attacks on A), we assume the following definition of the argument A being rejected by S and J:

Definition 9.25. *An argument A is rejected by the sets of arguments S and J when A is not strict and if (i) a proper sub-argument of A is in S or (ii) it is attacked by an argument supported by J.*

Definition 9.26. *The set of rejected arguments in a theory D w.r.t. J is* $RArgs_D(J) = \bigcup_{i=0}^{\infty} R_{D,i}$ *with*

- $R_{D,0}(J) = \varnothing$,
- $R_{D,i+1}(J) = \{a \in Args_D | a \quad is \quad rejected \quad by \quad R_{D,i}(J) \quad and \quad J\}$.

Definition 9.27. *A tagged assertion* $+\partial'\gamma$ *is rejected by J if there is no argument in* $Args_D - RArgs_D(J)$ *that ends with as a supportive rule for* $+\partial'\gamma$.

As shortcut, we say that an argument is rejected if it is rejected w.r.t. $JArgs_D$ and a literal is rejected if it is rejected by $JArgs_D$.

An argumentation semantics with ambiguity blocking can now be provided by characterising conclusions of Modal Defeasible Logic in argumentation terms:

Definition 9.28. *Let D be a defeasible theory and* γ *be an assertion.*

- $D \vdash +\Delta\gamma$ *iff there is a strict argument supporting* $+\Delta'\gamma$ *in* $Args_D$.
- $D \vdash -\Delta\gamma$ *iff there is no strict argument supporting* $+\Delta'\gamma$ *in* $Args_D$.
- $D \vdash +\partial\gamma$ *iff* $+\partial'\gamma$ *is justified.*
- $D \vdash -\partial\gamma$ *iff* $+\partial'\gamma$ *is rejected by* $Jargs_D$.

This argumentation semantics is consistent with the proof theories of the presented modal defeasible logic in the sense that conclusions get similarly tagged. The proof not provided here is similar to the one in [91]. It follows that for any defeasible theory, no argument is both justified and rejected, and thus no literal is both justified and rejected. Eventually, if the set $JArgs_D$ of justified arguments contains two arguments with conflicting conclusions then both arguments are strict. That is, inconsistent conclusions can be reached only when the strict part of the theory is inconsistent.

10

Temporal modal defeasible logic

Temporal modal defeasible logic extends modal defeasible logic with time. In [95, 92, 94, 96, 89] defeasible logic was extended to capture some temporal aspects in legal reasoning. [95] proved useful in modeling temporal aspects of normative reasoning such as temporal normative positions. [92, 94] allowed for a logical account of the notion of temporal viewpoint (the temporal position from which things are considered) and norm modifications. [89] provided a formal characterization of legal terms (e.g. deadlines). [96] proposes an extension to cope with durative events and with delays between antecedents (causes) and conclusions (effects) of rules in the logic.

All these variants can be criticized under two aspects that this variant aims to repair. Firstly, these extensions adopted a ‘synthetical approach’ in which all temporal dimensions and substantial elements of the norm are represented within the same sentence: a first objective of the present variant is to provide an ‘analytical approach’ in which one sentence represents the substantive content of the norm, and other sentences specify its temporal features (see Chapter 6). Secondly, no argumentation semantics was provided: a second objective here is to give such argumentation semantics.

In the remainder of this Section, we extend the modal defeasible logic of Chapter 9 with an analytical approach and temporal intervals. An argumentation semantics is proposed in Section 10.3. A Prolog meta-program implementing a fragment of the logic is presented in Section 10.4.

10.1 Language

We consider a linear discrete bounded set $\mathscr{T}$ of points of time termed “instants” and over it the order relation $> \subseteq \mathscr{T} \times \mathscr{T}$. We usually denotes the variables ranging over the elements of $\mathscr{T}$ by t and its eventual subscripts, the minimal unit by u. The lower and higher boundaries of $\mathscr{T}$ are denoted respectively by *min* and *max*.

Temporal intervals are defined as sets of instants between two indicated instants. Formally, an interval is a member of the set $\mathrm{Inter} = \{[t_1, t_2] \in \mathscr{T} \times \mathscr{T} | t_1 \leq t_2\}$. As can be noted, this definition allows 'punctual intervals', i.e., intervals of the form $[t,t]$. We shall usually denote intervals by T, plus eventual subscripts.

Temporal intervals are associated to the operators Hold_{ag}, Des_{ag}, Bring_{ag}, Obl_{ag}, Perm_{ag}, Forb_{ag}, Fac_{ag}. For example, the formula $\mathrm{Obl}_{\mathrm{mario}}^{[jan07,dec07]} cooperate(\mathrm{mario})$ means that Mario is obliged between January 2007 and December 2007 to cooperate.

As for basic defeasible logic and modal defeasible logic, a rule is a relationship between temporal modal literals, and we distinguish strict rules, defeasible rules and defeaters. A *strict rule* is an expression of the form $(\phi_1,\ldots,\phi_n \rightarrow \psi)$ such that whenever the premises are indisputable so is the conclusion . A *defeasible rule* is an expression of the form $(\phi_1,\ldots,\phi_n \Rightarrow \psi)$ whose conclusion can be defeated by contrary evidence. An expression $(\phi_1,\ldots,\phi_n \leadsto \psi)$ is a *defeater* used to defeat some defeasible rules by producing evidence to the contrary. Rules can be temporalised by prefixing them with a sequence of operators $(X_i)_{1..n}$, to indicate the modalities and the periods in which they are considered.

As for the modal defeasible logic proposed in Chapter 9, the types of agent are linked to the relations conflict and defeat between modalities (see Section 6.3).

In the following we define the language, that is, the set of rules specifying valid formulas composing a theory in this variant of temporal modal defeasible logic.

Definition 10.1 (Language). *Let $\mathscr{T}$ a linear discrete bounded ordered set of instants of time, in which the minimal unit is u and the lower and higher boundaries of $\mathscr{T}$ are denoted by min and max respectively. Let* Prop *be a set of propositional atoms,* Ag *a finite set of agents,* Lab *be a set of labels, $m,k \in \mathbb{N}$. The sets below are defined as the smallest sets closed under the following rules:*

Modal operators

$$\mathrm{Mod} = \{\mathrm{Hold}_{ag}, \mathrm{Des}_{ag}, \mathrm{Bring}_{ag}, \mathrm{Obl}_{ag}, \mathrm{Perm}_{ag}, \mathrm{Forb}_{ag}, \mathrm{Fac}_{ag} \,|\, ag \in Ag\}$$

Intervals

$$\mathrm{Inter} = \{[t_1,t_2] \,|\, t_1,t_2 \in \mathscr{T}, t_1 \leq t_2\}$$

Temporal modal operators

$$\mathrm{TMod} = \{X^T \,|\, X \in \mathrm{Mod},\, T \in \mathrm{Inter}\}$$

Temporal modal literals

$$\mathrm{TMLit} = \{(X_i)_{1..n}\neg^m\gamma \,|\, \gamma \in \mathrm{Prop}, X_i = \neg^k X,\, X \in \mathrm{TMod}\}$$

Rules

$$\begin{aligned}
&\mathrm{Rule}_s = \{r\colon \quad \phi_1,\ldots,\phi_n \rightarrow \psi \,|\, r \in \mathrm{Lab},\, \phi_1,\ldots,\phi_n, \psi \in \mathrm{TMLit} \cup \mathrm{TMRul}\}\\
&\mathrm{Rule}_d = \{r\colon \quad \phi_1,\ldots,\phi_n \Rightarrow \psi \,|\, r \in \mathrm{Lab},\, \phi_1,\ldots,\phi_n, \psi \in \mathrm{TMLit} \cup \mathrm{TMRul}\}\\
&\mathrm{Rule}_{dft} = \{r\colon \quad \phi_1,\ldots,\phi_n \leadsto \psi \,|\, r \in \mathrm{Lab},\, \phi_1,\ldots,\phi_n, \psi \in \mathrm{TMLit} \cup \mathrm{TMRul}\}\\
&\mathrm{Rule} = \mathrm{Rule}_s \cup \mathrm{Rule}_d \cup \mathrm{Rule}_{dft}
\end{aligned}$$

Temporal modal rules

$$\begin{aligned}
&\text{TMRul}_s = \{(X_i)_{1..n}\neg^m r \,|\, X_i \in \{\neg^k \text{Hold}^T_{ag} | ag \in \text{Ag}, T \in \text{Inter}\}\; r \in \text{Rule}_s\}\\
&\text{TMRul}_d = \{(X_i)_{1..n}\neg^m r \,|\, X_i \in \{\neg^k \text{Hold}^T_{ag} | ag \in \text{Ag}, T \in \text{Inter}\},\, r \in \text{Rule}_d\}\\
&\text{TMRul}_{dft} = \{(X_i)_{1..n}\neg^m r \,|\, X_i \in \{\neg^k \text{Hold}^T_{ag} | ag \in \text{Ag}, T \in \text{Inter}\},\, r \in \text{Rule}_{dft}\}\\
&\text{TMRul} = \text{TMRul}_s \cup \text{TMRul}_d \cup \text{TMRul}_{dft}
\end{aligned}$$

Conflict relations

$$\text{Conflict} = \{\text{conflict}(\gamma,\beta) \,|\, \gamma,\beta \in \text{TMLit} \cup \text{TMRul},\, \gamma =^T \beta\}$$

Defeat relations

$$\text{Defeat} = \{\text{defeat}(\gamma,\beta) \,|\, \gamma,\beta \in \text{TMLit} \cup \text{TMRul},\, \gamma =^T \beta\}$$

Superiority relations

$$\text{Sup} = \{(X_i)_{1..n}(s \succ r) \,|\, X_i \in \{\text{Hold}^T_{ag} | ag \in \text{Ag},\, T \in \text{Inter}\}, s, r \in \text{Lab}\}$$

Definition 10.2 (Defeasible Theory). A defeasible theory is a structure $D = (\mathscr{T}, F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$ where

- $\mathscr{T}$ is a discrete totally ordered set of instants of time,
- $F = F_1 \cup F_2$ is a finite set of facts, where
  - $F_1 \subseteq \text{TMLit}$ and,
  - $F_2 = \{\text{Hold}^T_{ag} force(r), \text{Hold}^T_{ag} efficacious(r) | T \in \text{Inter}, ag \in Ag, r \in R_2\}$
- $R = R_1 \cup R_2$ is a finite set of rules such that each rule has an unique label, and where:
  - $R_1 \subseteq \text{TMRul}$ and,
  - $R_2 = \{r_{\gamma\to\beta}:\quad \gamma \to \beta | \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}:\quad \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$
- $\mathscr{S} \subseteq \text{Sup}$ is a set of acyclic superiority relations,
- $Ag = Ag_1 \cup \{\text{obj}\}$ is a set of agents such that $\forall a \in Ag_1, a \neq \text{obj}$,
- $\mathscr{C} \subseteq \text{Conflict}$ is a set of conflict relations,
- $\mathscr{D} \subseteq \text{Defeat}$ is a set of defeat relations.

We use some abbreviations. The set of sequences of temporal operators $(X_i)_{1..n}$ which does not contain any negated temporal operator is denoted Seq^+:

$$Seq^+ = \{(X_i)_{1..n} \,|\, X_i \in \text{TMod}\}$$

The set of antecedents $\{\phi_1, \ldots, \phi_n\}$ of a rule is denoted $A(r)$, and its *consequent* is denoted. The set of rules whose consequent is γ is denoted $R[\gamma]$:

$$R[\gamma] = \{r | r \in R, C(r) = \gamma\}.$$

The set of strict rules whose consequent is γ is denoted $R_s[\gamma]$. The set of defeasible and strict rules whose consequent is γ is denoted $R_{sd}[\gamma]$.

10.2 Proof theory

A conclusion of a theory D is a tagged temporal modal literal or rule having one of the following forms:

$+\Delta\gamma$ meaning that γ is definitely provable in D.
$-\Delta\gamma$ meaning that γ is not definitely provable in D.
$+\partial\gamma$ meaning that γ is defeasible provable in D.
$-\partial\gamma$ meaning that γ is not defeasible provable in D.

Provability is based on the concept of a derivation (or proof) in D. A derivation is a finite sequence $P = (P(1),..,P(n))$ of literals tagged by $+\Delta$ or $+\partial$, or rules tagged by $+\Delta$ or $+\partial$ or $+\nabla$. $P(1..n)$ denotes the initial part of the sequence P of length n. Each tagged literal or rule satisfies some proof conditions, which correspond to inference rules for the four kinds of conclusions we have mentioned above.

Before moving to the conditions governing provability of conclusions, we need to introduce some preliminary notions.

Definition 10.3 (Intervals: basic relations). Let 'start()' and 'end()' be the functions that return the lower and upper bounds of an interval respectively. Let u be the temporal unit. For any interval $T_1, T_2, T_3 \in \mathrm{Inter}$, we have:

- $T_1 = T_2$ if and only if $\mathrm{start}(T_2) = \mathrm{start}(T_1)$ and $\mathrm{end}(T_1) = \mathrm{end}(T_2)$,
- $T_1 \sqsubseteq T_2$ if and only if $\mathrm{start}(T_2) \leq \mathrm{start}(T_1)$ and $\mathrm{end}(T_1) \leq \mathrm{end}(T_2)$,
- $\mathrm{over}(T_1, T_2)$ if and only if $\mathrm{start}(T_1) \leq \mathrm{end}(T_2)$ and $\mathrm{start}(T_2) \leq \mathrm{end}(T_1)$,
- $T_1 \sqcup T_2 = T_3$ if and only if $\mathrm{end}(T_1) + u = \mathrm{start}(T_2)$, $\mathrm{start}(T_3) = \mathrm{start}(T_1)$ and $\mathrm{end}(T_3) = \mathrm{end}(T_2)$.

These relations are not meant to be a proposal of an algebra of intervals (e.g. [5]), instead, they have been chosen in order to make the proof conditions as simple as possible. To lighten the presentation, we use some abbreviations consisting in placing temporal modal expression as an argument of the previous relations for intervals.

Definition 10.4. Let $(X_i^{T_{1i}})_{1..n}\gamma, (X_i^{T_{2i}})_{1..n}\gamma, (X_i^{T_{3i}})_{1..n}\gamma \in \mathrm{TMLit} \cup \mathrm{TMRul} \cup \mathrm{Sup}$, we have:

- $(X_i^{T_{1i}})_{1..n}\gamma = (X_i^{T_{2i}})_{1..n}\gamma$ if and only if $\forall i \in \{1..n\}$, $T_{1i} = T_{2i}$,
- $(X_i^{T_{1i}})_{1..n}\gamma =^T (Y_i^{T_{2i}})_{1..n}\beta$ if and only if $\forall i \in \{1..n\}$, $T_{1i} = T_{2i}$,
- $(X_i^{T_{1i}})_{1..n}\gamma \sqsubseteq (X_i^{T_{2i}})_{1..n}\gamma$ if and only if $\forall i \in \{1..n\}$, $T_{1i} \sqsubseteq T_{2i}$,
- $\mathrm{over}((X_i^{T_{1i}})_{1..n}\gamma, (X_i^{T_{2i}})_{1..n}\gamma)$ if and only if $\exists i \in \{1..n\}$, $\mathrm{over}(T_{1i}, T_{2i})$,
- $(X_i^{T_{1i}})_{1..n}\gamma \sqcup (X_i^{T_{2i}})_{1..n}\gamma = (X_i^{T_{3i}})_{1..n}\gamma$ if and only if $\exists i \in \{1..n\}$, $T_{1i} \sqcup T_{2i} = T_{3i}$, $\forall j \neq i$, $start(T_{1j}) = start(T_{2j}) = start(T_{3j})$, $end(T_{1j}) = end(T_{2j}) = end(T_{3j})$.

For example, we can write $\mathrm{Hold}_{\mathrm{mario}}^{[10,15]}\neg\mathrm{Obl}_{\mathrm{mario}}^{[30,80]}c \sqsubseteq \mathrm{Hold}_{\mathrm{mario}}^{[0,50]}\neg\mathrm{Obl}_{\mathrm{mario}}^{[0,80]}c$. We define two sets of conflict: the set $\mathscr{C}_{\mathrm{onflict}}^{1}(\gamma)$ which denotes the set of assertions in conflict with γ irrespective to any agent theory, and the set $\mathscr{C}^{*}_{\text{onflict}}(\gamma)$ which denotes the set of assertions in conflict with γ with respect to an agent theory.

Definition 10.5 (Conflicts^{1}). *The set $\mathscr{C}^{1}_{\text{onflict}}(\gamma)$, which denotes the set of assertions in conflict with γ irrespective to any agent theory, is defined as follows. Let $\sim\gamma \in \mathscr{C}^{1}_{\text{onflict}}(\gamma)$, we have:*

- $\mathscr{C}^{1}_{\text{onflict}}(\gamma) \supseteq \{\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\beta'),\, \beta' \in \mathscr{C}^{1}_{\text{onflict}}(\gamma)\}$,
- $\mathscr{C}^{1}_{\text{onflict}}(\gamma) \supseteq \{\beta \,|\, \gamma \in \mathscr{C}^{1}_{\text{onflict}}(\beta)\}$.

For any $\gamma \in \text{TMLit} \cup \text{TMRul}$,

- $\mathscr{C}^{1}_{\text{onflict}}(\gamma) = \{\neg\gamma\}$,

For any $X \in \{\text{Hold}_{ag}, \text{Des}_{ag}, \text{Bring}_{ag}\}$, $\gamma \in \text{TMLit} \cup \text{TMRul}$,

- $\mathscr{C}^{1}_{\text{onflict}}(X^{T}\gamma) \supseteq \{X^{T} \sim\gamma\}$,

For any $\gamma \in \text{TMLit}$,

- $\mathscr{C}^{1}_{\text{onflict}}(\text{Obl}^{T}_{ag}\gamma) \supseteq \{\neg\text{Perm}^{T}_{ag}\gamma,\, \text{Perm}^{T}_{ag}\sim\gamma\}$,
- $\mathscr{C}^{1}_{\text{onflict}}(\text{Fac}^{T}_{ag}\gamma) \supseteq \{\text{Obl}^{T}_{ag}\sim\gamma,\, \text{Obl}^{T}_{ag}\gamma\}$.

Definition 10.6 (Conflicts*). Let $\mathscr{C}^{*}_{\text{onflict}}$ a set of conflict relations $\text{conflict}^{*}(\gamma,\beta)$ defined as follows. Let $\gamma,\beta \in \text{TMLit} \cup \text{TMRul}$, and let a theory $D = (F,R,\mathscr{S},Ag,\mathscr{C},\mathscr{D})$, we have:

- $\text{conflict}^{*}(\gamma,\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\text{conflict}(\gamma,\beta) \in \mathscr{C}$,
- $\text{conflict}^{*}(\gamma,\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\exists\beta \in \mathscr{E}_{\text{quivalent}}(\beta')$, $\text{conflict}^{*}(\gamma,\beta') \in \mathscr{C}^{*}_{\text{onflict}}$,
- $\text{conflict}^{*}(\gamma,\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\gamma \sqsubseteq \gamma'$, $\beta \in \beta'$, $\text{conflict}^{*}(\gamma',\beta') \in \mathscr{C}^{*}_{\text{onflict}}$,
- $\text{conflict}^{*}(\text{Hold}^{T}_{\text{obj}}\gamma, \text{Hold}^{T}_{\text{obj}}\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\text{conflict}^{*}(\gamma,\beta) \in \mathscr{C}^{*}_{\text{onflict}}$,
- $\text{conflict}^{*}(\gamma,\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\text{conflict}^{*}(\text{Hold}^{T}_{\text{obj}}\gamma, \text{Hold}^{T}_{\text{obj}}\beta) \in \mathscr{C}^{*}_{\text{onflict}}$,
- $\text{conflict}^{*}(\text{Hold}^{[t_1,t_3]}_{ag}X^{[t_1,t_2]}_{ag}\gamma,\, \text{Hold}^{[t_1,t_3]}_{ag}Y^{[t_1,t_2]}_{ag}\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\text{conflict}^{*}(X^{[t_1,t_2]}_{ag}\gamma,\, Y^{[t_1,t_2]}_{ag}\beta) \in \mathscr{C}^{*}_{\text{onflict}}$,
- $\text{conflict}^{*}(X^{T}_{ag}\gamma,\, Y^{T}_{ag}\beta) \in \mathscr{C}^{*}_{\text{onflict}}$ if $\text{conflict}^{*}(\text{Hold}^{T}_{ag}X^{T}_{ag}\gamma,\, \text{Hold}^{T}_{ag}Y^{T}_{ag}\beta) \in \mathscr{C}^{*}_{\text{onflict}}$.

As abbreviation, we have $\mathscr{C}^{*}_{\text{onflict}}(\gamma) = \{\beta \,|\, \text{conflict}^{*}(\gamma,\beta)\, or\, \text{conflict}^{*}(\beta,\gamma) \in \mathscr{C}^{*}_{\text{onflict}}\}$.

We define the set $\mathscr{C}_{\text{onflict}}(\gamma) = \{\beta \,|\, \beta \in \mathscr{C}^{1}_{\text{onflict}}(\gamma) \cup \mathscr{C}^{*}_{\text{onflict}}(\gamma), \text{over}(\beta,\gamma)\}$ as the set of assertions in conflict with γ. Next, we define the equivalent assertions.

Definition 10.7 (Equivalences). The set $\mathscr{E}_{\text{quivalent}}(\gamma)$, which denotes the set of assertions equivalent to γ, is defined as follows. For any $\gamma \in \text{TMLit} \cup \text{TMRul}$, $X \in \text{TMod}$, $n \in \mathbb{N}$ and $\sim\gamma \in \mathscr{C}_{\text{onflict}}(\gamma)$, we have:

- $\mathscr{E}_{\text{quivalent}}(\gamma) = \{\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{E}_{\text{quivalent}}(\gamma) = \{\neg^{2n}\gamma\}$,
- $\mathscr{E}_{\text{quivalent}}(\neg^{n}\gamma) = \{\neg^{n}\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{E}_{\text{quivalent}}(\neg^{n}X\gamma) = \{\neg^{n}X\beta \,|\, \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{E}_{\text{quivalent}}(\text{Obl}^{T}_{ag}\gamma) = \{\neg\text{Perm}^{T}_{ag}\sim\gamma,\, \text{Forb}^{T}_{ag}\sim\gamma\}$.

Next, we provide by means of the conversion relations which modal expressions can be converted into which modal expressions. The set of assertions which can result by conversion from an assertion γ is denoted $\mathscr{C}_{\text{onvert}}(\gamma)$.

Definition 10.8 (Conversions). The set $\mathscr{C}_{\text{onvert}}(\gamma)$, which denotes the set of assertions which can result by conversion from an assertion γ, is defined as follows. For any $X \in \text{TMod}$, $\gamma \in \text{TMLit} \cup \text{TMRul}$ and $n \in \mathbb{N}$, we have:

- $\mathscr{C}_{\text{onvert}}(\gamma) \supseteq \{\beta \,|\, \beta \in \mathscr{C}_{\text{onvert}}(\beta'), \beta' \in \mathscr{E}_{\text{quivalent}}(\gamma)\}$,
- $\mathscr{C}_{\text{onvert}}(\gamma) \supseteq \{\text{Hold}^{T}_{\text{obj}}\gamma | T \in \text{Inter}\}$,
- $\mathscr{C}_{\text{onvert}}((\neg)^n\text{Hold}^{[t_1,t_2]}_{ag}\gamma) \supseteq \{\text{Hold}^{[t_1,t_3]}_{ag}(\neg)^n\text{Hold}^{[t_1,t_2]}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Hold}^{T}_{ag}(\neg)^n\text{Hold}^{T}_{ag}\gamma) \supseteq \{(\neg)^n\text{Hold}^{T}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Bring}^{T}_{ag}\gamma) \supseteq \{\text{Hold}^{T}_{\text{obj}}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Bring}^{T}_{ag}\gamma) \supseteq \{\text{Hold}^{T}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Obl}^{T}_{ag}\gamma) \supseteq \{\text{Perm}^{T}_{ag}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\text{Fac}^{T}_{ag}\gamma) \supseteq \{\text{Perm}^{T}_{ag}\gamma, \text{Perm}^{T}_{ag}{\sim}\gamma\}$,
- $\mathscr{C}_{\text{onvert}}(\neg^n\gamma) \supseteq \{\neg^n\beta \,|\, \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$,
- $\mathscr{C}_{\text{onvert}}(\neg^n X\gamma) \supseteq \{\neg^n X\beta \,|\, \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$.

Finally, based on the sets of defeating relations provided by a theory, we assume the following sets of defeating assertions.

Definition 10.9 (Defeats). Let $\mathscr{D}_{\text{efeat}}$ a set of defeat relations $\text{defeat}^*(\gamma,\beta)$ defined as follows. Let $\gamma,\beta \in \text{MLit} \cup \text{MRule}$, and let a theory $D = (\mathscr{T},F,R,\mathscr{S},Ag,\mathscr{C},\mathscr{D})$ we have:

- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}(\gamma,\beta) \in \mathscr{D}$,
- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\gamma',\beta') \in \mathscr{D}_{\text{efeat}}$, and $\gamma \sqsubseteq \gamma'$, $\beta \sqsubseteq \beta'$, $\gamma \sqsubseteq \beta$, $\beta \sqsubseteq \gamma$,
- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\exists\beta \in \mathscr{E}_{\text{quivalent}}(\beta')$, $\text{defeat}^*(\gamma,\beta') \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$ if $\exists\gamma \in \mathscr{E}_{\text{quivalent}}(\gamma')$, $\text{defeat}^*(\gamma',\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}(\gamma,\beta) \in \mathscr{C}^*_{\text{onflict}}$ if $\gamma \sqsubseteq \gamma'$, $\beta \in \beta'$, $\text{defeat}(\gamma',\beta') \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\text{Hold}^{T}_{\text{obj}}\gamma, \text{Hold}^{T}_{\text{obj}}\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\gamma,\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\text{Hold}^{[t_1,t_3]}_{ag}\text{Hold}^{[t_1,t_2]}_{ag}\gamma, \text{Hold}^{[t_1,t_3]}_{ag}\text{Hold}^{[t_1,t_2]}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\text{Hold}^{[t_1,t_2]}_{ag}\gamma, \text{Hold}^{[t_1,t_2]}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$,
- $\text{defeat}^*(\text{Hold}^{T}_{ag}\gamma, \text{Hold}^{T}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$ if $\text{defeat}^*(\text{Hold}^{T}_{ag}\text{Hold}^{T}_{ag}\gamma, \text{Hold}^{T}_{ag}\text{Hold}^{T}_{ag}\beta) \in \mathscr{D}_{\text{efeat}}$.

As abbreviation, we have $\mathscr{D}_{\text{efeat}}(\gamma) = \{\beta \,|\, \text{defeat}^*(\beta,\gamma') \in \mathscr{D}_{\text{efeat}}, \text{over}(\gamma,\gamma')\}$.

Definition 10.10 (Superiority $\mathscr{S}^*$). Let $\mathscr{S}^*$ a set of superiority relations $(X_i)_{1..n}(w \succ s)$ defined as follows. Let a theory $D = (\mathscr{T},F,R,\mathscr{S},Ag,\mathscr{C},\mathscr{D})$ we have:

- $(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $(X_i)_{1..n}(w \succ s) \in \mathscr{S}$,
- $\gamma \in \mathscr{S}^*$ if $\gamma \sqsubseteq \gamma'$, $\gamma' \in \mathscr{S}$,
- $\text{Hold}^{T}_{\text{obj}}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$,
- $\text{Hold}^{[t_1,t_3]}_{ag}\text{Hold}^{[t_1,t_2]}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $\text{Hold}^{[t_1,t_2]}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$,
- $\text{Hold}^{T}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$ if $\text{Hold}^{T}_{ag}\text{Hold}^{T}_{ag}(X_i)_{1..n}(w \succ s) \in \mathscr{S}^*$.

As abbreviation, we have $\mathscr{S}^*((X_i)_{1..n}(w \succ s)) = \{(X_i)_{1..n}(w \succ s) \mid (X_i)_{1..n}(w \succ s) \in \mathscr{S}^*\}$.

Definition 10.11 (Superiority $\mathscr{S}_{\text{up}}$). Let $\mathscr{S}_{\text{up}}$ denote the set of tuples $\langle (X_i)_{1..n}\gamma, (Y_i)_{1..n}(w \succ s)\rangle$ where $(X_i)_{1..n}\gamma$ is an assertion, and $(Y_i)_{1..n}(w \succ s) \in \mathscr{S}^*$. We have $\langle (X_i)_{1..n}\gamma, (Y_i)_{1..n}(w \succ s)\rangle \in \mathscr{S}_{\text{up}}$ if and only if $(f^{\succ}(X_i))_{1..n} = (Y_i)_{1..n}$ where $f^{\succ}(\neg^k X_{ag}) = \text{Hold}_{ag}$.

As abbreviation, we have $\langle (X_i)_{1..n}\gamma, (w \succ s)\rangle \in \mathscr{S}_{\text{up}}$ if and only if $\langle (X_i)_{1..n}\gamma, (Y_i)_{1..n}(w \succ s)\rangle \in \mathscr{S}_{\text{up}}$.

We present separately the condition for a rule to be applicable so that it supports a temporal modal literal or rule. If a temporal modal rule $(X_i)_{1..n}r$ is applicable then (1) the rule r has for consequent γ, (2) the (possible empty) sequence of operators $(X_i)_{1..n}$ does not contain a negated operator (i.e.) $(X_i)_{1..n} \in \mathit{Seq}^+$, (3) the temporal modal rule $(X_i)_{1..n}r$ is provable, (4) and in force, and (5) all the antecedents are proved at a temporal position of the rule's efficacy:

If $(X_i)_{1..n}r$ is $\Delta\gamma$-applicable then
(1) $r \in R_s[\gamma']$, $\gamma \sqsubseteq (X_i)_{1..n}\gamma'$, and
(2) $(X_i)_{1..n} \in \mathit{Seq}^+$, and
(3) $+\Delta(X_i)_{1..n}r \in P(1..i)$, and
(4) $+\Delta(X_i)_{1..n}\mathit{force}(r) \in P(1..i)$, and
(5) $\forall (Y_i)_{1..n}\alpha \in A(r)$,
(5.1) $+\Delta(X_i)_{1..n}(Y_i)_{1..n}\alpha \in P(1..i)$, and
(5.2) $X_n = Z^{T'}_{ag}$, $Y_1 = Z^{T}_{ag'}$, $+\Delta(X_i)_{1..n}\text{Hold}^{T}_{ag}\mathit{efficacious}(r) \in P(1..i)$.

A rule, which is not applicable, is discarded following the conditions below:

If $(X_i)_{1..n}r$ is $\Delta\gamma$-discarded then
(1) $r \in R_s[\gamma']$, $\gamma \not\sqsubseteq (X_i)_{1..n}\gamma'$, or
(2) $(X_i)_{1..n} \notin \mathit{Seq}^+$, or
(3) $-\Delta(X_i)_{1..n}r \in P(1..i)$, or
(4) $-\Delta(X_i)_{1..n}\mathit{force}(r) \in P(1..i)$, or
(5) $\exists (Y_i)_{1..n}\alpha \in A(r)$,
(5.1) $-\Delta(X_i)_{1..n}(Y_i)_{1..n}\alpha \in P(1..i)$, or
(5.2) $X_n = Z^{T'}_{ag}$, $Y_1 = Z^{T}_{ag'}$, $-\Delta(X_i)_{1..n}\text{Hold}^{T}_{ag}\mathit{efficacious}(r) \in P(1..i)$.

The conditions for a rule to be ∂-applicable (resp. ∂-discarded) are similar to those for Δ-applicable (resp. Δ-discarded), but where we replace Δ with ∂ and in which the modifiability is accounted for.

If $(X_i)_{1..n}r$ is $\partial\gamma$-applicable then
(1) $r \in R[\gamma']$, $\gamma \sqsubseteq (X_i)_{1..n}\gamma'$, and
(2) $(X_i)_{1..n} \in \mathit{Seq}^+$, and
(3) $+\partial(X_i)_{1..n}r \in P(1..i)$, and
(4) $+\partial(X_i)_{1..n}\mathit{force}(r) \in P(1..i)$, and
(5) $\forall (Y_i)_{1..n}\alpha \in A(r)$,

(5.1) $+\partial(X_i)_{1..n}(Y_i)_{1..n}\alpha \in P(1..i)$, and
(5.2) $X_n = Z^{T'}_{ag}$, $Y_1 = Z^{T}_{ag'}$, $+\partial(X_i)_{1..n}\mathrm{Hold}^{T}_{ag}efficacious(r) \in P(1..i)$.

If $(X_i)_{1..n}r$ is $\partial\gamma$-discarded then
(1) $r \in R[\gamma']$, $\gamma \not\sqsubseteq (X_i)_{1..n}\gamma'$, or
(2) $(X_i)_{1..n} \notin Seq^{+}$, or
(3) $-\partial(X_i)_{1..n}r \in P(1..i)$, or
(4) $-\partial(X_i)_{1..n}force(r) \in P(1..i)$, or
(5) $\exists(Y_i)_{1..n}\alpha \in A(r)$,
(5.1) $-\partial(X_i)_{1..n}(Y_i)_{1..n}\alpha \in P(1..i)$, or
(5.2) $X_n = Z^{T'}_{ag}$, $Y_1 = Z^{T}_{ag'}$, $-\partial(X_i)_{1..n}\mathrm{Hold}^{T}_{ag}efficacious(r) \in P(1..i)$.

We are now ready to define the proof theory that is, the inference conditions to derive tagged conclusions from a given theory D. Note that the formalism we have introduced allows us to temporalise rules, thus we have to admit the possibility that rules are not only given but can be proved to hold for certain span of time. Accordingly we have to give conditions that allow us to derive rules instead of literals. Proofs to determine whether a temporal modal literal or rule γ is a definite conclusion of a theory D follow the same path. A temporal modal literal or rule γ is definitely provable ($+\Delta$) if (1) there exists a temporal modal literal or rule γ' in the set F of facts or in the set R of rules such that $\gamma \sqsubseteq \gamma'$, or (2) there is an applicable strict rule to provide γ, or (3) γ is definitely provable in some temporal sub-intervals.

If $P(i+1) = +\Delta\gamma$ then
(1) $\exists\gamma' \in F \cup R$, $\gamma \sqsubseteq \gamma'$, or
(2) $\exists r \in R_s$, r is $\Delta\gamma$-applicable, or
(3) $\exists\gamma_1, \gamma_2, \gamma_1 \sqcup \gamma_2 = \gamma$, $+\Delta\gamma_1 \in P(1..i)$ and $+\Delta\gamma_2 \in P(1..i)$.

To prove that a temporal modal literal is not definitely provable we have to show that any attempt to give a definite proof fails. A temporal modal rule r is not definitely provable if (1) there is not such rule in the set of rules defined in some sup-interval, and (2) there is no applicable strict rule to provide γ, and (2) r is not defined in any sub-intervals.

If $P(i+1) = -\Delta\gamma$ then
(1) $\forall\gamma' \in F \cup R$, $\gamma \not\sqsubseteq \gamma'$, and
(2) $\forall r \in R_s$, r is $\Delta\gamma$-discarded, and
(3) $\forall\gamma_1, \gamma_2, \gamma_1 \sqcup \gamma_2 = \gamma$, $-\Delta\gamma_1 \in P(1..i)$ or $-\Delta\gamma_2 \in P(1..i)$.

We now turn our attention to defeasible derivations, that is, derivations giving an assertion as a defeasible conclusion of a theory D. We begin with the proof conditions to determine whether a rule is a defeasible conclusion. The proof conditions for a rule introduce a buffer-like proof tag ∇ that allows to consider intermediary steps. Defeasible provability ($+\partial$) for temporal modal literals consists of three phases. In the first phase, we put forward a supported reason for the assertion that we want to prove. Then in the second phase, we consider all possible attacks against the desired conclusion. Finally in the last phase, we have to counter-attack the attacks considered in the second phase.

If $P(i+1) = +\partial\gamma$ then

(1) $+\Delta\gamma \in P(1..i)$, or

(2) (2.1) $\forall\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $-\Delta\beta \in P(1..i)$, and

(2.2) $\exists r \in R_{sd}$, r is $\partial\gamma$-applicable,

(2.3) $\forall\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $\forall s \in R$,

(2.3.1) s is $\partial\beta$-discarded, or

(2.3.2) $\exists w \in R_{sd}$, w is $\partial\gamma$-applicable,

(2.3.2.1) $\gamma \in \mathscr{D}_{\text{efeat}}(\beta)$, $\langle\gamma, (s \succ w)\rangle \notin \mathscr{S}_{\text{up}}$, or

(2.3.2.2) $\langle\gamma, (w \succ s)\rangle \in \mathscr{S}_{\text{up}}$, or

(3) $\exists\gamma_1, \gamma_2, \gamma_1 \sqcup \gamma_2 = \gamma, +\partial\gamma_1 \in P(1..i)$ and $+\partial\gamma_2 \in P(1..i)$.

Let us illustrate the proof condition of the defeasible provability of γ. We have two cases: 1) We show that γ is already definitely provable; or 2) we need to argue using the defeasible part of D. In this second case, to prove γ defeasibly we must show that no complement temporal modal literals β definitely provable (2). We require then there must be a strict or defeasible (applicable) rule r providing γ (2.1). But now we need to consider possible attacks, i.e., that is, any rule s supporting attacking complement temporal modal literals. Note that here we consider defeaters, too, whereas they could not be used to support the conclusion γ. These attacking rules s have to be discarded (2.2.1), or must be defeated by a stronger rule w supporting γ (2.2.2). Finally, we have to cater for the case where γ is defeasible provable on sub-intervals making up γ (3).

To prove that a temporal modal literal is not defeasibly provable we have to show that any attempt to give a proof fails.

If $P(i+1) = -\partial\gamma$ then

(1) $-\Delta\gamma \in P(1..i)$, and

(2) (2.1) $\exists\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $+\Delta\beta \in P(1..i)$, or

(2.2) $\forall r \in R_{sd}$, r is $\partial\gamma$-applicable,

(2.3) $\exists\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$, $\exists s \in R$,

(2.3.1) s is $\partial\beta$-applicable, and

(2.3.2) $\forall w \in R_{sd}$, w is $\partial\gamma$-discarded,

(2.3.2.1) $\gamma \notin \mathscr{D}_{\text{efeat}}(\beta)$ or $\langle\gamma, (s \succ w)\rangle \in \mathscr{S}_{\text{up}}$, and

(2.3.2.2) $\langle\gamma, (w \succ s)\rangle \notin \mathscr{S}_{\text{up}}$, and

(3) $\forall\gamma_1, \gamma_2, \gamma_1 \sqcup \gamma_2 = \gamma, -\partial\gamma_1 \in P(1..i)$ or $-\partial\gamma_2 \in P(1..i)$.

Example 22 *Suppose the following theory $D_{22} = (\mathscr{T}, F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$ where $\mathscr{T}$ are dates of the Gregorian calendar, and*

$$
\begin{array}{ll}
F = \{\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{opened}(\mathrm{account}), & \\
\quad \mathrm{Hold}^{[1Jan08,31Dec08]}_{\mathrm{mario}}\mathit{positive}(\mathrm{account}), & \\
\quad \mathrm{Hold}^{[1Jan09,20Jan09]}_{\mathrm{mario}}\neg\mathit{positive}(\mathrm{account}), & \\
\quad \mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{force}(r_0), & \mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{efficacious}(r_0), \\
\quad \mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{force}(r_1), & \mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{efficacious}(r_1), \\
\quad \mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{force}(r_2), & \mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}\mathit{efficacious}(r_2), \\
\quad \mathrm{Hold}^{[15Jan09,max]}_{\mathrm{mario}}\mathit{force}(r_3), & \mathrm{Hold}^{[15Jan09,max]}_{\mathrm{mario}}\mathrm{Hold}^{[1Jan09,max]}_{\mathrm{mario}}\mathit{efficacious}(r_3)\}
\end{array}
$$

$R = R_1 \cup R_2$, *where*

$$
\begin{aligned}
R_1 = \{&\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}(r_0\colon\ \mathrm{Hold}^{t}_{x}\mathit{opened}(\text{account})\\
&\qquad\qquad\Rightarrow \mathrm{Obl}^{t}_{x}\mathit{positive}(\text{account})),\\
&\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}(r_1\colon\ \mathrm{Hold}^{t-1}_{x}\mathit{positive}(\text{account}), \mathrm{Hold}^{t}_{x}\neg\mathit{positive}(\text{account})\\
&\qquad\qquad\Rightarrow \mathrm{Perm}^{[t,t+15\,days]}_{x}\neg\mathit{positive}(\text{account})),\\
&\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}(r_2\colon\ \mathrm{Hold}^{t}_{x}\neg\mathit{positive}(\text{account}), \mathrm{Obl}^{t}_{x}\mathit{positive}(\text{account})\\
&\qquad\qquad\Rightarrow \mathrm{Hold}^{t}_{x}\mathit{blocked}(\text{account})),\\
&\mathrm{Hold}^{[1Jan09,max]}_{\text{mario}}(r_3\colon\ \mathrm{Hold}^{t-1}_{x}\mathit{positive}(\text{account}), \mathrm{Hold}^{t}_{x}\neg\mathit{positive}(\text{account})\\
&\qquad\qquad\Rightarrow \mathrm{Perm}^{[t,t+30\,days]}_{x}\neg\mathit{positive}(\text{account}))\},
\end{aligned}
$$

$R_2 = \{r_{\gamma\rightarrow\beta}\colon\ \ \gamma \rightarrow \beta | \beta \in \mathscr{E}_{\text{quivalent}}(\gamma)\} \cup \{r_{\gamma\Rightarrow\beta}\colon\ \ \gamma \Rightarrow \beta | \beta \in \mathscr{C}_{\text{onvert}}(\gamma)\}$,

$\mathscr{S} = \{\mathrm{Hold}^{[min,max]}_{\text{mario}}(r_1 \succ r_0), \mathrm{Hold}^{[min,max]}_{\text{mario}}(r_3 \succ r_0)\}$,

$Ag = \{\text{mario}, \text{obj}\}$,

$\mathscr{C} = \{\text{conflict}(\mathrm{Obl}^{T}_{\text{mario}}\gamma, \mathrm{Des}^{T}_{\text{mario}}{\sim}\gamma),\ \text{conflict}(\mathrm{Obl}^{T}_{\text{mario}}\gamma, \mathrm{Bring}^{T}_{\text{mario}}{\sim}\gamma)\}$,

$\mathscr{D} = \{\text{defeat}(\mathrm{Obl}^{T}_{\text{mario}}\gamma, \mathrm{Des}^{T}_{\text{mario}}{\sim}\gamma),\ \text{defeat}(\mathrm{Obl}^{T}_{\text{mario}}\gamma, \mathrm{Bring}^{T}_{\text{mario}}{\sim}\gamma)\}$.

The set R_2 includes the following rule:

$$R_{\mathrm{Hold}^{[t_1,t_2]}_{ag}\gamma\Rightarrow\mathrm{Hold}^{[t_1,t_3]}_{ag}\mathrm{Hold}^{[t_1,t_2]}_{ag}\gamma}\colon\quad \mathrm{Hold}^{[t_1,t_2]}_{ag}\gamma \Rightarrow \mathrm{Hold}^{[t_1,t_3]}_{ag}\mathrm{Hold}^{[t_1,t_2]}_{ag}\gamma$$

which is applicable and not attacked by any rule, so we obtain:

$$
\begin{gathered}
+\partial\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathit{opened}(\text{account}),\\
+\partial\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathrm{Hold}^{[1Jan08,31Dec08]}_{\text{mario}}\mathit{positive}(\text{account}),\\
+\partial\mathrm{Hold}^{[1Jan09,max]}_{\text{mario}}\mathrm{Hold}^{[1Jan06,20Jan]}_{\text{mario}}\neg\mathit{positive}(\text{account}),\\
+\partial\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathit{efficacious}(r_0),\\
+\partial\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathit{efficacious}(r_1),\\
+\partial\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathit{efficacious}(r_2),\\
+\partial\mathrm{Hold}^{[15Jan08,max]}_{\text{mario}}\mathrm{Hold}^{[15Jan08,max]}_{\text{mario}}\mathit{efficacious}(r_3).
\end{gathered}
$$

The rules r_0, r_1 and r_3 are applicable, but the application of the rule r_0 to obtain the putative conclusion $\mathrm{Hold}^{[1Jan08,max]}_{\text{mario}}\mathrm{Obl}^{[1Jan08,max]}_{\text{mario}}\mathit{positive}(\text{account})$ *is attacked by the application of the rule r_1 for* $\mathrm{Hold}^{[1Jan09,max]}_{\text{mario}}\mathrm{Perm}^{[1Jan09,15Jan09]}_{\text{mario}}\neg\mathit{positive}(\text{account})$ *and by the application of the rule r_3 for* $\mathrm{Hold}^{[15Jan09,max]}_{\text{mario}}\mathrm{Perm}^{[1Jan09,30Jan09]}_{\text{mario}}\neg\mathit{positive}(\text{account})$. *The rule r_1 and r_3 are stronger than the rule r_0, hence we derive:*

$$+\partial\mathrm{Hold}^{[1Jan08,31Dec08]}_{\text{mario}}\mathrm{Obl}^{[1Jan08,max]}_{\text{mario}}\mathit{positive}(\text{account}),$$

$$+\partial\text{Hold}_{\text{mario}}^{[1Jan09,14Jan09]}\text{Perm}_{\text{mario}}^{[1Jan09,15Jan09]}\neg positive(\text{account}),$$

$$+\partial\text{Hold}_{\text{mario}}^{[1Jan09,14Jan09]}\text{Obl}_{\text{mario}}^{[16Jan09,max]}positive(\text{account}),$$

$$+\partial\text{Hold}_{\text{mario}}^{[15Jan09,max]}\text{Perm}_{\text{mario}}^{[1Jan09,30Jan09]}\neg positive(\text{account}),$$

$$+\partial\text{Hold}_{\text{mario}}^{[16Jan09,max]}\text{Obl}_{\text{mario}}^{[31Jan09,max]}\neg positive(\text{account}).$$

The rule r_2 is applicable, hence we obtain:

$$+\partial\text{Hold}_{\text{mario}}^{[1Jan09,14Jan09]}\text{Hold}_{\text{mario}}^{[16jan07,max]}blocked(\text{account}),$$

$$-\partial\text{Hold}_{\text{mario}}^{[15Jan09,max]}\text{Hold}_{\text{mario}}^{[16jan07,max]}blocked(\text{account}).$$

10.3 Argumentation semantics

As for basic defeasible and modal defeasible logic, temporal modal defeasible logic can be characterized in terms of interacting arguments, giving for it an argumentation semantics. In this Section, we provide a slight adaptation of the argumentation semantics given for modal defeasible logic in order to integrate time.

10.3.1 Arguments

The argument layer defines what arguments are. An argument for a temporal modal assertion (i.e. a literal or a rule) is a proof tree (or monotonic derivation) of that assertion in temporal modal defeasible logic. Nodes are labeled by either temporal modal literals or temporal modal rules tagged by ∂' or Δ'. Nodes are connected by arrows that correspond to grounded inferences rules (see below).

Definition 10.12. *An argument is a proof tree such that:*

- *each node is labeled by a temporal modal literal or rule tagged by ∂' or Δ', and*
- *each leaf mode is labeled by $+\Delta'\gamma$ where $\gamma' \in F \cup R$ and $\gamma \sqsubseteq \gamma'$, and*
- *each compound arrow connecting nodes corresponds to a grounded inference rule of the following types:*

$$\frac{+\Delta'\gamma_1, +\Delta'\gamma_2, \gamma_1 \sqcup \gamma_2 = \gamma}{+\Delta'\gamma} \quad (10.1) \qquad \frac{+\partial'\gamma_1, +\partial'\gamma_2, \gamma_1 \sqcup \gamma_2 = \gamma}{+\partial'\gamma} \quad (10.3)$$

$$\frac{r\, is\, \Delta'\gamma - \text{applicable}}{+\Delta'\gamma} \quad (10.2) \qquad \frac{r\, is\, \partial'\gamma - \text{applicable}}{+\partial'\gamma} \quad (10.4)$$

$$\frac{+\Delta'\gamma}{+\partial'\gamma} \quad (10.5)$$

If $(X_i)_{1..n}r$ *is* $\Delta'(X_i)_{1..n}\gamma$-applicable *then*

(1) $r \in R_s[\gamma']$, $\gamma \sqsubseteq (X_i)_{1..n}\gamma'$, *and*

(2) $(X_i)_{1..n} \in Seq^+$, *and*

(3) $+\Delta'(X_i)_{1..n}r$, *and*

(4) $+\Delta'(X_i)_{1..n}force(r)$, *and*

(5) $\forall (Y_i)_{1..n}\alpha \in A(r)$,

(5.1) $+\Delta'(X_i)_{1..n}(Y_i)_{1..n}\alpha$, *and*

(5.2) $X_n = Z^{T'}_{ag}$, $Y_1 = Z^{T}_{ag'}$, $+\Delta'(X_i)_{1..n}\mathrm{Hold}^{T}_{ag}efficacious(r)$.

If $(X_i)_{1..n}r$ *is* $\partial'(X_i)_{1..n}\gamma$-applicable *then*

(1) $r \in R[\gamma']$, $\gamma \sqsubseteq (X_i)_{1..n}\gamma'$*and*

(2) $(X_i)_{1..n} \in Seq^+$, *and*

(3) $+\partial'(X_i)_{1..n}r$, *and*

(4) $+\partial'(X_i)_{1..n}force(r)$, *and*

(5) $\forall (Y_i)_{1..n}\alpha \in A(r)$,

(5.1) $+\partial'(X_i)_{1..n}(Y_i)_{1..n}\alpha$, *and*

(5.2) $X_n = Z^{T'}_{ag}$, $Y_1 = Z^{T}_{ag'}$, $+\partial'(X_i)_{1..n}\mathrm{Hold}^{T}_{ag}efficacious(r)$.

- *If the rule r in the inference (10.4) is a defeater then the post-condition* $+\partial'\gamma$ *is the root node.*

The last condition specifies that a defeater rule may only be used at the top of an argument: no chaining of defeaters is allowed.

Example 23 *Given the theory* D_{22}*, we can build among others the following arguments:*

- $F1$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}opened(\mathrm{account})]$,
- $F2$: $[+\Delta'\mathrm{Hold}^{[1Jan08,31Dec08]}_{\mathrm{mario}}positive(\mathrm{account})]$,
- $F3$: $[+\Delta'\mathrm{Hold}^{[1Jan09,20Jan09]}_{\mathrm{mario}}\neg positive(\mathrm{account})]$,
- $F4$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}force(r_0)]$,
- $F5$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}efficacious(r_0)]$,
- $F6$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}force(r_1)]$,
- $F7$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}efficacious(r_1)]$,
- $F8$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}force(r_2)]$,
- $F9$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}efficacious(r_2)]$,
- $F10$: $[+\Delta'\mathrm{Hold}^{[15Jan09,max]}_{\mathrm{mario}}force(r_3)]$,
- $F11$: $[+\Delta'\mathrm{Hold}^{[15Jan09,max]}_{\mathrm{mario}}\mathrm{Hold}^{[1Jan09,max]}_{\mathrm{mario}}efficacious(r_3)]$,
- $R0$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}r_0]$,
- $R1$: $[+\Delta'\mathrm{Hold}^{[1jan08,max]}_{\mathrm{mario}}r_1]$,
- $R2$: $[+\Delta'\mathrm{Hold}^{[1Jan08,max]}_{\mathrm{mario}}r_2]$,
- $R3$: $[+\Delta'\mathrm{Hold}^{[1Jan09,max]}_{\mathrm{mario}}r_3]$,
- $R4$: $[+\Delta'\mathrm{Hold}^{[t_1,t_2]}_{ag}\gamma \Rightarrow \mathrm{Hold}^{[t_1,t_3]}_{ag}\mathrm{Hold}^{[t_1,t_2]}_{ag}\gamma]$,

- *R5*: $[+\Delta'\mathrm{Hold}_{ag}^{[t_1,t_2]}\mathrm{Hold}_{ag}^{[t_1,t_2]}\gamma \Rightarrow \mathrm{Hold}_{ag}^{[t_1,t_2]}\gamma]$,
- *F1’*: $[[F1]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}opened(\mathrm{account})]$,
- *F2’*: $[[F2]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,31Dec08]}positive(\mathrm{account})]$,
- *F3’*: $[[F3]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,20Jan09]}\neg positive(\mathrm{account})]$,
- *F4’*: $[[F4]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}force(r_0)]$,
- *F5’*: $[[F5]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}efficacious(r_0)]$,
- *F6’*: $[[F6]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}force(r_1)]$,
- *F7’*: $[[F7]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}efficacious(r_1)]$,
- *F8’*: $[[F8]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}force(r_2)]$,
- *F9’*: $[[F9]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}efficacious(r_2)]$,
- *F10’*: $[[F10]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[15Jan09,max]}force(r_3)]$,
- *F11’*: $[[F11]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[15Jan09,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}efficacious(r_3)]$,
- *R0’*: $[[R0]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}r_0]$,
- *R1’*: $[[R1]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1jan08,max]}r_1]$,
- *R2’*: $[[R2]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}r_2]$,
- *R3’*: $[[R3]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}r_3]$,
- *R4’*: $[[R4]+\partial'\mathrm{Hold}_{ag}^{[t_1,t_2]}\gamma \Rightarrow \mathrm{Hold}_{ag}^{[t_1,t_3]}\mathrm{Hold}_{ag}^{[t_1,t_2]}\gamma]$,
- *R5’*: $[[R5]+\partial'\mathrm{Hold}_{ag}^{[t_1,t_2]}\mathrm{Hold}_{ag}^{[t_1,t_2]}\gamma \Rightarrow \mathrm{Hold}_{ag}^{[t_1,t_2]}\gamma]$,
- *F1”*: $[R4',F1']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}opened(\mathrm{account})]$,
- *F2”*: $[[R4',F2']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,31Dec08]}positive(\mathrm{account})]$,
- *F3”*: $[[R4',F3']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,20Jan09]}\neg positive(\mathrm{account})]$,
- *F5”*: $[[R4,F5']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}efficacious(r_0)]$,
- *F7”*: $[[R4,F7]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}efficacious(r_1)]$,
- *F9”*: $[[R4,F9]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,max]}efficacious(r_2)]$,
- $F2_B$: $[[R4',F1']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}\mathrm{Hold}_{\mathrm{mario}}^{[31Dec08]}positive(\mathrm{account})]$,
- $F3_B$: $[[R4',F3']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09]}\neg positive(\mathrm{account})]$,
- $F4_B$: $[[F4]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}force(r_0)]$,
- $F7_B$: $[[R4,F7]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,max]}efficacious(r_1)]$,
- $R1_B$: $[[R1]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1jan09,max]}r_1]$,
- $A_{\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,31Dec08]}\mathrm{Obl}_{\mathrm{mario}}^{[1Jan08,max]}}$:
  $[[F1'',R0',F4',F5'']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan08,31Dec08]}\mathrm{Obl}_{\mathrm{mario}}^{[1Jan08,max]}positive(\mathrm{account})]$,
- $A_{\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,14Jan09]}\mathrm{Obl}_{\mathrm{mario}}^{[16Jan08,max]}}$:
  $[[F1'',R0',F4',F5'']+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,14Jan09]}\mathrm{Obl}_{\mathrm{mario}}^{[16Jan08,max]}positive(\mathrm{account})]$,
- $B_{\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,14Jan09]}\mathrm{Perm}_{\mathrm{mario}}^{[1Jan09,15Jan09]}}$:
  $[[F2_B,F3_B,R1_B,F4_B,F7_B]+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,14Jan09]}\mathrm{Perm}_{\mathrm{mario}}^{[1Jan09,15Jan09]}\neg positive(\mathrm{account})]$.

Definition 10.13. *A (proper) sub-argument of an argument A is a (proper) sub-tree of the tree associated to A.*

Definition 10.14. *A tagged assertion ($+\Delta'\gamma$ or $+\partial'\gamma$) is a conclusion of an argument if it labels a node of the argument.*

A more usual alternative would be to regard only the root of an argument as its unique conclusion, but this choice would make the other definitions more complex. Since conclusions can be differently qualified depending on the rules used, arguments are differentiated as follows:

Definition 10.15. *A supportive argument is a finite argument in which no defeater is used.*

Definition 10.16. *A strict argument is an argument in which any node is tagged by $+\Delta'$.*

Definition 10.17. *An argument that is not strict is called defeasible.*

Example 24 *The argument $B_{\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,14Jan09]}\mathrm{Perm}_{\mathrm{mario}}^{[1Jan09,15Jan09]}}$ is a supportive argument for $+\partial'\mathrm{Hold}_{\mathrm{mario}}^{[1Jan09,14Jan09]}\mathrm{Perm}_{\mathrm{mario}}^{[1Jan09,15Jan09]}\neg positive(\mathrm{account})]$. It is not a strict argument and thus it is a defeasible argument.*

10.3.2 Acceptable arguments

The precedent Section defined the argument layer and isolated the concept of argument. This Section presents the dialectical layer which is concerned with relations standing amongst arguments. It defines the notion of support and attack, and focuses on the interaction amongst arguments. Firstly, we introduce the notion of support:

Definition 10.18. *A set of arguments S supports a defeasible argument A if every proper sub-argument of A is in S.*

Note that, in our setting, the atomic arguments, constituted of a fact or a rule of the theory, are supported by the empty set.

The conditions that determine which argument can attack another argument are defined in the following. In the Section presenting the proof theory, a defeasible conclusion is shown to have a proof condition consisting of three phases. In the first phase, a supporting rule r is provided. In the second phase, all possible attacks against the desired conclusion are considered, that is, an attack consists of any rule s supporting a complemement temporal modal literal. In the third phase, counter-attacks are proposed by means of a defeating rule w.
So in the proof condition, the relation of attack between the first and second phase is somewhat different of the relation of attack between the second and third phase. To reflect this, we provide the notion of attack and defeat between arguments in the following.

Definition 10.19. *An argument S attacks a defeasible argument R if and only if*

- *$+\#'\beta$ and $+\partial'\gamma$ are conclusions of the arguments S and R respectively, where $\# \in \{\Delta', \partial'\}$, and*
- *$\beta \in \mathscr{C}_{\text{onflict}}(\gamma)$.*

Example 25 *The argument $B_{\text{Hold}^{[1Jan09,14Jan09]}_{\text{mario}}\text{Perm}^{[1Jan09,15Jan09]}_{\text{mario}}}$ attacks the argument $A_{\text{Hold}^{[1Jan08,31Dec08]}_{\text{mario}}\text{Obl}^{[1Jan08,max]}_{\text{mario}}}$ and vice versa.*

Definition 10.20. *An argument W defeats a defeasible argument S if and only if*

(1) $+\#\gamma$ and $+\partial'\beta$ are conclusions of a rule $w \in R[\gamma]$ and a rule $s \in R[\beta]$ respectively, where $\# \in \{\Delta', \partial'\}$, and
(2) if $\# = \partial$, (2.1) $\gamma \in \mathscr{D}_{\text{efeat}}(\beta)$, $\langle \gamma, (s \succ w) \rangle \notin \mathscr{S}_{\text{up}}$, or
(2.2) $\langle \gamma, (w \succ s) \rangle \in \mathscr{S}_{\text{up}}$.

Example 26 *The argument $A_{\text{Hold}^{[1Jan08,31Dec08]}_{\text{mario}}\text{Obl}^{[1Jan08,max]}_{\text{mario}}}$ attacks but does not defeat the argument $B_{\text{Hold}^{[1Jan09,14Jan09]}_{\text{mario}}\text{Perm}^{[1Jan09,15Jan09]}_{\text{mario}}}$ whereas the latter attacks and defeats the former.*

Defeasible reasoning differentiates traditionally between rebuttal and undercutting. We stick to the tradition and define the notion of undercutting as follows:

Definition 10.21. *An argument A undercuts a defeasible argument B if A attacks a proper sub-argument of B.*

In this setting, an argument that is attacked but not undercut is said to be rebutted.

Definition 10.22. *A set of arguments S undercuts a defeasible argument B if there is an argument A supported by S that attacks a proper sub-argument of B.*

Definition 10.23. *A set of arguments S defeats a defeasible argument B if there is an argument A supported by S that defeats B.*

Comparing arguments by pairs is not enough since an attacking argument can in turn be attacked by other arguments. In the following, we will define justified arguments, i.e. arguments that have no viable attacking argument in the discourse, and rejected arguments that are attacked by justified argument. As in many argumentation systems, we base the status justified or rejected of arguments on the concept of acceptability of an argument w.r.t. to set of arguments S. That an argument A is acceptable w.r.t. to set of arguments S means that any attacker against A is defeated by an argument supported by S. In this line, we next present a slightly adapted version of P.M.Dung 's definition of acceptability [68].

Definition 10.24. *An argument A is acceptable w.r.t. a set of arguments S if and only if either*

(1) A is strict, or
(2) for any argument B attacking A
(2.1) B is undercut by S, or
(2.2) B is defeated by S.

The condition (2.1) aims to provide an ambiguity blocking semantics of the system, whereas the condition (2.2) aims to provide a team defeat feature of the system.

10.3.3 Justified arguments

Based on the concept of acceptability we proceed to define justified arguments and justified assertion. That an argument A is justified means that it resists every refutation. Given a defeasible theory D, the set of arguments that can be generated from D is denoted by $Args_D$. The following definition is based on [171]'s definition of fixed point semantics.

Definition 10.25. *The set of justified arguments in a theory D is* $JArgs_D = \bigcup_{i=0}^{\infty} J_{D,i}$ *with*

- $J_{D,0} = \varnothing$,
- $J_{D,i+1} = \{arg \in Args_D \,|\, arg \quad is \quad acceptable \quad w.r.t. \quad J_{D,i}\}$.

So, an argument A is acceptable w.r.t. $J_{D,i+1}$ if either A is strict, or any argument B attacking A is undercut by $J_{D,i}$ (i.e. there is an argument C supported by $J_{D,i}$ that attacks a proper sub-argument of B) or defeated by an argument supported by $J_{D,i}$.

Definition 10.26. *A tagged assertion* $+\partial'\gamma$ *is justified if and only if it is the conclusion of a supportive argument in* $JArgs_D$.

A justified tagged assertion $+\partial'\gamma$ means that it is defeasibly provable (+∂). However, temporal modal defeasible logic permits to express when a conclusion is not provable (-∂). Briefly, that a conclusion is not provable means that every possible argument for that conclusion has been refuted. In the following, this notion is captured by assigning the status rejected to arguments that are refuted. Roughly speaking, an argument is rejected if it has a rejected sub-argument or it cannot overcome an attack from a justified argument. Given an argument A, a set S of arguments (to be thought of as arguments that have already been rejected), and a set J of arguments (to be thought of as justified arguments that may be used to support attacks on A), we assume the following definition of the argument A being rejected by S and J:

Definition 10.27. *An argument A is rejected by the sets of arguments S and J when A is not strict and if (i) a proper sub-argument of A is in S or (ii) it is attacked by an argument supported by J.*

Definition 10.28. *The set of rejected arguments in a theory D w.r.t. J is* $RArgs_D(J) = \bigcup_{i=0}^{\infty} R_{D,i}$ *with*

- $R_{D,0}(J) = \varnothing$,
- $R_{D,i+1}(J) = \{arg \in Args_D \,|\, arg \;\; is \;\; rejected \;\; by \;\; R_{D,i}(J) \;\; and \;\; J\}$.

Definition 10.29. *A tagged assertion $+\partial'\gamma$ is rejected by J if there is no argument in $Args_D - RArgs_D(J)$ that ends with as a supportive rule for $+\partial'\gamma$.*

As shortcut, we say that an argument is rejected if it is rejected w.r.t. $JArgs_D$ and a literal is rejected if it is rejected by $JArgs_D$.

An argumentation semantics with ambiguity blocking can now be provided by characterizing conclusions in argumentation terms:

Definition 10.30. *Let D be a defeasible theory and γ be an assertion.*

- $D \vdash +\Delta\gamma$ *iff there is a strict argument supporting* $+\Delta'\gamma$ *in* $Args_D$.
- $D \vdash -\Delta\gamma$ *iff there is no strict argument supporting* $+\Delta'\gamma$ *in* $Args_D$.
- $D \vdash +\partial\gamma$ *iff* $+\partial'\gamma$ *is justified.*
- $D \vdash -\partial\gamma$ *iff* $+\partial'\gamma$ *is rejected by* $Jargs_D$.

This argumentation semantics is consistent with the proof theories of the presented Temporal Deontic Defeasible Logic in the sense that conclusions get similarly tagged. The proof not provided here is similar to the one in [91]. It follows that for any defeasible theory, no argument is both justified and rejected, and thus no literal is both justified and rejected. Eventually, if the set $JArgs_D$ of justified arguments contains two arguments with conflicting conclusions then both arguments are strict. That is, inconsistent conclusions can be reached only when the strict part of the theory is inconsistent.

10.4 A Prolog meta-program

In this Section, we describe a Prolog meta-program $\mathscr{M}^t$ which is an extension of a meta-program $\mathscr{M}$ for basic defeasible logic [12].

$\mathscr{M}^t$ implements a fragment of the temporal modal defeasible logic provided in this Chapter. The restrictions concern the language and defeasible theories.

Definition 10.31 (Language). *Let $\mathscr{T} = \mathbb{N}$,* Prop *be a set of propositional atoms,* Ag *a finite set of agents,* Lab *be a set of labels, $n \in \mathbb{N}^*$, and $m, k, m', k' \in \{0, 1\}$. The sets below are defined as the smallest sets closed under the following rules:*

Modal operators

$$\mathrm{Mod} = \{\mathrm{Hold}_{ag}, \mathrm{Des}_{ag}, \mathrm{Bring}_{ag}, \mathrm{Obl}_{ag} \,|\, ag \in Ag\}$$

Punctual intervals

$$\mathrm{Inter} = \{[t,t] \,|\, t \in \mathscr{T}\}$$

Temporal modal operators

$$\text{TMod} = \{\text{Hold}^T_{ag}, \text{Des}^T_{ag}, \text{Bring}^T_{ag}, \text{Obl}^T_{ag} \,|\, T \in \text{Inter}, \, ag \in Ag\}$$

Temporal modal literals

$$\text{TMLit} = \{\neg^k X \neg^m \gamma \,|\, X \in \text{TMod}, \, \gamma \in \text{Prop}\}$$

Rules

$$\begin{aligned}
&\text{Rule}_s = \{r\colon \quad \phi_1, \ldots, \phi_n \rightarrow \psi \,|\, r \in \text{Lab}, \, \phi_1, \ldots, \phi_n, \psi \in \text{TMLit}\}\\
&\text{Rule}_d = \{r\colon \quad \phi_1, \ldots, \phi_n \Rightarrow \psi \,|\, r \in \text{Lab}, \, \phi_1, \ldots, \phi_n, \psi \in \text{TMLit}\}\\
&\text{Rule}_{dft} = \{r\colon \quad \phi_1, \ldots, \phi_n \leadsto \psi \,|\, r \in \text{Lab}, \, \phi_1, \ldots, \phi_n, \psi \in \text{TMLit}\}\\
&\text{Rule} = \text{Rule}_s \cup \text{Rule}_d \cup \text{Rule}_{dft}
\end{aligned}$$

Persistence rules

$$\begin{aligned}
\text{Rule}^{pers}_s = \{r\colon \quad & \neg^k X^{[t,t]}_{ag} \neg^m \gamma \rightarrow \neg^k X^{[t+1,t+1]}_{ag} \neg^m \gamma\\
& |\, r \in \text{Lab}, X \in \text{Mod}, \, ag \in Ag, \, t \in \mathscr{T}, \, \gamma \in \text{Prop}\}\\
\text{Rule}^{pers}_d = \{r\colon \quad & \neg^k X^{[t,t]}_{ag} \neg^m \gamma \Rightarrow \neg^k X^{[t+1,t+1]}_{ag} \neg^m \gamma\\
& |\, r \in \text{Lab}, X \in \text{Mod}, \, ag \in Ag, \, t \in \mathscr{T}, \, \gamma \in \text{Prop}\}\\
\text{Rule}^{pers} = \text{Rule}^{pers}_s & \cup \text{Rule}^{pers}_d
\end{aligned}$$

Conflict relations

$$\begin{aligned}
\text{Conflict} = \{&\text{conflict}(\neg^k X^{[t,t]}_{ag} \neg^m \gamma, \neg^{k'} Y^{[t,t]}_{ag} \neg^{m'} \beta)\\
&|\, X \in \text{Mod}, \, ag \in Ag, t \in \mathscr{T}, \, \gamma, \beta \in \text{Prop}\}
\end{aligned}$$

Defeat relations

$$\begin{aligned}
\text{Defeat} = \{&\text{defeat}(\neg^k X^{[t,t]}_{ag} \neg^m \gamma, \neg^{k'} Y^{[t,t]}_{ag} \neg^{m'} \beta)\\
&|\, X \in \text{Mod}, \, ag \in Ag, t \in \mathscr{T}, \, \gamma, \beta \in \text{Prop}\}
\end{aligned}$$

Superiority relations

$$\text{Sup} = \{\text{Hold}^T_{ag}(s \succ r) \,|\, ag \in \text{Ag}, \, T \in \text{Inter}, \, s, r \in \text{Lab}\}$$

Definition 10.32 (Defeasible Theory). A defeasible theory is a structure $D = (\mathscr{T}, F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$ where

- $\mathscr{T}$ is a discrete totally ordered set of instants of time,
- $F \subseteq \text{TMLit}$ is a finite set of facts,
- $R = R_1 \cup R_2$ is a finite set of rules such that each rule has an unique label, and where:

$$\begin{aligned}
&R_1 \subseteq \text{Rule} \cup \text{Rule}^{pers},\\
&R_2 = \{r_{\text{Bring}^{[t,t]}_{ag} \gamma \Rightarrow \text{Bring}^{[t,t]}_{ag} \gamma}\colon \quad \text{Bring}^{[t,t]}_{ag} \gamma \Rightarrow \text{Hold}^{[t,t]}_{ag} \gamma,\\
&\qquad r_{\text{Bring}^{[t,t]}_{ag} \gamma \Rightarrow \text{Bring}^{[t,t]}_{ag} \gamma}\colon \quad \text{Bring}^{[t,t]}_{\text{obj}} \gamma \Rightarrow \text{Hold}^{[t,t]}_{\text{obj}} \gamma\},
\end{aligned}$$

- $\mathscr{S} \subseteq \text{Sup}$ is a set of acyclic superiority relations which hold at any time, and $\forall r, s,\ r \in R - \text{Rule}^{pers},\ s \in R \cup \text{Rule}^{pers}$ we have $\text{Hold}^{T}_{ag}(r \succ s)$, $\forall r, s,\ r, s \in R \cup \text{Rule}^{pers}$ we have $\text{Hold}^{T}_{ag}(r \succ s)$ and $\text{Hold}^{T}_{ag}(s \succ r)$,
- $Ag = Ag_1 \cup \{\text{obj}\}$ is a set of agents such that $\forall a \in Ag_1, a \neq \text{obj}$,
- $\mathscr{C} \subseteq \text{Conflict}$ is a set of conflict relations which holds at any time,
- $\mathscr{D} \subseteq \text{Defeat}$ is a set of defeat relations which holds at any time.

Furthermore, we assume the absence of cycles in a defeasible theory D. Rules are in force and in efficacy at any time for any agent. As abbreviation, punctual intervals $[t,t]$ and $[t+1,t+1]$ are denoted t and $t+1$, respectively.

$\mathscr{M}^t$ is thus a simple meta-program to be considered as a basis for further extensions. $\mathscr{M}^t$ consists of the following clauses. We have permitted ourselves some syntactic flexibility and removed temporal functions (allowing displaced effects for example) in order to lighten the presentation. We have three clauses defining classes of rules:

```
rule(R, Head, supportive, Body) :-
rule(R, Head, strict, Body).

rule(R, Head, supportive, Body) :-
rule(R, Head, defeasible, Body).

persistent(R) :-
rule(R, mlit(mod(X, Ag, T+1), Lit), _, [mlit(mod(X, Ag, T), Lit)]).
```

The two first clauses indicate that strict and defeasible rules are supportive rules. The last clause defines persistence rules. Next we define the strict and defeasible applicability of rules which are not persistence rules.

```
definitelyApplicable(rule(R, X, Type, Body)):-
rule(R, X, Type, Body),
not persistent(R),
permutation(Body, Body2),
definitelybody(Body2).

definitelybody([]).
definitelybody([H|T]) :-
definitely(H),
definitelybody(T).

defeasiblyApplicable(rule(R, X, Type, Body)):-
rule(R, X, Type, Body),
not persistent(R),
permutation(Body, Body2),
defeasiblybody(Body2).
defeasiblybody([]).
defeasiblybody([H|T]) :-
defeasibly(H),
defeasiblybody(T).
```

Definite provability is defined by the following clauses. A modal literal `mlit(mod(X, Ag, T), Lit)` is definitely provable if it is a fact (first clause), or if there exists a definitely applicable strict rule with consequent `mlit(mod(X, Ag, T), Lit)` (remaining clauses). The second and third clause cater for non persistence rules, the fourth and fifth clause for persistence rules.

```
definitely(X) :-
fact(X).

definitely(mlit(mod(X, Ag, T), Lit)) :-
ground(T),
definitelyApplicable(rule(_, mlit(mod(X, Ag, T), Lit), strict, _)).

definitely(mlit(mod(X, Ag, T2), Lit)) :-
not ground(T2),
definitelyApplicable(rule(_, mlit(mod(X, Ag, T), Lit), strict, _)),
T2 is T.

definitely(mlit(mod(X, Ag, T2), Lit)) :-
rule(R, mlit(mod(X, Ag, T+1), Lit), strict, _),
persistent(R),
definitelyApplicable(strict(_, mlit(mod(X, Ag, T), Lit), _)),
computepers(T, T2).

definitely(mlit(mod(X, Ag, T2), Lit)) :-
rule(R, mlit(mod(X, Ag, T+1), Lit), strict, _),
persistent(R),
fact(mlit(mod(X, Ag, T), Lit)),
computepers(T, T2).
```

Defeasible provability is defined by the following clauses. A modal literal `mlit(mod(X, Ag, T), Lit)` is defeasibly provable if it is definitely provable (first clause), or if (i) there exists a defeasible applicable strict or defeasible rule with consequent `mlit(mod(X, Ag, T), Lit)` which is not defeated (remaining clauses). The second and third clause cater for non persistence rules, the fourth and fifth clause for persistence rules.

```
defeasibly(X) :- definitely(X).

defeasibly(mlit(mod(X, Ag, T), Lit)) :-
ground(T),
defeasiblyApplicable(rule(_, mlit(mod(X, Ag, T), Lit), supportive, _)),
conflict(Lit, NonLit),
not definitely(mlit(mod(X, Ag, T), NonLit)),
not attacked(mlit(mod(X, Ag, T), Lit)).

defeasibly(mlit(mod(X, Ag, T2), Lit)) :-
not ground(T2),
defeasiblyApplicable(rule(_, mlit(mod(X, Ag, T), Lit), supportive, _)),
T2 is T,
conflict(Lit, NonLit),
not definitely( mlit(mod(X, Ag, T2), NonLit)),
not attacked(mlit(mod(X, Ag, T2), Lit)).

defeasibly(mlit(mod(X, Ag, T2), Lit)) :-
rule(R, mlit(mod(X, Ag, T+1), Lit), supportive, _),
persistent(R),
defeasiblyApplicable(rule(_, mlit(mod(X, Ag, T), Lit), supportive, _)),
computepers(T, T2),
not broken(mlit(mod(X, Ag, T1), Lit), mlit(mod(X, Ag, T2), Lit)).

defeasibly(mlit(mod(X, Ag, T2), Lit)) :-
rule(R, mlit(mod(X, Ag, T+1), Lit), defeasible, _),
persistent(R),
fact(mlit(mod(X, Ag, T), Lit)),
computepers(T, T2),
not broken(mlit(mod(X, Ag, T), Lit), mlit(mod(X, Ag, T2), Lit)).
```

The following clauses indicate if and how a rule can be attacked and specify the defeat of attacking rules by stronger counter-attacking rules.

```
attacked(mlit(mod(X, Ag, T), Lit)) :-
conflict(mlit(mod(X, Ag, T), Lit), mlit(mod(Y, Ag, T), NonLit)),
defeasiblyApplicable(rule(S, mlit(mod(Y, Ag, T), NonLit), _, _)),
not defeated(S, mlit(mod(Y, Ag, T), NonLit), mlit(mod(X, Ag, T), Lit)).

broken(mlit(mod(X, Ag, T1), Lit), mlit(mod(X, Ag, T2), Lit)) :-
conflict(mlit([mod(X, Ag, _)], Lit), mlit([mod(Y, Ag, _)], NonLit)),
defeasiblyApplicable(rule(S, mlit(mod(Y, Ag, T), NonLit), _, _)),
T>=T1,
T=<T2,
not defeated(S, mlit(mod(Y, Ag, T), NonLit), mlit(mod(X, Ag, T), Lit)).

broken(mlit(mod(X, Ag, T1), Lit), mlit(mod(X, Ag, T2), Lit)) :-
conflict(mlit(mod(X, Ag, _), Lit), mlit(mod(Y, Ag, _), NonLit)),
fact(mlit(mod(Y, Ag, T), NonLit)),
T>=T1,
T=<T2.
```

```
defeated(S, mlit(mod(Y, Ag, T), NonLit), mlit(mod(X, Ag, T), Lit)) :-
defeat(mlit(mod(X, Ag, _), Lit), mlit(mod(Y, Ag, _), NonLit)),
defeasiblyApplicable(rule(W, mlit(mod(X, Ag, T), Lit), supportive, _)),
not mlit(mod(hold, Ag, _), sup(S, W)).

defeated(S, mlit(mod(_, Ag, T), _), mlit(mod(X, Ag, T), Lit)) :-
mlit(mod(hold, Ag, _), sup(W, S)),
defeasiblyApplicable(rule(W, mlit(mod(X, Ag, T), Lit), supportive, _)).
```

The auxiliary predicate `computepers(T1, T2)` is defined according whether the time instant T2 is grounded or not.

```
computepers(T1, T2) :-
ground(T2),
T2 >= T1.

computepers(T1, T2) :-
not ground(T2),
T2 is T1.
```

Finally, given a defeasible theory $D = (\mathscr{T}, F, R, \mathscr{S}, Ag, \mathscr{C}, \mathscr{D})$, the domain dependent clauses of the meta-program are introduced as follows.

- For any $X^T_{ag}\gamma \in F$:
  fact(mlit(mod(X, Ag, T), γ)).
- For any strict rule r: $\phi_1, \ldots, \phi_n \rightarrow \psi \in R$:
  rule(r, ψ, strict, [ϕ_1,..., ϕ_n]).
- For any defeasible rule r: $\phi_1, \ldots, \phi_n \Rightarrow \psi \in R$:
  rule(r, ψ, defeasible, [ϕ_1,..., ϕ_n]).
- For any defeater r: $\phi_1, \ldots, \phi_n \leadsto \psi \in R$:
  rule(r, ψ, defeater, [ϕ_1,..., ϕ_n]).
- For any conflict relation conflict$(X^T_{ag}\gamma, Y^T_{ag}\beta) \in \mathscr{D}$:
  conflict(mlit(mod(X, Ag, T), γ), mlit(mod(Y, Ag, T), β)).
- For any conflict relation defeat$(X^T_{ag}\gamma, Y^T_{ag}\beta) \in \mathscr{D}$:
  defeat(mlit(mod(X, Ag, T), γ), mlit(mod(Y, Ag, T), β)).
- For any superiority relation $X^T_{ag}(\mathrm{r} \succ \mathrm{s}) \in \mathscr{S}$:
  mlit(mod(X, Ag, T), sup(r,s)).

For example, consider the provision indicating that customers must pay within 30 days after receiving the invoice. This example can be formalized by the following rules:

$$\begin{array}{ll} r_{init}: & \mathrm{Hold}^t_{ag} get\_invoice(ag) \Rightarrow \mathrm{Obl}^t_{ag} pay(ag), \\ r_{pers}: & \mathrm{Obl}^t_{ag} pay(ag) \Rightarrow \mathrm{Obl}^{t+1}_{ag} pay(ag), \\ r_{term}: & \mathrm{Hold}^t_{ag} pay(ag) \leadsto \neg \mathrm{Obl}^t_{ag} pay(ag), \\ r_{viol}: & \mathrm{Hold}^t_{ag} get\_invoice(ag), \mathrm{Obl}^{t+30}_{ag} pay(ag) \Rightarrow \mathrm{Hold}^t_{ag} violation. \end{array}$$

The obligation to pay is triggered by receipt of the invoice (rule r_{init}). The obligation persists (rule r_{pers}) and terminates when it is complied with (rule r_{term}). The termination is modeled by using a defeater rule. If the customer has not paid within 30 days after receiving the invoice then a violation is triggered (rule r_{viol}). Note that the obligation persists after the deadline. In the prolog meta-program, the rules given above are captured by the following:

```
rule(rinit, mlit(mod(obl, Ag, T), pay(Ag)), defeasible,
             [mlit(mod(hold, Ag, T), get_invoice(Ag))]).
rule(rpers, mlit([mod(obl, Ag, plus(T,1))], pay(Ag)), defeasible,
             [mlit(mod(obl, Ag, T), pay(Ag))]).
rule(rterm, mlit([mod(non(obl), Ag, plus(T,1))], pay(Ag)), defeater,
             [mlit(mod(hold, Ag, T), pay(Ag))]).
rule(rviol, mlit([mod(hold, Ag, plus(T, 30))], violation), defeasible,
             [mlit(mod(hold, Ag, T), get_invoice(Ag)),
             mlit([mod(obl, Ag, plus(T, 30))], pay(Ag))]).
```

Suppose that we have the fact $\mathrm{Hold}^{0}_{\mathrm{mario}}\mathit{get\_invoice}(\mathrm{mario})$:

```
fact(mlit(mod(hold, mario, 0), get_invoice(mario))).
```

If we make the query `defeasibly(mlit(mod(X, Ag, 30), Lit))`, then we obtain:

```
defeasibly(mlit(mod(hold, mario, 30), violation(mario))).
defeasibly(mlit(mod(obl, mario, 30), pay(mario))).
```

In other words, a violation is triggered at time 30 because the customer did not pay, and the obligation to pay still exists. If we have the fact $\mathrm{Hold}^{0}_{\mathrm{mario}}\mathit{get\_invoice}(\mathrm{mario})$ and $\mathrm{Hold}^{15}_{\mathrm{mario}}\mathit{pay}(\mathrm{mario})$:

```
fact(mlit(mod(hold, mario, 0), get_invoice(mario))).
fact(mlit(mod(hold, mario, 15), pay(mario))).
```

then the query `defeasibly(mlit(mod(X, Ag, 30), Lit))` cannot be satisfied because neither the obligation nor the violation holds at time 30.

11

Conclusions

The motivation of this dissertation stems from the need for sustainable computer systems in unpredictable environments. We assume that the paradigm of autonomous software agents is a solution to build such sustainable computer systems, and acknowledge that norms can be used for controlling autonomous software agents while respecting their autonomy. Cognitive agents form the basis for modeling normative bindingness.

In this Chapter, we discuss the main results of this research, some prospects and limits.

11.1 Results

The main contribution of this dissertation is the development of a formal framework in temporal modal defeasible logic to account for the interactions between software cognitive agents and normative systems.

Basic defeasible logic was extended with (sequence of) modalities for epistemic, practical and normative reasoning. As conflicts may appear amongst mental states and norms, the combination of modalities with the defeasible features of the logic accounts for the bindingness of norms with respect to autonomous cognitive agents. To some extend, the framework is also a contribution of computer science to model normative bindingness.

The proposed modal defeasible logic was at its turn extended to relate some inherent temporal aspects of dynamic environments. The extension was made by associating temporal intervals to modalities. As we admitted sequences of modalities, our framework caters for temporal viewpoints. A special attention was given to some temporal aspects of the legal domain as the time of force and the time of efficacy, retroactive and ultra-active provisions. Bearing in mind the law of parsimony "entia non sunt multiplicanda praeter necessitatem", our temporal account of legal reasoning was related to a common-sense temporal account of cognition: doing the introduction of new logic entities was limited. Some temporal and textual normative modifications were also investigated, and their integration in the framework discussed.

Finally, a Prolog meta-program of a fragment of temporal modal defeaible logic was proposed.

11.2 Prospects and limits

This dissertation is part of an ongoing effort in norm-governed computer systems to provide an appropriate formalization of the interaction between software cognitive agents and normative systems. In this view, many possible extensions of the proposed framework come to mind, and we shall suggest only those closely correlated to the present work.

At the normative level, we have limited our framework to basic deontic operators for obligations, prohibition, (strong) permissions, facultativeness. A prospect is to enrich this set of normative notions, for example by the integration of weak permissions, rights including obligative rights, permissive rights, ergo-omnes rights and exclusionary rights, or legal powers, enabling powers, potestive rights, declarative powers etc. (see e.g. [191, 192]). The notion of role can be also integrated.

Concerning temporal aspects, while we focused on absolute time reference using intervals of time instants, a challenging prospect is the integration of relative time reference (e.g. ‘before’, ‘after’ etc.) and temporal constraints. For example, suppose a scheduling agent having to comply with a medical prescription as given below [195]:

> A patient requires three medical exams, each followed within 12 hours by a treat ment session. Exams and treatments cannot overlap. Both are completed within 4 hours and must be at least 8 hours apart. The exams require resources available from the 8th to the 12ve and from the 20th to the 21st of the month.

Associated with some other possible normative constraints, some example of queries (as a slight defeasible adaptation of [195]) are: find a (de)feasible schedule (if any), find all (de)feasible schedules, what are the (de)feasible times for an exam or a treatment?, what are the (de)feasible relations between two exams or treatments?, what are the (de)feasible relations between all exams and treatments?

Beside temporal aspects, we did not account for spatial dimensions. The spatial dimensions are important if we acknowledge that the application of norms is usually spatially grounded to specific locations. For example, French legislation usually applies on the French territory, some private policies applies on particular working places, etc.

Also, in the current framework, priorities between rules, conflict and defeat relations among statements to characterize agent profiles are captured at the inference level. An interesting prospect is a setting at the theory level to develop dynamic priorities among rules (see [7]) and dynamic agent profiles. Dynamic priorities could be appropriate to capture for example a combination of time-based priority (lex posterior derogat legi priori), hierarchy-based priority (lex superior derogat legi inferiori), and specificity-based priority (lex specialis derogat legi generali).

Beside conceptual extensions of the framework, a challenging and important prospect concerns implementation matters.

As a Prolog meta-program to emulate a fragment of temporal defasible logic was proposed, its extension to integrate other notions as legal temporal dimensions, intervals and combinations of modalities is a prospect.

A Prolog meta-program is a possibility of implementation among others. If our intention is to build a forward reasoning engine, then we can adopt the algorithm suggested by [130] by considering snapshots of the theory at any temporal position to built an equivalent ground instantiated theory. Based on the ground instantiated theory, we can apply the algorithm of [130] to compute in linear time the extension. However, this solution is hardly acceptable since it involves a combinational explosion of the number of grounded instances. In other words, the amounts of resources required for the execution of such solution to compute the extension of a theory do not permit an efficient implementation. To slightly overcome the issue, we can consider some restrictions on the language, for example, by considering only instant intervals so that the linear complexity is ‘only’ augmented by a factor t equals to the number of time instants. An efficient algorithm has still to be invented.

Beside technical issues, bearing in mind the proverb “Science sans conscience n’est que ruine de l’âme” by F. Rabelais, we close this dissertation on some ethical issues. Is our field of research compatible with ethical demands? Numerous science fiction stories nourished worries and fantasies on autonomous artificial agents. Some people could claim that there is no matter of urgency. Arguably, there is less urgency to deal with autonomous artificial agents than with genetic engineering. Though most of dangers are faraway from reality, the use of agent softwares has already generate legal issues (see e.g. [1]).

For example, as norms often deal with ethical issues, is it wise to delegate artificial agents ethical tasks? or is human discretion needed? A primer issue concerns for instance the responsibility for violations or errors made by software agents. What are the future impacts of such technologies in human beings’ societies? Nowadays, such impacts are hardly predictable.

References

1. Report on legal issues of software agents, Deliverable 14, 2006. Legal Issues for the Advancement of Information Society Technologies (IST-2-004252-SSA).
2. C. E. Alchourrón and E. Bulygin. *Normative Systems*. Springer, 1971.
3. H. Aldewereld, F. Dignum, A. García-Camino, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra. Operationalisation of norms for usage in electronic institutions. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 223–225. ACM Press, 2006.
4. V. Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.
5. J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
6. L. Allen. Symbolic logic: A razor edge tool for drafting and interpreting legal documents. *Yale Law Journal*, 66:833–879, 1957.
7. G. Antoniou. Defeasible logic with dynamic priorities. *International Journal Intelligent Systems*, 19(5):463–472, 2004.
8. G. Antoniou and M. Arief. Modelling business rules using defeasible logic. In *Proceedings of the 2000 Information Resources Management Association International Conference on Challenges of Information Technology Management in the 21st Century*, pages 1020–1022. IGI Publishing, 2000.
9. G. Antoniou and A. Bikakis. DR-Prolog: A system for defeasible reasoning with rules and ontologies on the semantic web. *IEEE Transaction on Knowledge and Data Engineering*, 19(2):233–245, 2007.
10. G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. A flexible framework for defeasible logics. In *Proceedings of the 17th American National Conference on Artificial Intelligence*, pages 401–405. AAAI/MIT Press, 2000.
11. G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255–287, 2001.
12. G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. Embedding defeasible logic into logic programming. *Theory and Practice of Logic Programming*, 6(6):703–735, 2006.
13. G. Antoniou, D. Billington, G. Governatori, M. J. Maher, and A. Rock. A family of defeasible reasoning logics and its implementation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 459–463. IOS Press, 2000.

14. G. Antoniou, D. Billington, and M. J. Maher. On the analysis of regulations using defeasible rules. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, 1999.
15. G. Antoniou, N. Dimaresis, and G. Governatori. A system for modal and deontic defeasible reasoning. In *Proceedings of the 20th Australian Conference on Artificial Intelligence*, pages 609–613. Springer, 2007.
16. G. Antoniou, M. J. Maher, and D. Billington. Defeasible logic versus logic programming without negation as failure. *Journal of Logic Programming*, 42(1):47–57, 2000.
17. G. Antoniou, M. J. Maher, D. Billington, and G. Governatori. A comparison of sceptical naf-free logic programming approaches. In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 347–356. Springer, 1999.
18. K. R. Apt and R. N. Bol. Logic programming and negation: A survey. *Journal of Logic Programming*, 19/20:9–71, 1994.
19. A. Artikis, J. Pitt, and M. J. Sergot. Animated specifications of computational societies. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1053–1061. ACM Press, 2002.
20. K. D. Ashley. *Modelling legal argument: reasoning with cases and hypotheticals*. PhD thesis, University of Massachusetts, 1988.
21. N. Bassiliades, G. Antoniou, and G. Governatori. Proof explanation in the DR-DEVICE system. In *Proceedings of the 1st International Conference on Web Reasoning and Rule Systems*, pages 249–258. Springer, 2007.
22. N. Bassiliades, G. Antoniou, and I. Vlahavas. DR-DEVICE: A defeasible logic system for the Semantic Web. In *2nd Workshop on Principles and Practice of Semantic Web Reasoning*, pages 134–148. Springer, 2004.
23. N. Bassiliades, G. Antoniou, and I. P. Vlahavas. A defeasible logic reasoner for the semantic web. *International Journal Semantic Web Information Systems*, 2(1):1–41, 2006.
24. F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. Jade a white paper, September 2003.
25. T. Bench-Capon. Deep models, normative reasoning and legal expert systems. In *Proceedings of the 2nd International Conference on Artificial Intelligence and Law*, pages 37–45. ACM Press, 1989.
26. T. Bench-Capon and G. Sartor. Theory based explanation of case law domains. In *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 12–21. ACM Press, 2001.
27. T. Bench-Capon and G. Sartor. Using values and theories to resolve disagreement in law. In *Proceedings of the The Thirteenth Annual Conference on Legal Knowledge and Information Systems*, pages 73–84. IOS Press, 2001.
28. T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1-2):97–143, 2003.
29. T. Bench-Capon and P. R. S. Visser. Open texture and ontologies in legal information systems. In *Proceedings of the 8th International Workshop on Database and Expert Systems Applications*, page 192. IEEE Computer Society, 1997.
30. F. Benigni. *Metodi di marcatura per il consolidamento degli atti normativi*. PhD thesis, Università degli Studi di Bologna, 2006.
31. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web (berners-lee et. al 2001). *Scientific American*, May 2001.
32. D. Billington. Defeasible logic is stable. *Journal of Logic and Computation*, 3(4):379–400, 1993.

33. N. Bobbio. Norma. *Enciclopedia Einaudi*, 9:876–907, 1980.
34. G. Boella and L. van der Torre. Permissions and obligations in hierarchical normative systems. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 109–118. ACM Press, 2003.
35. G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. In *Proceedings of the Dagstuhl Seminar on Normative Multi-agent Systems*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), 2007.
36. G. Boella, L. W. N. van der Torre, and H. Verhagen. Normative multi-agent systems. In *Normative Multi-agent Systems*, volume 07122 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.
37. A. Boer, T. Gordon, K. van den Berg, M. di Bello, A. Förhécz, and R. Vas, 2007. Specification of the legal knowledge interchange format. Deliverable 1.1, European project for Standardized Transparent Representations in order to Extend Legal Accessibility (ESTRELLA, IST-4-027655).
38. A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.
39. R. Borruso, R. M. D. Giorgi, L. Mattioli, and M. Ragona. *L'informatica del diritto*. Giuffrè, 2004.
40. K. Branting. An agenda for empirical research in ai and law. In *Proceedings of the Workshop on Evaluation of Legal Reasoning and Problem-Solving Systems*, pages 28–35. ACM Press, 2003.
41. I. Bratko. *Prolog (3rd ed.): programming for artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc., 2001.
42. M. E. Bratman. *Intentions, Plans and Practical Reason*. Harvard University Press, 1987.
43. G. Brewka and T. Eiter. Prioritizing default logic. In *Intellectics and Computational Logic (to Wolfgang Bibel on the occasion of his 60th birthday)*, pages 27–45. Kluwer, 2000.
44. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. W. N. van der Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 9–16. ACM Press, 2001.
45. J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the BOID architecture. *Cognitive Science Quarterly*, 2(3-4):428–447, 2002.
46. J. Broersen, M. Dastani, and L. W. N. van der Torre. Resolving conflicts between beliefs, obligations, intentions, and desires. In *Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 568–579. Springer-Verlag, 2001.
47. R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, 2(1):14–23, 1986.
48. E. Bulygin. Permissive norms and normative systems. In *Automated Analysis of Legal Texts*, pages 211–218. Publishing Company, 1986.
49. E. Bulygin. On logic in the law: Something, but not all. *Ratio Juris*, 21(1):150–156, March 2008.
50. M. Cadoli and M. Schaerf. A survey of complexity results for nonmonotonic logics. *Journal of Logic Programming*, 17(2/34):127–160, 1993.
51. M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.
52. J. Carmo and A. J. Jones. Deontic logic and contrary-to-duties. *In Handbook of Philosophical Logic (2nd edition)*, 8, 2000.

53. C. Castelfranchi. Slides of the 10th International Conference on Artificial Intelligence and Law, 2005.
54. R. Chisholm. *Perceiving*. Princeton University Press, 1957.
55. R. Chisholm. *Theory of Knowledge*. Prentice-Hall, 1966.
56. K. L. Clark. Negation as failure. In *Readings in Nonmonotonic Reasoning*, pages 311–325. Morgan Kaufmann, 1987.
57. R. Conte and C. Castelfranchi. *Cognitive and social action*. University College of London Press, 1995.
58. R. Conte, R. Falcone, and G. Sartor. Introduction: Agents and norms: How to fill the gap? *Artificial Intelligence and Law*, 7(1):1–15, 1999.
59. M. Dastani, G. Governatori, A. Rotolo, and L. van der Torre. Preferences of agents in defeasible logic. In *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, pages 695–704. Springer, 2005.
60. M. Dastani, G. Governatori, A. Rotolo, and L. van der Torre. Programming cognitive agents in defeasible logic. In *Proceedings of 11th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 621–636. Springer, 2005.
61. M. Dastani and L. W. N. van der Torre. Programming BOID-plan agents: Deliberating about conflicts among defeasible mental attitudes and plans. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 706–713. IEEE Computer Society, 2004.
62. D. Dennett. *The intentional stance*. Bradford/MIT Press, 1987.
63. F. Dignum, D. Morley, L. Sonenberg, and L. Cavedon. Towards socially sophisticated BDI agents. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, pages 111–118. IEEE Computer Society, 2000.
64. V. Dignum, J. Vázquez-Salceda, and F. Dignum. OMNI: Introducing social structure, norms and ontologies into agent organizations. In *Proceedings of the 2nd International Workshop Programming Multi-Agent Systems*, pages 181–198. Springer, 2004.
65. N. Dimaresis and G. Antoniou. Implementing modal extensions of defeasible logic for the semantic web. In *Proceedings of the 22nd American National Conference on Artificial Intelligence*, pages 1848–1849. AAAI Press, 2007.
66. Y. Dimopoulos and A. C. Kakas. Logic programming without negation as failure. In *Proceedings of the 1995 International Symposium on Logic Programming*, pages 369–383. MIT Press, 1995.
67. M. d'Inverno, D. Kinny, M. Luck, and M. Wooldridge. A formal specification of dMARS. In *Proceedings of the 4th International Workshop on Agent Theories, Architectures and Languages*, volume 1365, pages 155—176. Springer-Verlag, 1998.
68. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.
69. J. M. Epstein and R. Axtell. *Growing artificial societies*. MIT Press, 1996.
70. M. Esteva, D. de la Cruz, B. Rosell, J. L. Arcos, J. A. Rodríguez-Aguilar, and G. Cuní. Engineering open multi-agent systems as electronic institutions. In *Proceedings of the 19th American National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence*, pages 1010–1011. AAAI/MIT Press, 2004.
71. M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specifications of electronic institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective.*, pages 126–147. Springer-Verlag, 2001.
72. M. Esteva, B. Rosell, J. A. Rodríguez-Aguilar, and J. L. Arcos. AMELI: An agent-based middleware for electronic institutions. In *Proceedings of the 3rd International*

Joint Conference on Autonomous Agents and Multiagent Systems, pages 236–243. IEEE Computer Society, 2004.

73. I. A. Ferguson. Integrated control and coordinated behaviour: A case for agent models. In *Proceedings of the European Conference on Artificial Intelligence Workshop on Agent Theories, Architectures, and Languages*, volume 890, pages 203–218. Springer, 1994.
74. T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In *Proceedings of the 3rd International Conference on Information and Knowledge Management*, pages 456–463. ACM Press, 1994.
75. FIPA. www.fipa.org, 2007.
76. M. Fisher. A survey of concurrent metatem - the language and its applications. In *Proceedings of the 1st International Conference on Temporal Logic*, pages 480–505. Springer-Verlag, 1994.
77. I. T. Foster. The anatomy of the grid: Enabling scalable virtual organizations. In *Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*, pages 1–4. Springer-Verlag, 2001.
78. S. Franklin and A. Graesser. Is it an agent or just a program?: A taxonomy for autonomous agents. In *Proceedings of the 3rd International Workshop on Agent Theories, Architectures and Languages*. Springer-Verlag, 1996.
79. B. G., D. J., and K. K. *Nonmonotonic Reasoning: An Overview*. CSLI Publications, 1997.
80. D. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. *Logics and models of concurrent systems*, pages 439–457, 1985.
81. C. Garion and L. Cholvy. Deriving individual obligations from collective obligations. In *Proceedings of the Dagstuhl Seminar on Normative Multi-agent Systems*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), 2007.
82. J. Gelati, G. Governatori, A. Rotolo, and G. Sartot. Declarative power, representation, and mandate. a formal analysis. In *Proceedings of the 15h Annual Conference on Legal Knowledge and Information Systems*, pages 41–52. IOS Press, 2002.
83. M. Gelfond and V. Lifschitz. Logic programs with classical negation. *Logic programming*, pages 579–597, 1990.
84. M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Stanford University, 1992.
85. M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *Proceedings of the 6th American National Conference on Artificial Intelligence*, pages 677–682. AAAI Press, 1987.
86. T. F. Gordon. The pleadings game: an exercise in computational dialectics. *Artificial Intelligence and Law*, 2(4):239–292, 1994.
87. T. F. Gordon. *The Pleadings Game An Artificial Intelligence Model of Procedural Justice*. Kluwer, 1995.
88. G. Governatori. Representing business contracts in ruleml. *International journal of cooperative information System*, 14:181–216, 2005.
89. G. Governatori, J. Hulstijn, R. Riveret, and A. Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Proceedings of the 20th Australian Conference on Artificial Intelligence*, pages 486–496. Springer, 2007.
90. G. Governatori and M. J. Maher. An argumentation-theoretic characterization of defeasible logic. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 469–473. IOS Press, 2000.
91. G. Governatori, M. J. Maher, D. Billington, and G. Antoniou. Argumentation semantics for defeasible logics. *Journal of Logic and Computation*, 14(5):675–702, 2004.

92. G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Norm modifications in defeasible logic. In *Proceedings of the 18th International Conference on Legal Knowledge and Information Systems*, pages 13–22. IOS Press, 2005.
93. G. Governatori and A. Rotolo. Defeasible logic: Agency, intention and obligation. In *Proceedings of the 7th International Workshop on Deontic Logic in Computer Science*, number 3065 in LNAI, pages 114–128. Springer-Verlag, 2004.
94. G. Governatori, A. Rotolo, R. Riveret, M. Palmirani, and G. Sartor. Variants of temporal defeasible logics for modelling norm modifications. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 155–159. ACM Press, 2007.
95. G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, pages 25–34. ACM Press, 2005.
96. G. Governatori and P. Terenziani. Temporal extensions to defeasible logic. In *Proceedings of the 20th Australian Conference on Artificial Intelligence*, pages 476–485. Springer, 2007.
97. B. N. Grosof. Prioritized conflict handling for logic programs. In *ILPS*, pages 197–211, 1997.
98. S. Haack. *Philosophy of Logics*. Cambridge University Press, 1978.
99. S. Haack. On logic in the law: Something, but not all. *Ratio Juris*, 20(1):1–31, March 2007.
100. S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33(3):379–412, 1987.
101. J. Hansen, G. Pigozzi, and L. van der Torre. Ten philosophical problems in deontic logic. In *Proceedings of the Dagstuhl Seminar on Normative Multi-agent Systems*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), 2007.
102. H. L. Hart. *The Concept of Law*. Oxford University Press, 1961.
103. J. Hintikka. *Knowledge and Belief*. Cornell Univeristy Press, 1962.
104. C. Hohfeld. Fundamental legal conceptions as applied in judicial reasoning. *Yale Law Journal*, 1913.
105. C. Iglesias, M. Garrijo, and J. Gonzalez. A survey of agent-oriented methodologies. In *Proceedings of the 5th International Workshop on Intelligent Agents: Agent Theories, Architectures, and Languages*, volume 1555, pages 317–330. Springer-Verlag, 1999.
106. A. J. Jones and M. Sergot. The characterisation of law and computer systems: the normative systems perspective. In *Proceedings of the 1st International Workshop on Deontic Logic in Computer Science: Normative System Specification.*, pages 275–307. John Wiley and Sons, 1993.
107. A. J. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logic*, 4(3):429–445, 1996.
108. J. Jorgensen. Imperatives and logic. *Erkenntnis*, 7:288–296, 1938.
109. L. Kagal and T. Finin. Modeling conversation policies using permissions and obligations. *Autonomous Agents and Multi-Agent Systems*, 14(2):187–206, 2007.
110. A. C. Kakas, R. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.
111. S. Kanger. Law and logic. *Theoria*, 38, 1972.
112. R. Klischewski. Semantic web for e-government. In *Electronic Government*, pages 288–295. Springer, 1994.
113. M. J. Kollingbaum and T. J. Norman. NoA - a normative agent architecture. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1465–1466. Morgan Kaufmann, 2003.

114. R. Kowalski. Predicate logic as programming language. In *FIP Cong 1974*, pages 569–574. North-Holland Pub Co, 1974.
115. R. Kowalski. The early years of logic programming. *Communications of the ACM*, 31(1):38–43, 1988.
116. R. Kowalski. The treatment of negation in logic programs for representing legislation. In *Proceedings of the 2nd International Conference on Artificial Intelligence and Law*, pages 11–15. ACM Press, 1989.
117. R. Kowalski. Legislation as logic programs. In *Proceedings of the 2nd International Logic Programming Summer School on Logic Programming in Action*, pages 203–230. Springer-Verlag, 1992.
118. R. Kowalski and F. Sadri. Reconciling the event calculus with the situation calculus. *Journal of Logic Programming*, 31(1-3):39–58, 1997.
119. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.
120. S. Kraus, D. J. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
121. D. B. Lange and M. Oshima. Seven good reasons for mobile agents. *Communications ACM*, 42(3):88–89, 1999.
122. Y. Lespérance, H. J. Levesque, F. Lin, D. Marcu, R. Reiter, and R. B. Scherl. Foundations of a logical approach to agent programming. In *Proceedings of the workshop on Agent Theories, Architectures, and Languages*, pages 331–346. Springer, 1995.
123. L. Lindahl. *Position and Change. A study in Law and Logic.* Dordrecht: D. Reidel, 1977.
124. L. Loevinger. Jurimetrics. the next step forward. *Minnesota Law Review*, 33:455–493, 1949.
125. F. López y López and M. Luck. Modelling norms for autonomous agents. In *Proceedings of the 4th Mexican International Conference on Computer Science*, page 238. IEEE Computer Society, 2003.
126. F. López y López, M. Luck, and M. d'Inverno. Normative agent reasoning in dynamic societies. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 732–739. IEEE Computer Society, 2004.
127. M. Luck, R. Ashri, and M. d'Inverno. *Agent-Based Software Development*. Artech House, Inc., 2004.
128. P. Maes. Artificial life meets entertainment: lifelike autonomous agents. *Communications of the ACM*, 38(11):108–114, 1995.
129. M. J. Maher. A denotational semantics of defeasible logic. In *Proceedings of the 1st International Conference on Computational Logic*, pages 209–222. Springer, 2000.
130. M. J. Maher. Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming*, 1(6):691–711, 2001.
131. M. J. Maher. A model-theoretic semantics for defeasible logic. In *Proceedings of the Workshop on Paraconsistent Computational Logic*, volume 95, pages 67–80, 2002.
132. M. J. Maher, A. Rock, G. Antoniou, D. Billington, and T. Miller. Efficient defeasible reasoning systems. *International Journal on Artificial Intelligence Tools*, 10(4):483–501, 2001.
133. R. H. Marín and G. Sartor. Time and norms: a formalisation in the event-calculus. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, pages 90–99. ACM Press, 1999.
134. J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
135. J. McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

136. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence 4*, pages 463–502, 1969.

137. L. T. McCarty. Some arguments about legal arguments. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, pages 215–224. ACM Press, 1997.

138. P. McNamara. Deontic logic. http://plato.stanford.edu/entries/logic-deontic/ [Accessed December 29, 2007].

139. D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical report, Hewlett Packard, 2002.

140. E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, 2006.

141. J. P. Müller, M. Pischel, and M. Thiel. Modeling reactive behaviour in vertically layered agent architectures. In *Proceedings of the European Conference on Artificial Intelligence Workshop on agent theories, architectures, and languages on Intelligent agents*, pages 261–276. Springer-Verlag, 1995.

142. S. Munzer. Retroactive law. *The Journal of Legal Studies*, 6(2):373–397, 1977.

143. S. Munzer. A theory of retroactive legislation. *Texas Law Review*, 61:425–80, 1982.

144. R. Nannucci (a cura di). *Lineamenti di informatica giuridica. Teoria, Metodi, Applicazioni*. Edizioni Scientifiche Italiane, 2002.

145. U. Nilsson and J. Maluszynski. *Logic, Programming, and PROLOG*. John Wiley & Sons, 1995.

146. T. J. Norman, D. V. Carbogim, E. C. W. Krabbe, and D. N. Walton. Argument and multi-agent systems. In *Argumentation Machines*, pages 15–54. Kluwer Academic Publishers, 2004.

147. D. Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3. Oxford University Press, 1987.

148. D. Nute. Defeasible reasoning. In *Proceedings of 20th Hawaii International Conference on System Science*. IEEE press, 1987.

149. D. Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 353–395. Oxford University Press, 1993.

150. D. Nute. Norms, priorities, and defeasible logic. In *Norms, Logics and Information Systems: New Studies on Deontic Logic and Computer Science*. IOS Press, 1998.

151. J. Odell. Objects and agents compared. *Journal of Object Technology*, 1(1):41–53, 2002.

152. A. Omicini and S. Ossowski. Objective versus subjective coordination in the engineering of agent systems. *Intelligent Information Agents: The Agentlink Perspective*, 2003.

153. A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf. *Coordination of Internet Agents: Models, Technologies, and Applications*. Springer, 2001.

154. R. Pagano. *L'arte di legiferare*. Giuffré, 2001.

155. M. Palmirani. Draft of classification of the modificatory provisions for the legislative consolidation the case of africa legislative traditions.

156. M. Palmirani. Time model in normative information systems. In *Proceedings of the Workshop on The Role of Legal Knowledge in e-Government*, pages 15–26. Wolf Legal Publishers, 2005.

157. M. Palmirani and R. Brighi. Time model for managing the dynamic of normative system. In *Proceedings of 5th International Conference on Electronic Government*, pages 207–218. Springer, 2006.

158. H. V. D. Parunak. A practitioners' review of industrial agent applications. *Autonomous Agents and Multi-Agent Systems*, 3(4):389–407, 2000.

159. A. Peczenik. *On Law and Reason*. Kluwer, 1989.

160. C. Perelman and L. Olbrechts-Tyteca. *The New Rhetoric: A Treaty on Argumentation*. Trans. J. Wilkinson and P. Weaver. Notre Dame, Ind.: University of Notre Dame Press (1st ed. in French 1958.), 1969.
161. L. Philipps and G. Sartor. Introduction: From legal theories to neural networks and fuzzy reasoning. *Artificial Intelligence and Law*, 7(2-3):115–128, 1999.
162. L. R. Phillips and H. E. Link. The role of conversation policy in carrying out agent conversations. In *Issues in Agent Communication*, pages 132–143. Springer-Verlag, 2000.
163. J. L. Pollock. Criteria and our knowledge of the material world. *Philosophical Review*, 76:28–62, 1967.
164. J. L. Pollock. The structure of epistemic justification. *Monograph Series: American Philosophical Quarterly*, 4:62–78, 1970.
165. J. L. Pollock. *Knowledge and Justification*. Princeton University Press, 1974.
166. J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.
167. J. L. Pollock. *Cognitive Carpentry*. MIT Press, 1995.
168. D. Poole. On the comparison of theories: Preferring the most specific explanation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 144–147. Morgan Kaufmann, 1985.
169. D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–47, 1988.
170. H. Prakken. *Logical Tools for Modelling Legal Argument. A Study of Defeasible Reasoning in Law.* Kluwer Law and Philosophy Library, 1997.
171. H. Prakken and G. Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4(3-4):331–368, 1996.
172. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1), 1997.
173. H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6(2-4):231–287, 1998.
174. H. Prakken and G. Sartor. The role of logic in computational models of legal argument: A critical survey. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II*, pages 342–381. Springer-Verlag, 2002.
175. H. Prakken and G. Sartor. The three faces of defeasibility in the law. *Ratio Juris*, 17:118–139, 2004.
176. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.
177. R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. *Artificial intelligence and mathematical theory of computation: papers in honour of John McCarthy*, pages 359–380, 1991.
178. E. L. Rissland, K. D. Ashley, and R. P. Loui. AI and Law: A fruitful synergy. *Artificial Intelligence*, 150(1-2):1–15, 2003.
179. R. Riveret, A. Rotolo, H. Prakken, and G. Sartor. Heuristics in argumentation: a game-theoretical investigation. In *Proceedings of the 2nd International Conference on Computational Models of Argument*. IOS Press, 2008.
180. J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, 12(1):23–41, 1965.
181. A. Ross. *Directives and norms*. Routledge and Kegan Paul, 1968.
182. W. D. Ross. *The Right and the Good*. Oxford University Press, 1930.
183. W. D. Ross. *Foundations of Ethics*. Oxford University Press, 1939.
184. B. Roth, R. Riveret, A. Rotolo, and G. Governatori. Strategic argumentation: a game theoretical investigation. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 81–90. ACM Press, 2007.

185. S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2002.
186. G. Sartor. *Linguaggio giuridico e linguaggi di programmazione*. Clueb, 1992.
187. G. Sartor. *Intelligenza artificiale e diritto. Un'introduzione*. Giuffré, 1996.
188. G. Sartor. I linguaggi (e i sistemi) informatici e linguaggio giuridico. In *In Proceedings of the conference: Il diritto nella società dell'Informazione. Firenze: Istituto per la documentazione giuridica.*, pages 275–307, 1998.
189. G. Sartor. I linguaggi (e i sistemi) informatici: un vincolo per il giurista. *Rivista del notariato*, 7(5):825–859, 1998.
190. G. Sartor. Teleological arguments and theory-based dialectics. *Artificial Intelligence and Law*, 10:95–112, 2002.
191. G. Sartor. *Legal Reasoning: A Cognitive Approach to the Law*. Springer, 2005.
192. G. Sartor. Fundamental legal concepts: a formal and teleological characterisation. *Artificial Intelligence and Law*, 14(1):101–142, 2006.
193. B. T. R. Savarimuthu and M. Purvis. Mechanisms for norm emergence in multiagent societies. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–3. ACM Press, 2007.
194. M. Schumacher. *Objective coordination in multi-agent system engineering: design and implementation*. Springer-Verlag, 2001.
195. E. Schwalb and L. Vila. Temporal constraints: A survey. *Constraints*, 3(2/3):129–149, 1998.
196. J. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.
197. J. R. Searle. What is a speech act? In *Philosophy in America*. Allen and Unwin, 1965.
198. M. J. Sergot. Prospects for representing the law as logic programs. In *Logic Programming*, pages 33–42. Academic Press, 1982.
199. M. J. Sergot, F. Sadri, R. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.
200. M. Shanahan. The event calculus explained. In *Artificial Intelligence Today: Recent Trends and Developments*, pages 409–430. Springer, 1999.
201. Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the 10th American National Conference on Artificial Intelligence*, pages 276–281. AAAI/MIT Press, 1992.
202. M. P. Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12):40–47, 1998.
203. R. G. Smith. The contract net protocol: high-level communication and control in a distributed problem solver. *Distributed Artificial Intelligence*, pages 357–366, 1988.
204. R. H. Thomason. Desires and defaults: A framework for planning with inferred goals. In *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pages 702–713. Morgan Kaufmann, 2000.
205. J. Vázquez-Salceda, H. Aldewereld, and F. Dignum. Implementing norms in multiagent systems. In *Proceedings of the 2nd German Conference on Multiagent System Technologies*, pages 313–327. Springer, 2004.
206. S. Vida. *Norma e condizione. Uno studio dell'implicazione normativa*. Giuffré, 2001.
207. A. von der Lieth Gardner. *An artificial intelligence approach to legal reasoning*. MIT Press, 1987.
208. G. H. von Wright. *Norm and Action. A logical Inquiry*. Routledge and Kegan Paul, 1963.

209. G. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1-2):225–279, 1997.
210. W3C. www.w3.org, 2007.
211. D. Walton. *Argumentation Methods for Argumentation in Law*. Springer, 2005.
212. I. Watson and F. Marir. Case-based reasoning: A review. *The Knowledge Engineering Review*, 9(4):355–381, 1994.
213. M. Weiser. Some computer science issues in ubiquitous computing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3):12, 1999.
214. M. Whitsey. Logical omniscience: A survey. Unpublished, 2003.
215. M. Wooldridge. Verifying that agents implement a communication language. In *Proceedings of the 16th American National conference on Artificial Intelligence and the 11th Conference on Innovative Applications of Artificial Intelligence*, pages 52–57. AAAI Press, 1999.
216. M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.
217. M. Wooldridge, N. R. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.