LEGIS: A Proposal to Handle Legal Normative Exceptions and Leverage Inference Proofs Readability

Cleyton Mário de Oliveira Rodrigues
University of Pernambuco, Garanhuns–PE, Brazil
Center of Informatics, Federal University of Pernambuco, Recife–PE, Brazil
cleyton.rodrigues@upe.br

Eunice Palmeira
Federal Institute of Alagoas (IFAL), Maceió–AL, Brazil
eunicepalmeira@ifal.edu.br

Fred Freitas
Center of Informatics, Federal University of Pernambuco, Recife–PE, Brazil
fred@cin.ufpe.br

Italo Oliveira
Faculty of Law, Federal University of Pernambuco, Recife–PE, Brazil
italojsoliveira1@gmail.com

Ivan Varzinczak
Centre de Recherche en Informatique de Lens, Université d'Artois, Lens, France
varzinczak@cril.fr

Abstract

Although the representation of normative texts and simulation of legal acts are commonly interdisciplinary themes in the field of Artificial Intelligence and Law (AI & Law), some questions remain open or are yet explored. Among them, we can mention the formalization of the legal body in the face of explicit or implicit exceptions in the juridical reasoning, and the treatment of readability issues, in exposing or justifying decision-making. In this paper, we present the prototype LEGIS and discuss about a proposal to simulate legal action on two fronts. We adopt a non-monotonic semantics for knowledge representation that is appropriate to the singularities of the legal realm, the Preferential Semantics, and propose a transformation to a formal logic argumentation style, the Sequent Calculus, in order to raise the inference proofs to a level of legibility not yet conveniently attained by conventional reasoners.

1 Introduction

The interdisciplinary field of AI & Law has witnessed the construction of conceptual ontologies capable of mapping the complexity of the legal domain, and of simulating legal actions based on normative texts. Despite intense research in recent years, some subareas still require further investigation. Two not fully resolved issues in this universe are the inability to produce a coherent system w.r.t. the judicial reality (for example, able to handle exceptions between written rules), and the technical language used by the formalisms of knowledge representation, which undermines the understanding of those who would use the system in practice.

Figure 1 pictures a peculiar situation where a juridical normative knowledge-based system would be quite applicable. An agent A deliberately kills an agent B; without further information, the situation normally leads to a simple homicide classification with basic prison sentences. Additional circumstances, such as behavior driven either by frivolous[1] or moral reasons, would increase or decrease the calculus of the punishment, respectively. In addition, more exclusive circumstances, such as those related to sex-based hate, may lead to specific homicide extensions (in this case, a Femicide), overriding previous generic inferences. In this perspective, we argue that a system capable of reasoning over the legal corpus covering possible exceptions, as well as being able to respond (in a (controlled) natural language) about crimes, penalties, and conflicts between norms, is unprecedented and necessary. Although in different proportions, such a practical system would be fruitful for different users, ranging from ordinary laymen, passing law students to lawyers and judges.

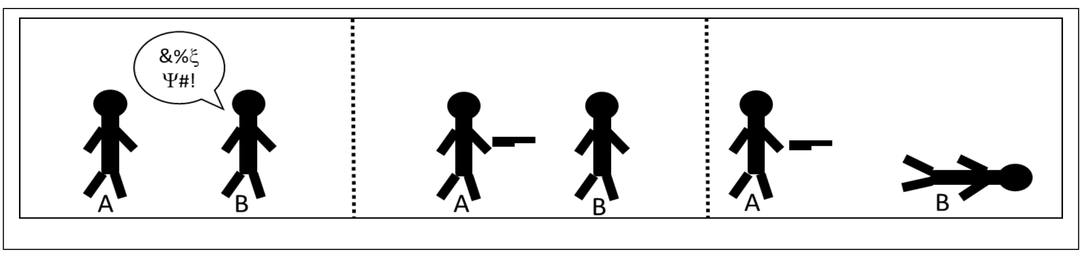

Figure 1: An arbitrary homicide situation

This paper, therefore, proposes the development of a prototype, known as LEGIS (the acronym for LEgal analysIS), addressing the aforementioned issues. We focus on normative legal knowledge, that is, that derived from written legal rules and from legal principles; the ontological basis of LEGIS represents a portion of the Brazilian Penal code. In this context, the exceptions dealt with are those that occur between crimes surrounded by specific circumstances (e.g., an infanticide) in relation to more "normal" crimes (e.g., a typical

[1]In Figure 1, the murder was motivated by a silly discussion among the agents, that is, a shallow reason.

homicide). Besides, the second problem addressed consists of clarifying the users about reasoning and decision-making. For that purpose, LEGIS is soon encompassing an approach that transforms connection-based proofs over the Semantic Web language OWL [1] into sequent calculus' proofs. Such proofs are already quite close to natural language, and an additional translation to text will certainly serve users, e.g., to justify their arguments better while relying on LEGIS.

The paper is organized as follows. Section 2 presents the architecture proposal for LEGIS. Section 3 highlights how exceptions can arise in legal texts. Section 4 shows the syntax, semantics, and reasoning tasks for the Classical and Preferential Description Logic. An axiomatization of the Penal Code based on these logics are discussed in Section 5. In Section 6, we briefly introduce the Connection Calculus proof search, and how to transform connection proofs into more intelligible Sequent Proofs. Finally, conclusions about LEGIS and the ongoing works are discussed in Section 7.

2 LEGIS Proposal

LEGIS is a collaborative effort aimed at reasoning on legal norms. In this perspective, the project unfolds in some dimensions, such as: classical vs. defeasible knowledge bases; monotonic vs. non-monotonic approaches to reasoning; practical implementations with parsimonious use of resources (time, memory, ...); and a justification module for inference proofs. A holistic view of LEGIS extensions is highlighted in Figure 2.

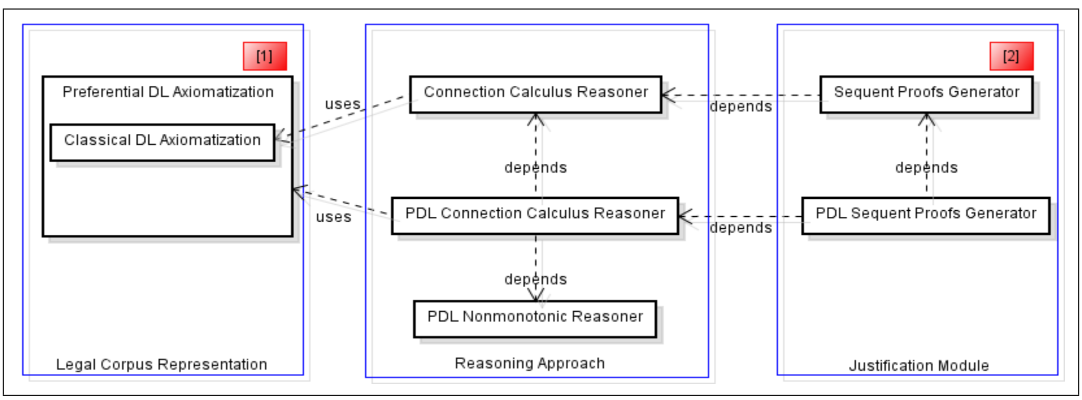

Figure 2: The Holistic View of LEGIS Prototype

Roughly, the idea of LEGIS can be broken down into three levels: one for the representation of knowledge of the legal realm, another for the reasoning strategies, and a third module to provide explanations of the inference proofs closer to natural language. In

Figure 2, the knowledge base should be used by the reasoning module, which provides a customized entry for the transformation process embedded within the justification module.

In particular, in order to attain the previous issues, this work has a two-fold purpose. The first concerns the use of a non-monotonic logical semantics capable of representing and reasoning with exceptions, common in legal texts (labeled in Figure 2 with the identifier [1]). The ontological basis of LEGIS is formed by so-called classical ontologies, i.e. those which reason according to the principles of classical logic, as well as by defeasible ontologies, which allow provisional inferences to be removed as possibly contradictory information is added. These ontologies conceptualize norms of the Brazilian legal system, and the exceptions dealt here involve those that happen between a more general norm and a more specific one. Norms, in turn, are specialized into written rules and general principles. Another goal is the use of a formal logic argumentation in order to make the inference proofs more intelligible to the end user (labeled in Figure 2 with the identifier [2]). This purpose lies in the efficiency vs. readability trade-off regarding the inference engines. So far, LEGIS' reasoning tasks include lawsuit simulation, classification of criminal behavior, and penalty calculus. LEGIS' architecture is illustrated in Figure 3.

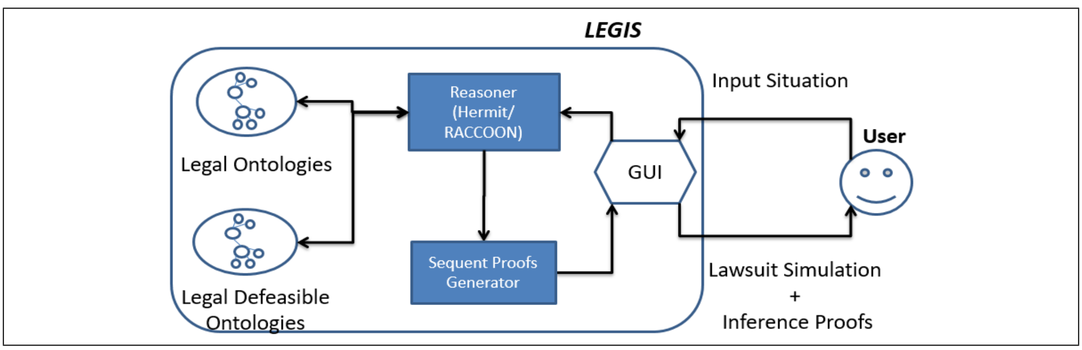

Figure 3: LEGIS Architecture

Through a Graphical User Interface (GUI), an user poses an arbitrary situation. By means of specific reasoners, consistent OWL ontologies will serve as a basis for classifying input instances, even using defeasible axioms. In case of using an inference engine based on the connection calculus (such as RACCOON, an OWL reasoner based on a Description Logic connection calculus, developed under our group [2]), it is possible to transform connection proofs into Sequent proofs (rooted in Sequent Calculus [3]), returning the simulation result and a more readable proof of the inferences made. In the following sections we detail how LEGIS addresses these specific issues.

3 Exceptions in Legal Regulations

Mapping legal normative knowledge into a mathematical formalism free of ambiguities demands time and effort. Some inherent peculiarities of the domain make it especially challenging. Potential sources of anomalies are, for example, the volume of data, the heterogeneity of legal sources, and the legal jargon itself, which uses syntactic inversions, referential ambiguities, and vague terms (open-textured concepts) [4]. In addition, legal systems often present singularities from their political, social and cultural contexts, which makes hard to find a general formalism suited to all of them.

On one hand, it is unfeasible to draw up a normative document capable of anticipating all possible and relevant circumstances. This is why in some cases there are general legal principles that may override the rules, in order to avoid injustices or unwanted conclusion from the literal and direct application of rules. On the other hand, exceptions can be explicitly added throughout the text to accommodate potential specificities of a more general case. In addition to the legal domain peculiarities aforementioned, which may lead to exceptions, Atienza and Manero (2012) [5] argue that the very interplay among laws with legal principles, and apparent conflicts between rules can lead to exceptions and even lack of consensus among lawyers themselves, creating defeasible scenarios of regulation.

In order to illustrate such situation, we exemplify a scenario where an agent's conduct matches the typification of crimes against property, as well as the Trifling principle which removes any criminal liability if the subtracted good is of irrelevant value. The crimes against property correspond to the protected legal interest in the crimes set out in Articles 155-180 of the Brazilian Penal Code[2]. The Trifling principle is entirely related to the globally accepted principle known as *De Minimis Non Curat Lex* [6], in which a behavior with extremely low transgression of the law is not classified as illegal. We transcribe the related legislation below, followed by two didactic examples.

- In Portuguese:
  - **Furto**: *Subtrair, para si ou para outrem, coisa alheia móvel. (Art. 155).*
- In English:
  - **Theft**: *To take a chattel[3], for himself or others.* (Art. 155).

Example 3.1. Will is in a restaurant, and momentarily leaves his wallet on the table to go to the bathroom. John, as he walks past Will's desk, grabs his wallet and leaves.

[2]http://www.planalto.gov.br/ccivil_03/decreto-lei/Del2848compilado.htm
[3]An item of personal property that is movable.

Example 3.2. John is a family man who is unemployed. John often asks for money from passersby near a bakery. Taking advantage of the distraction of an attendant in this bakery, and very hungry, John steals two loaves that were on a nearby counter.

The behavior in Example 3.1 matches a typical theft crime. Regarding the behavior in Example 3.2, at first sight, the conduct falls under article 155 of the Penal Code. Nevertheless when analyzing the patrimonial issue, emerge questions like: Does somebody suffer serious injury? Was the bakery impoverished? How much do two loaves cost? Was the act previously planned? From a material point of view, the action becomes atypical, as it does not apply a very serious legal injury, thus not involving criminal charges. The principle of insignificance overrides the theft imputability. Therefore, it is assumed that the legal texts are defeasible.

Although debates on the use of non-monotonic legal reasoning persist today [7], in the 1980s, Gardner (1987) [8] elicited the minimum requirements for legal reasoning accordingly what happens in legal practice: ability to reason with cases, and to handle open-textured predicates, exceptions, conflicts between rules, besides the ability to handle change and non-monotonicity.

The research developed under the umbrella of AI & Law has relied on full synergy with Description Logic formalism. However, as legal regulations are somehow defeasible, open to implicit exceptions; the inferences made in the legal field are not completely linear, they are usually overruled by new information acquired. Therefore, generalizations are only valid for more typical cases. In the following section, we briefly introduce the syntax and semantics of Description Logic, as well as a DL extension addressing a defeasible subsumption constructor to axiomatize exceptions for typical situations. In addition, we discuss a portion of the axiomatization of the Brazilian Criminal domain.

4 Description Logics

4.1 Classical Description Logic

Description Logics (DLs) [9] are a family of formalisms to knowledge representation and reasoning, able to balance the trade-off between expressiveness and decidability for classical monotonic logic. DLs can be seen as subsets of First-order Logic (FOL), in particular, a well-behaved fragment of L2 FOL (first order predicates with 2 variables). DLs accommodate a range of different flavours, each with its own requirements of decidability and expressiveness. For the sake of clarity, throughout the text, we focus on the sublanguage $\mathcal{ALC}$ (Attributive Language with Complements) which allows axiomatizing an arbitrary domain through conjunction, disjunction, negation, existential and universal restriction constructors.

4.1.1 DL Syntax

A DL language is structured in terms of elementary building blocks, i.e., atomic concepts (A), atomic roles (R), and Individuals (I). Complex concept expressions (C, D) may be constructed on these basic descriptions. In especial, $\mathcal{ALC}$ grammar allows the following concept expressions:

$$C, D ::= A \mid C \sqcap C \mid C \sqcup C \mid \neg C \mid \top \mid \bot \mid \exists R.C \mid \forall R.C$$

An $\mathcal{ALC}$ knowledge base ($\mathcal{KB} := \langle \mathcal{A}, \mathcal{T} \rangle$) is conveniently divided into two disjoint components, one comprising terminological axioms ($\mathcal{T}$), such as concept inclusion and equivalence ($C \sqsubseteq D$ and $C \equiv D$, respectively) and the second with assertional axioms ($\mathcal{A}$), such as concept and role assertion ($C(I)$ and $R(I \times I)$). Hereinafter, we will refer to these components as TBox and ABox, respectively.

4.1.2 DL Semantics

As for its semantics, DL is based on the Open-World Assumption (OWA) [10], since in practice it is inevitably common to handle in the knowledge base with incomplete information. DL semantics is built on top of FOL interpretations, as described in [9]. In short, an Interpretation ($\mathcal{I}$) is a tuple $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ represents the non-empty set known as the domain of $\mathcal{I}$; and $\cdot^{I}$ is a function that maps concepts to subsets of $\Delta^{\mathcal{I}}$, relations to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, from which we can ascribe the following semantics for the $\mathcal{ALC}$ constructors:

- Individual Name (a): $a^{\mathcal{I}}$;
- Atomic Role (R): $R^{\mathcal{I}}$;
- Atomic Concept (A) : $A^{\mathcal{I}}$;
- Intersection ($C \sqcap D$): $C^{\mathcal{I}} \cap D^{\mathcal{I}}$;
- Union ($C \sqcup D$): $C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- Complement ($\neg C$): $\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$;
- Top Concept ($\top$): $\Delta^{\mathcal{I}}$;
- Bottom Concept ($\bot$): $\emptyset$;
- Existential Restriction ($\exists R.C$): $\{a \in \Delta^{\mathcal{I}} \mid \exists b, (a, b) \in R^{\mathcal{I}}, b \in C^{\mathcal{I}}\}$;
- Universal Restriction ($\forall R.C$): $\{a \in \Delta^{\mathcal{I}} \mid \forall b, (a, b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}}\}$;
- Subsumption ($C \sqsubseteq D$): $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;
- Equivalence ($C \equiv D$): $C^{\mathcal{I}} = D^{\mathcal{I}}$;
- Concept Assertion ($C(a)$): $a^{\mathcal{I}} \in C^{\mathcal{I}}$;
- Role Assertion ($R(a, b)$): $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$

4.1.3 DL Reasoning Tasks

From the first-order interpretations, some reasoning tasks [9] are available in DL, such as *Concept Satisfiability* and *Logical Implication*. Given an arbitrary concept C, C is satisfiable iff it admits a model. An interpretation $\mathcal{I}$ is a model of a concept C if $C^{\mathcal{I}} \neq \emptyset$. Likewise, an interpretation $\mathcal{I}$ is a model of a general concept subsumption ($C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Some reasoning tasks can be applied directly to the knowledge base as a whole, in both TBox and ABox sub components. Taking into account the terminological part, we emphasize the following inference tasks:

- *Knowledge Base Satisfiability*: Given a knowledge base $\mathcal{KB}$, and two concepts C and D, $\mathcal{KB}$ is satisfiable if it admits a model, that is, an Interpretation $\mathcal{I}$, which for every axiom $C \sqsubseteq D$ in $\mathcal{KB}$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

- *Concept Satisfiability w.r.t. Knowledge Base ($\mathcal{KB} \not\models C \equiv \bot$)*: Given a knowledge base $\mathcal{KB}$, and a concept C, C is satisfiable w.r.t. $\mathcal{KB}$ if there is an Interpretation $\mathcal{I}$, which is a model for $\mathcal{KB}$, and further a model for C, that is, $C^{\mathcal{I}} \neq \emptyset$.

- *Logical Implication ($\mathcal{KB} \models C \sqsubseteq D$)*: Given a knowledge base $\mathcal{KB}$, and two concepts C and D, D subsumes C, if for all models $\mathcal{I}$ of $\mathcal{KB}$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

For the assertional component, the following reasoning tasks stand out:

- *Concept Instantiation ($\mathcal{KB} \models x : C$)*: Given a knowledge base $\mathcal{KB}$, and an individual x, x is an instance of concept C w.r.t. $\mathcal{KB}$ if $x^{\mathcal{I}} \in C^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{KB}$;

- *Role Name Instantiation ($\mathcal{KB} \models (x, y) : R$)*: Given a knowledge base $\mathcal{KB}$, and some individuals x, y, the pair of individuals (x, y) is an instance of role name R w.r.t. $\mathcal{KB}$ if $\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{KB}$;

From the considerations made so far, we emphasize that an interpretation $\mathcal{I}$ is a model of a $\mathcal{KB} := \langle \mathcal{A}, \mathcal{T} \rangle$ if $\mathcal{I}$ is a model of $\mathcal{T}$ and a model of $\mathcal{A}$.

Example 4.1. To exemplify the classical DL, let us say that something that has a criminal act is a crime. In addition, a theft has an action of subtraction[4], and that any role "has" is associated with a criminal act. $\mathcal{KB}_{crime}$ represents the DL axioms:

$$\mathcal{KB}_{crime} = \left\{ \begin{array}{l} \exists \mathsf{has.CriminalAct} \sqsubseteq \mathsf{Crime} \\ \mathsf{Theft} \sqsubseteq \exists \mathsf{has.Subtraction} \sqcap \forall \mathsf{has.CriminalAct} \end{array} \right\}$$

From $\mathcal{KB}_{crime}$, by Logical Implication inference task, we have:

$$\mathcal{KB}_{crime} \models \mathsf{Theft} \sqsubseteq \mathsf{Crime}$$

[4] In our context, "subtraction" is a convenient synonym for stealing.

4.2 Preferential Description Logic

DL entailment is non-ampliative and non-defeasible, features that are sought for when reasoning with incomplete information and (potential) exceptions. In order to cope with exceptionality, Britz et al. (2011) [11] introduced the Preferential Description Logic (PDL), a DL extension addressing a defeasible subsumption constructor ($\sqsubseteq\!\!\!\sim$). The principal idea is to organize the elements of a domain in degrees of normality, from bottom (the most typical) to up. We note that this kind of knowledge base stratification is fully aligned with the way humans actually reason under incomplete information. In carrying out the reasoning, humans being do not explicitly think of all special cases that would prevent a conclusion from being drawn. Instead, we base our reasoning only on the information at our disposal and provisionally jump to the conclusion. It is only when we come across new information that we accommodate it with the previous knowledge we had and, usually, we do it in a non-disruptive way.

4.2.1 Preferential DL Syntax and Semantics

By extending the DL semantics with non-monotonic reasoning, Britz et al. (2011) [11] have proposed a partial order to set out the levels of typicality. Therefore, the semantics of Preferential DL is organized in terms of strictly partially-ordered structures, $\mathcal{P} := \langle \Delta^{\mathcal{P}}, \cdot^{\mathcal{P}}, \prec^{\mathcal{P}} \rangle$, where: $\langle \Delta^{\mathcal{P}}, \cdot^{\mathcal{P}} \rangle$ is an ordinary DL interpretation; and, $\prec^{\mathcal{P}}$ is a irreflexive, anti-symmetric and transitive partial order on $\Delta^{\mathcal{P}}$. Therefore, given a preferential DL interpretation $\mathcal{P}$ and a defeasible subsumption statement $C \sqsubseteq\!\!\!\sim D$, the semantics of this defeasible axiom is given by:

$$\mathcal{P} \Vdash C \sqsubseteq\!\!\!\sim D \text{ iff } \min_{\prec^{\mathcal{P}}}(C^{\mathcal{P}}) \subseteq D^{\mathcal{P}}$$

The intuition is that objects lower don in $\prec^{\mathcal{P}}$ are more **normal** than those higher up. Thus, $\min_{\prec^{\mathcal{P}}}(C^{\mathcal{P}})$ denotes the most typical elements in $C^{\mathcal{P}}$. In order to explain this preferential semantics, Figure 4 pictures a domain stratified in levels of typicality addressing the criminal domain. We have introduced the concept of EVENT, that is, a category of elements that happens in time, such as an action. EVENT1 maps approximately to Example 3.1, while EVENT2 focuses on the violation addressed in the Example 3.2. In this sense, instead of axiomatizing that events in which an item was subtracted from someone is definitely a theft, it is said that "typically" (that is, in the most normal case), these events are thefts. In such domains, these normal cases is organized in the lower part (EVENT1) of the preferential interpretation of EVENT OF SUBTRACTION domain. In the higher level, EVENT2 is a subtraction of an object (LOAF) whose value is so derisory that the Trifle Principle would be triggered to ward off any indication of crime. Thus, regarding this domain, we have the preferential domain $\mathcal{P}$ defined in terms of $\langle \Delta^{\mathcal{P}}, \cdot^{\mathcal{P}}, \prec^{\mathcal{P}} \rangle$, where:

$$\mathcal{P} : \left\{ \begin{array}{l} \Delta^{\mathcal{P}} = \{event1, event2, loaf, wallet\} \\ \textsc{Theft}^{\mathcal{P}} = \{event1\} \\ \textsc{NonCriminal}^{\mathcal{P}} = \{event2\} \\ \textsc{Object}^{\mathcal{P}} = \{wallet, loaf\} \\ \textsc{WorthlessObject}^{\mathcal{P}} = \{loaf\} \\ \prec^{\mathcal{P}} = \{(event1, event2)\} \end{array} \right\}$$

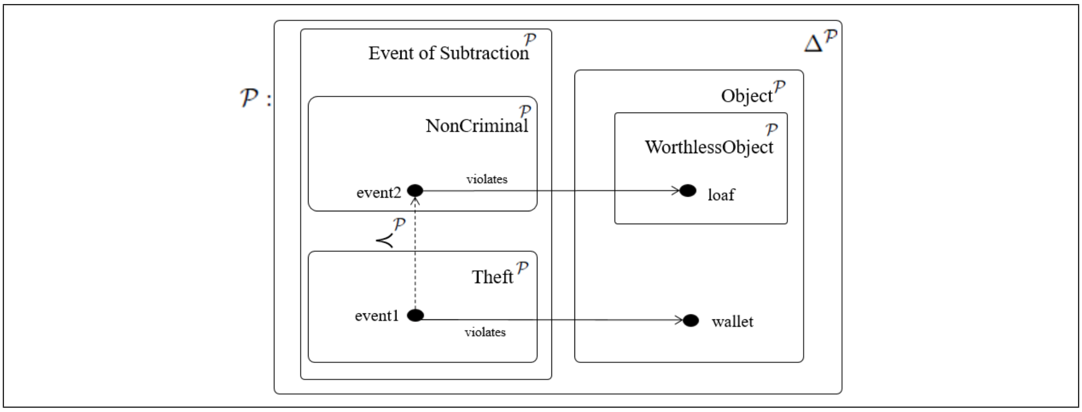

Figure 4: Hierarchy of Item Subtraction Events

Accordingly, we say that **normally**, an item subtraction event is a theft:

$$\textsc{EventofSubtraction} \underset{\sim}{\sqsubset} \textsc{Theft}$$

4.2.2 Preferential and Rational Entailment

To provide reasoning capabilities within a defeasible knowledge base, the newly introduced subsumption constructor also allows for inference tasks, namely the Preferential and Rational entailment tasks. A subsumption relation $C \underset{\sim}{\sqsubset} D$ is preferentially entailed by a given defeasible knowledge base $\mathcal{KB}$ iff $C \underset{\sim}{\sqsubset} D$ is a statement of the preferential closure of $\mathcal{KB}$ [11], i.e., it is a derivation from $\mathcal{KB}$ using the following rules of Preferential Subsumption (derived from the KLM theory [12]):

$$\left\{ \begin{array}{ll} \textsc{Reflexivity} : C \underset{\sim}{\sqsubset} C & \textsc{Left Logical Equivalence} : \frac{C \equiv D, C \underset{\sim}{\sqsubset} E}{D \underset{\sim}{\sqsubset} E} \\ \textsc{And} : \frac{C \underset{\sim}{\sqsubset} D, C \underset{\sim}{\sqsubset} E}{C \underset{\sim}{\sqsubset} D \sqcap E} & \textsc{Or} : \frac{C \underset{\sim}{\sqsubset} E, D \underset{\sim}{\sqsubset} E}{C \sqcup D \underset{\sim}{\sqsubset} E} \\ \textsc{Right Weakening} : \frac{C \underset{\sim}{\sqsubset} D, D \sqsubseteq E}{C \underset{\sim}{\sqsubset} E} & \textsc{Cautious Monotonicity} : \frac{C \underset{\sim}{\sqsubset} D, C \underset{\sim}{\sqsubset} E}{C \sqcap D \underset{\sim}{\sqsubset} E} \end{array} \right\}$$

As usual, inferences in the legal domain should be ampliative beyond retractable. Nevertheless, the preferential entailment does not cover such requirement, since there is no way, from the provided properties, to have $\mathcal{P} \Vdash C \sqcap E \precsim D$ from $\mathcal{P} \Vdash C \precsim D$. In order to accomplish this, Britz et al. (2011) [11] define further the Rational entailment task. Therefore, an additional property, the Rational Monotonicity (RM), should also be ensured by the defeasible subsumption constructor:

$$\textsc{Rational Monotonicity}: \frac{C \precsim D, C \not\precsim \neg E}{C \sqcap E \precsim D}$$

It is worth to mention that we have adopted one of the fundamental principles of rationality in non-monotonic reasoning, namely the principle of *presumption of typicality*, formalized by Lehmann (1995) [13]. Briefly, the principle of presumption of typicality is at the heart of a form of ampliative reasoning and states that we shall always assume that we are dealing with the most typical possible situation compatible with the information at our disposal. Therefore, in the absence of opposite information, RM property infers that individuals are as typical as possible (plausible, though provisional inferences). In this sense, a subsumption relation $C \precsim D$ is rationally entailed by a defeasible knowledge base $\mathcal{KB}$ [11], if $C \precsim D$ is an axiom inferred by the above-mentioned properties including Rational Monotonicity. Suppose, for example, the following TBox:

$$\mathcal{T} = \left\{ \begin{array}{lcl} \mathsf{EventOfSubtraction} & \precsim & \mathsf{Theft} \\ \mathsf{Theft} & \sqsubseteq & \mathsf{Crime} \\ \mathsf{EventOfSubtraction} & \precsim & \neg\exists \mathsf{violates.WorthlessObject} \end{array} \right\}$$

By the Right Weakening property, we have:

$$[1]\frac{\{\ \mathsf{EventOfSubtraction} \precsim \mathsf{Theft},\ \ \mathsf{Theft} \sqsubseteq \mathsf{Crime}\ \}}{\models \mathsf{EventOfSubtraction} \precsim \mathsf{Crime}}$$

In the same way, considering the result of the inference in [1], for any concept expression D, since $\mathsf{EventOfSubtraction} \not\precsim \neg D$, we have by the Rational Monotonicity:

$$[2]\frac{\{\ \mathsf{EventOfSubtraction} \precsim \mathsf{Crime},\ \ \mathsf{EventOfSubtraction} \not\precsim \neg D\ \}}{\models \mathsf{EventOfSubtraction} \sqcap D \precsim \mathsf{Crime}}$$

Obviously, by the same conditions, we can not consider that we will be dealing with the most typical situations possible, considering that D is $\exists\mathsf{violates.WorthlessObject}$. Therefore, an arbitrary reasoner **cannot** infer:

$$\mathcal{T} \models \mathsf{EventOfSubtraction} \sqcap \exists \mathsf{violates.WorthlessObject} \precsim \mathsf{Crime}$$

5 A Proposal to Axiomatize the Legal Domain

In this section, we explore how it is possible to axiomatize the criminal domain, taking into account the exceptions between the norms through the (Preferential) Description Logic. We are not interested in presenting a complete axiomatization of the penal code, but rather how we can extend a legal corpus base with defeasible axioms. The full ontology can be found at `https://github.com/cleytonrodrigues/Tese`.

Throughout the development of the legal conceptual model, we seek to align our domain with some foundational (or upper) ontology, favoring the ontological adequacy, that is, the degree of closeness to reality [14]. As the upper ontology, we chose to stick to UFO (Unified Foundational Ontology [15]) for grounding our concepts with the UFO categories, thus avoiding typical mistakes while building our ontology hierarchy. UFO is a collection of domain-independent ontologies that makes explicit as much as possible the assumptions and rationales w.r.t. the commonsense, through a rich axiomatization of the vocabulary used. In particular, UFO is based on the ontologies of universals, besides providing a profile with constraints that govern how to construct ontologically valid models that are consistent with reality. In the definition of its categories, UFO incorporates, among others, the principle of identity (which provides for the possibility of judging two entities as being the same, i.e, sortals and non-sortals types), besides the principle of rigidity, which investigates whether a type can be instantiated imperiously in all contexts or not (derived from [16]). Table 1 shows a part of this profile.

Stereotype	**Type**	**Constraint**
«kind»	Rigid Sortal	Supertype cannot be a member of «subkind», «phase», «role», «roleMixin».
«subkind»	Rigid Sortal	Supertype cannot be a member of «phase», «role», «roleMixin», and there must be exactly one «kind» as the supertype
«phase»	Anti-Rigid Sortal	Instantiated only in certain contexts, and defined as part of a partition. There must be exactly one «kind» as the supertype.
«role»	Anti-Rigid Sortal	Instantiated only in certain contexts, and dependent on an external relationship. Cardinality on the opposite side of the «role» type should be ≥ 1. There must be exactly one «kind» as the supertype.

Table 1: Modeling Profile of UFO [15]

In particular, UFO addresses the dichotomy between endurant (UFO-A subontology) and perdurant (UFO-B subontology) categories, as shown at the top of Figure 5. For the first component, we have those entities that persist in time (as an agent, an object), and for the second, those which occur in time (i.e., framed by a time interval), as an event. Endurants may be existentially independent (Substantial) or exist only when associated with another entity (Moment). A notoriously complex type of endurant is Situation, a portion of reality recognized as a whole, a state of affairs. In practice, situations are fulfilled by other endurants, including other minor situations. Dependent moment instances may be tied to either a single entity – Intrinsic Moment –, or to an assortment of these: a Relator.

Another reason for choosing this top-level ontology comes from the fact that it provides an ontology of social entities, known as UFO-C [17]. The legal domain is conceived as a description of social reality, where a group of individuals behaves according to a set of State-approved rules that either allow, forbid, or force them to act under some specific circumstances. UFO-C already considers some assumptions of the legal universe. Agents and objects are part of UFO-C subontology. However, unlike an inanimate object, an agent creates actions (Action Contribution).

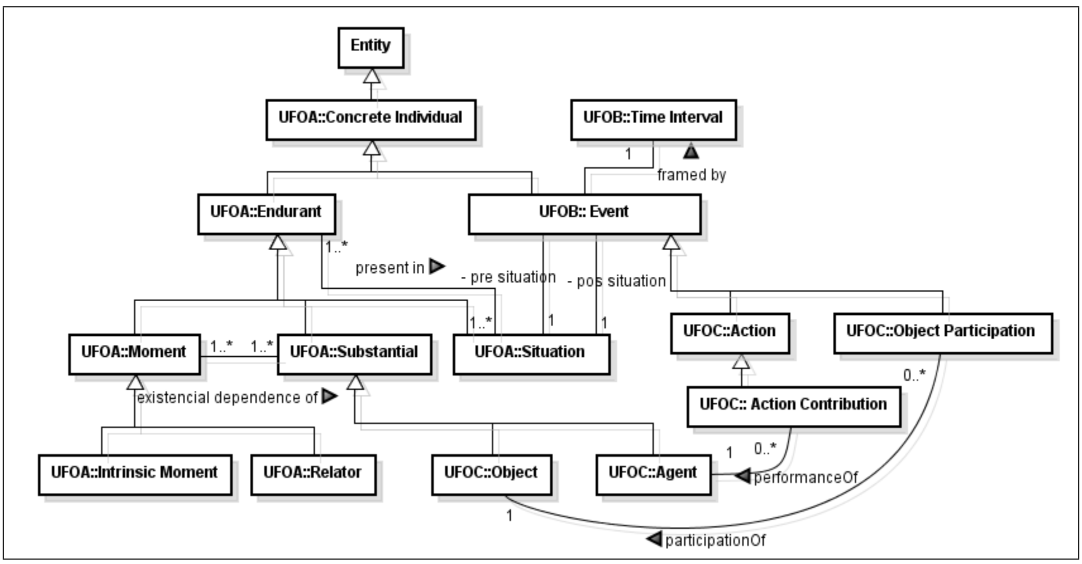

Figure 5: UFO Concepts

Figure 6 illustrates a brief overview of the conceptualization of Crime, regarding the Brazilian Criminal Law. Actually, it is an update of the studies discussed in [18] and [19]. The engineering of this conceptual model was elaborated according to a middle-out approach [20], where intermediate categories of elements are identified first. These are then specialized to match the concepts extracted from legal texts, and generalized towards more generic concepts extracted from the UFO foundational ontology. A crime is a kind of event having as the central kernel an action (we do not consider the crimes of omission, that is, those where an agent had a legal obligation to act, but decides not to accomplish it). CRIMINALACT, therefore, represents these legal actions performed by an OFFENDER, who violates some object of the VICTIM. A LEGALOBJECT can be abstract (honor, life, public peace), or physical (patrimony). An event, in general, starts from an earlier situation towards a result. It is worth mentioning that other important criminal entities, such as space-temporal occurrence, deontic notions of prohibition/permission, norms and punishments are outside the scope of this work; therefore, they are not displayed in the model of Figure 6[5]. Next, we present a DL axiomatization with pure classical axioms. Then, we show an elaborated base enriched with defeasible axioms, highlighting the problems resolved.

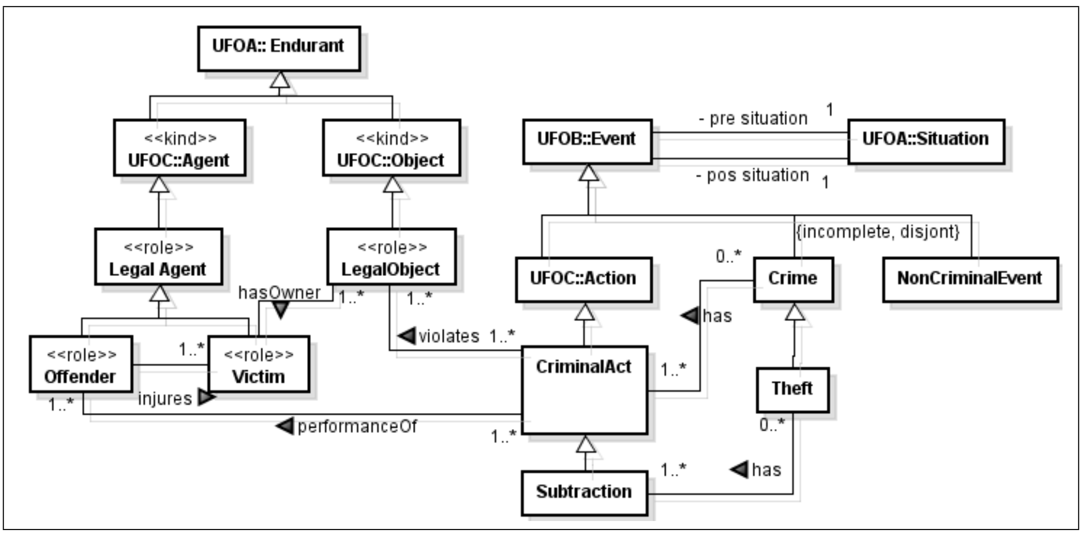

Figure 6: Conceptualization of a Criminal Event

5.1 A Pedagogical Example in DL

In order to make clear the problems arising from the exceptions in the legal texts, we axiomatized a knowledge base related to example 3.2. The base holds the terminological axioms ($\mathcal{T}$) and the assertional ones ($\mathcal{A}$). The terminological axioms map descriptions of a Theft (a subkind of criminal event). In addition, this TBox addresses further a slightly modified set of circumstances that rule out the classification of a crime. For the latter case, we consider the aforementioned Trifle principle.

[5]Additional information can be found at https://github.com/cleytonrodrigues/Tese

$$\mathcal{T} = \left\{ \begin{array}{lll} \mathsf{Crime} & \equiv & \mathsf{Event} \sqcap \exists \mathsf{has.CriminalAct} \\ \mathsf{CriminalAct} & \equiv & \mathsf{Action} \sqcap \exists \mathsf{performanceOf.Offender} \sqcap \exists \mathsf{violates.LegalObject} \\ \mathsf{Offender} & \sqsubseteq & \exists \mathsf{injures.Victim} \\ \mathsf{Subtraction} & \sqsubseteq & \mathsf{CriminalAct} \\ \mathsf{Event} & \sqcap & \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \sqsubseteq \mathsf{Theft}, \\ \mathsf{Event} & \sqcap & \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \\ & & \sqcap \exists \mathsf{violates.WorthlessObject} \sqsubseteq \mathsf{NonCriminalEvent}, \\ \mathsf{Theft} & \sqsubseteq & \mathsf{Crime}, \\ \mathsf{NonCriminalEvent} & \sqsubseteq & \neg \mathsf{Crime} \end{array} \right\}$$

$$\mathcal{A} = \left\{ \begin{array}{l} \mathsf{Event(johnBehavior), Subtraction(loafSubtraction), ChattelObject(loaf),} \\ \mathsf{WorthlessObject(loaf), has(johnBehavior, loafSubtraction),} \\ \mathsf{violates(johnBehavior, loaf).} \end{array} \right\}$$

An Event carried out by means of a Subtraction, violating a ChattelObject classifies the behavior as a theft. New information acquired as the despicable value of the object (WorthlessObject) should refute the previous inference, causing a retraction of the knowledge base. Under the new condition, the event no longer meets the typical theft. It is not possible to keep both inferences, because they are disjoint. It is therefore suggested that the Trifle principle is an exception to the normal case.

Classical DL, therefore, does not address what happens in legal practice. Considering the ABox from the same example, John's behavior would be classified as a theft and a non-criminal event, making the knowledge base inconsistent, i.e., $\mathcal{KB} \models \top \sqsubseteq \bot$, since:

$$[3]\ \frac{\left\{ \begin{array}{l} \mathsf{Event} \sqcap \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \sqsubseteq \mathsf{Theft}, \\ \mathsf{Event(johnBehavior),} \\ \mathsf{Subtraction(loafSubtraction), has(johnBehavior, loafSubtraction),} \\ \mathsf{violates(johnBehavior, loaf), ChattelObject(loaf)} \end{array} \right\}}{\models \mathsf{Theft(johnBehavior)}}$$

$$[4]\ \frac{\left\{ \begin{array}{l} \mathsf{Event} \sqcap \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \sqcap \exists \mathsf{violates.WorthlessObject} \\ \quad \sqsubseteq \mathsf{NonCriminalEvent}, \\ \mathsf{Event(johnBehavior), WorthlessObject(loaf),} \\ \mathsf{Subtraction(loafSubtraction), has(johnBehavior, loafSubtraction),} \\ \mathsf{violates(johnBehavior, loaf), ChattelObject(loaf)} \end{array} \right\}}{\models \mathsf{NonCriminalEvent(johnBehavior)}}$$

$$\mathcal{KB} \models \{\mathsf{Theft(johnBehavior), NonCriminalEvent(johnBehavior), NonCriminalEvent} \sqsubseteq \neg \mathsf{Theft}\}$$

We therefore need a non-monotonic extension of DL Logic capable of dealing satisfactorily with exceptions, as in the interplay between principles and legal laws. Therefore, based on the Preferential DL semantics, the terminological component w.r.t. the john's behavior needs to be slightly modified. In particular, it is necessary to axiomatize that: (1) an event with a subtraction of a chattel object is **typically** a theft, and (2) these events do not **typically** violate a worthless object, and (3) an event with a trifling value object subtraction is **typically** a non-criminal event. The new TBox is shown as follows (the other axioms in $\mathcal{T}$ remain unchanged):

$$\mathcal{T} = \left\{ \begin{array}{l} \mathsf{Event} \sqcap \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \precsim \mathsf{Theft}, \textbf{(1)} \\ \mathsf{Event} \sqcap \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \\ \qquad \precsim \neg \exists \mathsf{violates.WorthlessObject}, \textbf{(2)} \\ \mathsf{Event} \sqcap \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \\ \qquad \sqcap \exists \mathsf{violates.WorthlessObject} \precsim \mathsf{NonCriminalEvent}, \textbf{(3)} \\ \mathsf{Theft} \sqsubseteq \mathsf{Crime}, \\ \mathsf{NonCriminalEvent} \sqsubseteq \neg \mathsf{Crime} \end{array} \right\}$$

Back to Example 3.2, the Rational Monotonicity property rightly prevents John's behavior from being classified as a Theft, but we still have:

$$[5]\ \frac{\left\{ \begin{array}{l} \mathsf{Event} \sqcap \exists \mathsf{has.Subtraction} \sqcap \exists \mathsf{violates.ChattelObject} \sqcap \exists \mathsf{violates.WorthlessObject} \\ \ \precsim \mathsf{NonCriminalEvent}, \\ \mathsf{Event(johnBehavior)}, \mathsf{WorthlessObject(loaf)}, \\ \mathsf{Subtraction(loafSubtraction)}, \mathsf{has(johnBehavior, loafSubtraction)}, \\ \mathsf{violates(johnBehavior, loaf)}, \mathsf{ChattelObject(loaf)} \end{array} \right\}}{\models \mathsf{NonCriminalEvent(johnBehavior)}}$$

In the following section, we discuss the second objective of this study, as highlighted in Figure 2; specifically, the development of the LEGIS module that is capable of producing more readable inference proofs.

6 A Proposal to Sequent Proofs Generator

As previously discussed, it is not enough to develop systems of legal simulation, without guaranteeing an understandable proof verification. Therefore, we discuss an approach based on a formal logic argumentation, in order to provide legible inferences proofs. Nevertheless, the proposal showed here deals only with Classical DL. The extension addressing the Preferential counterpart is discussed in the final remarks.

Freitas and Otten (2016) [1] have proposed a Connection Calculus for the Description Logic $\mathcal{ALC}$ (DL connection method $\mathcal{ALC}$ θ-CM), in the search for a reasoning method that makes a parsimonious usage of memory. In addition, an efficient implementation of this reasoning, known as RACCOON, was developed by Melo Filho et al. (2017) [2]. RACCOON is also highlighted in Figure 3 as an inference engine capable of parsing and reasoning about OWL 2 $\mathcal{ALC}$ ontologies. However, Proof Calculus is far from easy to assimilate. Sequent Calculus [3], a calculus for expressing line-by-line logical arguments, is a more intuitive proof logic. Therefore, we propose in the next subsections a transformation of $\mathcal{ALC}$ Connection Proofs into $\mathcal{ALC}$ Sequent Proofs.

6.1 Non-clausal $\mathcal{ALC}$ θ-Connection Proofs

Our focus is on the non-clausal $\mathcal{ALC}$ θ-Connection Calculus, which is based on the Connection Calculus [21]. Connection calculus is a clear and effective inference method applied successfully over First-Order Logic (FOL). The main idea of connection calculus is checking paths through the FOL formula represented as a matrix, with the purpose of connecting a literal P with its complement $\neg P$. Each pair sets up a connection, which coincides with a tautology in the search branch being examined; therefore, one formula is valid if each path through its matrix representation has a connection. However, before attempting to find a proof, connection calculus converts a formula into a disjunctive normal form (or clausal form), while non-clausal $\mathcal{ALC}$ θ-connection calculus works directly on the structure of the original formula, hence avoiding any translation steps. The later uses $\mathcal{ALC}$ formula with polarity and non-clausal matrices.

An $\mathcal{ALC}$ formula can be expressed as a literal L, or by a disjunction ($C \sqcap D$), or an universal restriction ($\forall R.C$), or a conjunction ($C \sqcup D$), or an existential restriction ($\exists R.C$). C and D are arbitrary concept expressions and L is either an atomic concept or role, possibly negated or instantiated. The *polarity* is denoted by F^p, where F is an $\mathcal{ALC}$ formula and p is the polarity ($p \in \{0, 1\}$). It is used to represent negation in a matrix, i.e. if F and $\neg F$ are $\mathcal{ALC}$ formulae, F has polarity 0 and $\neg F$ has polarity 1 (represented by F^0 and F^1, respectively). The non-clausal matrix is a set of clauses, and each clause is a set of literals and (sub)matrices. The matrix of F^p, denoted by $M(F^p)$, is defined inductively according to Table 2. Therefore, the F matrix is $M(F^0)$. Connection Calculus provides further a graphical representation, in which clauses are organized horizontally, while literals and (sub-)matrices of each clause are arranged vertically. A matrix M can be simplified by replacing matrices and clauses of the form $M = \{\ldots, \{X_1, \ldots, X_n\}, \ldots\}$ within M by $M' = \{\ldots, X_1, \ldots, X_n, \ldots\}$. Restrictions are represented by lines; restrictions with indexes (i.e., the notation $L_{i,j}$) are horizontal lines; restrictions without indexes are vertical lines.

In order to define the non-clausal matrix of an arbitrary $\mathcal{ALC}$ formula, the process starts by the root position ($\models$ or $\sqsubseteq$), which has polarity 0. For example, suppose the example 6.1 drawn from the universe of crime and theft, and the query F_1:

Example 6.1.

$$\left.\begin{array}{l}(\exists \text{has.CriminalAct} \sqsubseteq \text{Crime}) \ \sqcap \\ (\text{Theft} \sqsubseteq \exists \text{has.Subtraction} \sqcap \forall \text{has.CriminalAct})\end{array}\right\} \begin{array}{l}\models \text{Theft} \sqsubseteq \\ \text{Crime}\end{array}$$

Type	F^p	$M(F^p)$	Type	F^p	$M(F^p)$
Atomic	A^0	$\{\{A^0\}\}$	β	$(C \sqcap D)^0$	$\{\{M(C^0), M(D^0)\}\}$
	A^1	$\{\{A^1\}\}$		$(C \sqcup D)^1$	$\{\{M(C^1), M(D^1)\}\}$
α	$(\neg C)^0$	$M(C^1)$		$(C \sqsubseteq D)^1$	$\{\{M(C^0), M(D^1)\}\}$
	$(\neg C)^1$	$M(C^0)$	γ	$(\forall R.D)^1$	$\{\{M(\underline{R^0}), M(\underline{D^1})\}\}$
	$(C \sqcap D)^1$	$\{\{M(C^1)\}, \{M(D^1)\}\}$		$(\exists R.D)^0$	$\{\{M(\underline{R^0}), M(\underline{D^0})\}\}$
	$(C \sqcup D)^0$	$\{\{M(C^0)\}, \{M(D^0)\}\}$	δ	$(\forall R.D)^0$	$\{\{M(\underline{R^1})\}, \{M(\underline{D^0})\}\}$
	$(C \sqsubseteq D)^0$	$\{\{M(C^1)\}, \{M(D^0)\}\}$		$(\exists R.D)^1$	$\{\{M(\underline{R^1})\}, \{M(\underline{D^1})\}\}$
	$(C \models D)^0$	$\{\{M(C^1)\}, \{M(D^0)\}\}$			

Table 2: Matrix of a formula $\mathcal{ALC}$ F^p.

The simplified non-clausal matrix M_1 of F_1 is:

$$
\begin{aligned}
\{\ &\{\underline{\mathsf{has}^0,\ \mathsf{CriminalAct}^0},\ \mathsf{Crime}^1\},\\
&\{\mathsf{Theft}^0, \{\{\underline{\mathsf{has}_1^1}\}, \{\underline{\mathsf{Subtraction}_1^1}\}, \{\underline{\mathsf{has}^0,\ \mathsf{CriminalAct}^1}\}\}\},\\
&\{\mathsf{Theft(johnBehavior)}^1\},\ \{\mathsf{Crime(johnBehavior)}^0\}\ \}
\end{aligned}
$$

Its graphical representation (without polarity notation) is shown in Figure 7. The validation process consists in checking paths through DL formulae, represented as a matrix with the purpose of connecting a literal P with its complement $\neg P$, which are in different clauses. Therefore, a *path* is a disjunction of literals of the form $P_1 \sqcup \ldots \sqcup P_n$.

$$
\left[\left[\begin{matrix}\mathsf{has}\\ \mathsf{CriminalAct}\\ \neg\mathsf{Crime}\end{matrix}\right]\left[\begin{matrix}\mathsf{Theft}\\ \left[[\underline{\neg\mathsf{has}_1}][\underline{\neg\mathsf{Subtraction}_1}]\left[\begin{matrix}\mathsf{has}\\ \neg\mathsf{CriminalAct}\end{matrix}\right]\right]\end{matrix}\right][\neg\mathsf{Theft}(jB)][\mathsf{Crime}(jB)]\right]
$$

Figure 7: Graphical representation of non-clausal matrix for F_1.

Stemming from query F_1 and its graphical matrix representation, the non-clausal $\mathcal{ALC}$ θ-connection proof is depicted in Figure 8. This process is guided by an active path, a subset of a path being investigated through the matrix. It consists of a set of literals that have been connected to reach the current path of proof. In the first step, a clause of the consequent side is selected, $Crime(jB)$, and through an extension step, $Crime(jB)$ is connected to $\neg Crime(jB)$ applying the θ-substitution, which assigns each (possibly omitted) variable an individual or another variable. All remaining paths through the second matrix of the first clause have to be investigated. In order to accomplish this, the second proof extension step connects $CriminalAct$ to $\neg CriminalAct$. The third step connects has to $\neg has$. Likewise, $Theft$ is connected with $\neg Theft(jB)$. Finally, a reduction step connects has to $\neg has$ literal in the active path. This ends the proof showing that every path through the related matrix contains a θ-complementary connection. Therefore, the $\mathcal{ALC}$ query is valid.

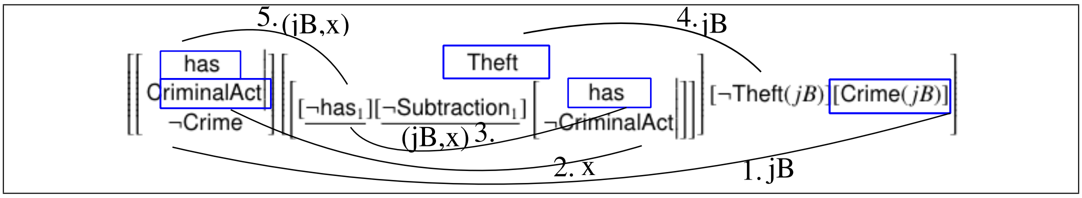

Figure 8: $\mathcal{ALC}$ θ-connection proof using the graphical representation.

6.2 Translating $\mathcal{ALC}$ Connection Proofs into $\mathcal{ALC}$ Sequent Proofs

Given an $\mathcal{ALC}$ formula and its $\mathcal{ALC}$ non-clausal matrix proof, the conversion procedure begins by representing the $\mathcal{ALC}$ formula in its corresponding syntactic tree, where each node can have up to two child nodes. Every node is structured in terms of: (i) a position that identifies each element (predicate or connective) in the formula and is denoted by $a_0, a_1, a_2, \ldots$; (ii) a label consisting of a connective or a logical quantifier (or the predicate itself, if it is atomic); (iii) a polarity (0 or 1), determined by the label and polarity of its parent nodes (root position has polarity 0); and (iv) a type labelled by one Greek letter (α, β, α', β', γ or δ), which is determined by its polarity and its label. Leaf node has no type. Polarity and type of a node are presented in Table 3. The first entry, $(C \sqcap D)^1$ for example, means that a node labeled with $\sqcap$ and polarity 1 has type α and its successor nodes have polarity 1. The syntactic tree for F_1 is shown in Figure 9. The literals names are abbreviated due to space limitations.

Type α			Type β			Type δ		
$(C \sqcap D)^1$	C^1	D^1	$(C \sqcap D)^0$	C^0	D^0	$(\forall R.C)^0$	R^1	C^0
$(C \sqcup D)^0$	C^0	D^0	$(C \sqcup D)^1$	C^1	D^1	$(\exists R.C)^1$	R^1	C^1
$(\neg C)^1$	C^0							
$(\neg C)^0$	C^1							
Type α'			Type β'			Type γ		
$(C \sqsubseteq D)^0$	C^1	D^0	$(C \sqsubseteq D)^1$	C^0	D^1	$(\forall R.C)^1$	R^0	C^1
$(C \models D)^0$	C^1	D^0				$(\exists R.C)^0$	R^0	C^0

Table 3: Polarity and types of nodes for $\mathcal{ALC}$

Back to translation, a position is assigned to each corresponding elements in the non-clausal matrix, as shown in Figure 10. After this, the $\mathcal{ALC}$ non-clausal matrix proof is read, and for each connection found, the tree is examined in order to find leaf nodes that

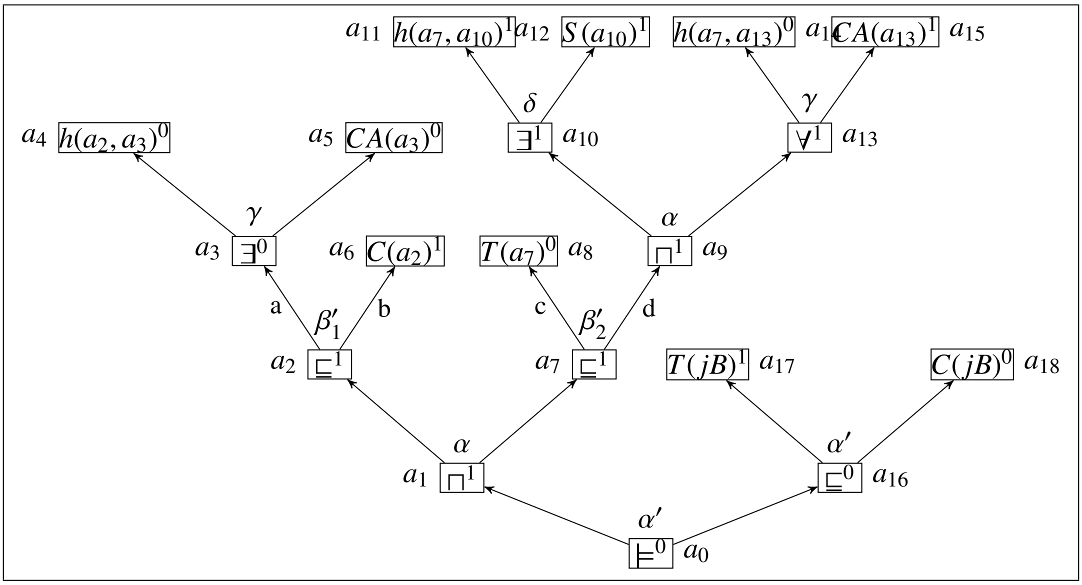

Figure 9: Formula Tree for F_1.

correspond to the connection. The paths between the root node and these nodes in the tree are then analyzed to determine the order of the nodes to be worked on, and thus building a (partial) sequent proof structure. This structure provides information about the ordering in which a given formula F has to be transformed by the rules of the sequent calculus. In addition, it brings out information about branches in the sequent given by positions of type β and β', as shown in Figure 10.

Furthermore, a complete sequent proof (Figure 11) is constructed from the partial sequent proof obtained in the process and by the correspondence between the node and the rules of the sequent, described in Tables 4 and 5 (appendix A). Rules of Sequent Calculus for $\mathcal{ALC}$ [22] is described in appendix B.

The conversion method might be used in practical applications, in areas that employ DL reasoning and generate descriptions on natural language inferences for lay users. Proof conversion can help users understand why a particular situation is characterized as a criminal event, making its use viable in practice, if an additional translation from these sequents to natural language is accomplished. Our research group is already working in this second translation, which will be available soon.

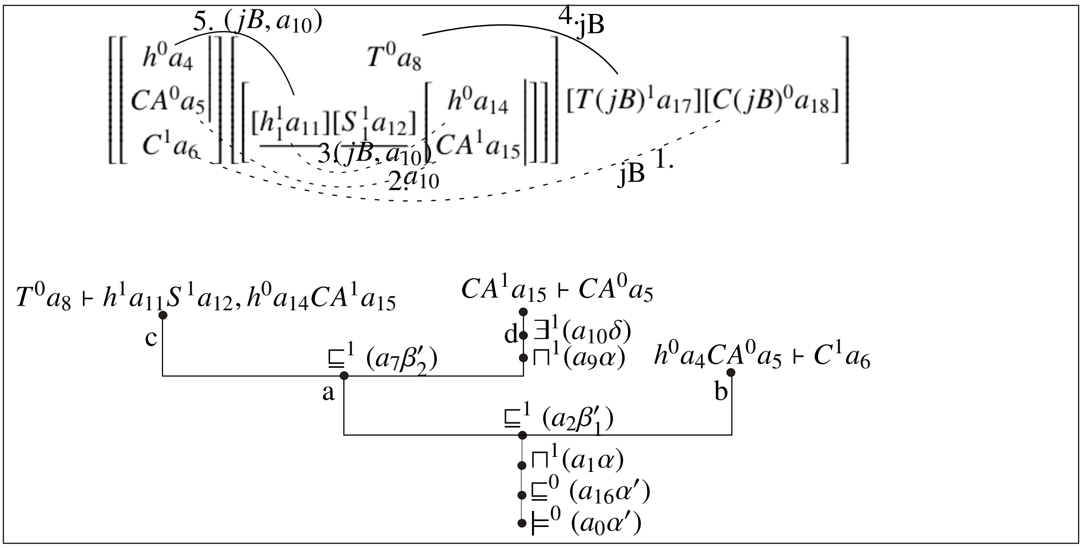

Figure 10: Matrix and structure of the Sequent Proof for F_1.

$$
\dfrac{
  \dfrac{
    T \vdash \exists h.S \sqcap \forall h.CA
    \qquad
    \dfrac{
      \dfrac{
        \dfrac{}{S, CA \vdash CA}\;=
      }{\exists h.S, \forall h.CA \vdash \exists h.CA}\;\mathrm{l}\exists
    }{\exists h.S \sqcap \forall h.CA \vdash \exists h.CA}\;\mathrm{l}\sqcap
  }{T \vdash \exists h.CA}\;\mathrm{cut}
  \qquad
  \exists h.CA \vdash C
}{
  \dfrac{(\exists h.CA \vdash C,\ T \vdash \exists h.S \sqcap \forall h.CA) \vdash (T \vdash C)}{\Big(((\exists h.CA \vdash C) \sqcap (T \vdash \exists h.S \sqcap \forall h.CA)) \vdash (T \vdash C)\Big)}\;\mathrm{l}\sqcap
}\;\mathrm{cut}
$$

Figure 11: Complete representation of the resulting $\mathcal{ALC}$ Sequent Proof for F_1.

7 Final Remarks and Ongoing Works

LEGIS is a legal action simulation proposal that should address an ontological basis of legal norms and principles, some inference mechanisms for efficient reasoning, and a justification module capable of generating intuitive proofs. In the present study, we proposed an axiomatization based on the Preferential Description Logic to address the possible levels of exception between the norms. In addition, the conversion process between connection and sequent proofs highlighted is complete. Since the prototype is the (partial) result of a joint effort, other activities have also been carried out in related studies:

- an ontology for Crimes against Life [19];

- an ontology for Crimes against Property [18]; and,
- an implementation for the Connection calculus for the Description Logic $\mathcal{ALC}$ [2];

With respect to the ongoing works, it is under investigation how to extend the current OWL 2 reasoners to enable non-monotonic inferences, according to Preferential DL perspective. Similarly, a next step will be to engineer a connection calculus implementation to Preferential $\mathcal{ALC}$. In particular, one future work is to investigate how to extend the Protégé reasoner plugin DIP (Defeasible Inference Platform) – a scalable implementation for the preferential semantics [23] – to accomplish such task. Currently, we are also implementing an automatic translation system from the connections proofs to sequent proofs and another to natural language. According, a sequent proof generator for Preferential DL entailment is also expected.

Finally, we intend to make LEGIS available as a web-based front-end system through which it is possible to perform functional and accessible legal simulations by the mapped ontologies. We hope that the results obtained so far can improve the layperson's legal understanding and assist in the labor-intensive task of lawsuits performed by professional lawyers. A prototype is available at `https://github.com/cleytonrodrigues/Tese`. Currently, for arbitrary situations, the prototype is able to infer about the presence of some crime, the violated norms, and the penalties imposed.

Acknowledgement

This research is part of project APQ-0550-1.03/16 (*Reconciling Description logic and non-monotonic reasoning in the legal domain*), supported by Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco, by the Institut National de Recherche en Informatique et Automatique and by the Centre National de la Recherche Scientifique.

References

[1] Freitas, F., Otten, J.: A connection calculus for the description logic $\mathcal{ALC}$. In Khoury, R., Drummond, C., eds.: Advances in Artificial Intelligence, Cham, Springer International Publishing (2016) 243–256

[2] Melo Filho, D., Freitas, F., Otten, J.: Raccoon: A connection reasoner for the description logic alc. In Eiter, T., Sands, D., eds.: LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning. Volume 46 of EPiC Series in Computing., EasyChair (2017) 200–211

[3] Gentzen, G.: Untersuchungen über das logische Schließen II. Mathematische Zeitschrift **39** (1935)

[4] Rissland, E.L., Ashley, K.D., Loui, R.: Ai and law: A fruitful synergy. Artificial Intelligence **150**(1) (2003) 1 – 15

[5] Atienza, M., Manero, J.R.: Rules, principles and defeasibility. In Beltrán, J.F., Ratti, G.B., eds.: The Logic of Legal Requirements - Essays on Defeseability. Oxford University Press, Oxford, United Kingdom (2012) 238–253

[6] Samuels, A.: De minimis non curat lex. Statute Law Review **6**(1) (1985) 167–169

[7] Araszkiewicz, M. In: Legal Rules: Defeasible or Indefeasible? Springer International Publishing, Cham (2015) 415–431

[8] Gardner, A.: An Artificial Intelligence Approach to Legal Reasoning. MIT Press, Cambridge, MA, USA (1987)

[9] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications. 2nd edn. Cambridge University Press, New York, NY, USA (2010)

[10] Reiter, R. In: On Closed World Data Bases. Springer US, Boston, MA (1978) 55–76

[11] Britz, K., Meyer, T., Varzinczak, I.: Semantic foundation for preferential description logics. In Wang, D., Reynolds, M., eds.: AI 2011: Advances in Artificial Intelligence, Berlin, Heidelberg, Springer Berlin Heidelberg (2011) 491–500

[12] Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44**(1-2) (1990) 167–207

[13] Lehmann, D.: Another perspective on default reasoning. Annals of Mathematics and Artificial Intelligence **15**(1) (1995) 61–82

[14] Bringuente, A.C.O., de Almeida Falbo, R., Guizzardi, G.: Using a foundational ontology for reengineering a software process ontology. JIDM **2**(3) (2011) 511–526

[15] Guizzardi, G.: Ontological foundations for structural conceptual models. PhD thesis, University of Twente (2005)

[16] Guarino, N., Welty, C.: An overview of ontoclean. In Staab, S., Studer, R., eds.: Handbook on Ontologies. International Handbooks on Information Systems. Springer Berlin Heidelberg (2004) 151–171

[17] Guizzardi, G., Wagner, G.: A unified foundational ontology and some applications of it in business modeling. In Grundspenkis, J., Kirikova, M., eds.: CAiSE Workshops (3), Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia (2004) 129–143

[18] Rodrigues, C.M.d.O., Azevedo, R.R., Freitas, F.L.G., da Silva, E.P., da Silva Barros, P.V.: An ontological approach for simulating legal action in the brazilian penal code. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing. SAC '15, New York, NY, USA, ACM (2015) 376–381

[19] Rodrigues, C.M.d.O., de Freitas, F.L.G., da Silva Oliveira, I.J.: An ontological approach to the three-phase method of imposing penalties in the brazilian criminal code. In: 2017 Brazilian Conference on Intelligent Systems, Uberlândia, Brazil, October 2-5. (2017) 414–419

[20] Uschold, M., Gruninger, M.: Ontologies: Principles, Methods, Applications. Volume 11. (1996)

[21] Bibel, W.: Automated theorem proving. 2 edn. Künstliche Intelligenz. Vieweg, Braunschweig, Germany (1987)

[22] Borgida, A., Franconi, E., Horrocks, I.: Explaining ALC Subsumption. ECAI - European Conference on Artificial Intelligence (2000)

[23] Casini, G., Meyer, T.A., Moodley, K., Sattler, U., Varzinczak, I.J.: Introducing defeasibility into OWL ontologies. In: The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II. (2015) 409–426

[24] Girard, J.Y., Taylor, P., Lafont, Y.: Proofs and Types. Cambridge University Press, New York, NY, USA (1989)

A Matrices for Translating $\mathcal{ALC}$ Connection Proofs into $\mathcal{ALC}$ Sequent Proofs

Type α	Rule
$\neg^1$	$\mathrm{r}\neg\neg$
$\neg^0$	$\mathrm{l}\neg\neg$
$\sqcap^1$	$\mathrm{r}\neg\sqcap$
$\sqcup^0$	$\mathrm{l}\neg\sqcup$

Type β	Rule
$\sqcap^0$	$\mathrm{l}\neg\sqcap$
$\sqcup^1$	$\mathrm{r}\neg\sqcup$
Type δ	**Rule**
$\forall^0$	$\mathrm{l}\neg\forall$
$\exists^1$	$\mathrm{r}\neg\exists$

Table 4: Correspondence between label, polarity and type of a node, preceded by a node labeled by a negation, with the sequent rules

Type α	Rule
$\sqcap^1$	$\mathrm{l}\sqcap$
$\sqcup^0$	$\mathrm{r}\sqcup$
$\neg^1$	$\varnothing$
$\neg^0$	$\varnothing$
Type α'	**Rule**
$\sqsubseteq^0$	$\varnothing$
$\models^0$	$\varnothing$

Type β	Rule
$\sqcap^0$	$\mathrm{r}\sqcap$
$\sqcup^1$	$\mathrm{l}\sqcup$
Type β'	**Rule**
$\sqsubseteq^1$	$\dfrac{\Gamma \vdash \Delta, A \qquad A, \Sigma \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi}$

Type δ	Rule
$\forall^0$	$\mathrm{r}\forall$
$\exists^1$	$\mathrm{l}\exists$
Type γ	**Rule**
$\forall^1$	$\varnothing$
$\exists^0$	$\varnothing$

Table 5: Correspondence between label, polarity and type of a node, not preceded by negation, with the sequent rules

B A Sequent Calculus for $\mathcal{ALC}$

The calculus consists of three parts, where the first two describe sets of rules, while the latter describes a set of axioms, see figure 12, and the Cut Elimination Theorem is applied according to the proposition 1.

Proposition 1. Cut Elimination Theorem [24]. Let S be a set of sequents (axioms) and s an individual sequent. $S \vdash_{SC} s$, if and only if, there is a proof in SC of s whose leaves are either logical or sequent axioms obtained by the substitution of S-belonging sequents , where the cut rule, $\frac{\Gamma \vdash \Delta, A \quad A, \Sigma \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi}$, is only applied with a premise being an axiom.

Rules for the propositional formulae

$$\frac{X, a, b \vdash Y}{X, a \sqcap b \vdash Y}\ (l\sqcap) \qquad \frac{X \vdash a, Y \quad X \vdash b, Y}{X, \vdash a \sqcap b, Y}\ (r\sqcap)$$

$$\frac{X, \neg a \vdash Y \quad X, \neg b \vdash Y}{X, \neg(a \sqcap b) \vdash Y}\ (l\neg\sqcap) \qquad \frac{X \vdash \neg a, \neg b, Y}{X \vdash \neg(a \sqcap b), Y}\ (r\neg\sqcap)$$

$$\frac{X, a \vdash Y \quad X, b \vdash Y}{X, a \sqcup b \vdash Y}\ (l\sqcup) \qquad \frac{X \vdash a, b, Y}{X \vdash a \sqcup b, Y}\ (r\sqcup)$$

$$\frac{X, \neg a, \neg b \vdash Y}{X, \neg(a \sqcup b) \vdash Y}\ (l\neg\sqcup) \qquad \frac{X \vdash \neg a, Y \quad X \vdash \neg b, Y}{X \vdash \neg(a \sqcup b), Y}\ (r\neg\sqcup)$$

$$\frac{X, a \vdash Y}{X, \neg\neg a \vdash Y}\ (l\neg\neg) \qquad \frac{X \vdash a, Y}{X \vdash \neg\neg a, Y}\ (r\neg\neg)$$

Rules for quantified formulae

$$\frac{X' \vdash b, Y'}{X \vdash \forall r.b, Y}\ (r\forall) \qquad \frac{X', b \vdash Y'}{X, \exists r.b \vdash Y}\ (l\exists)$$

$$\frac{X', \neg b \vdash Y'}{X, \neg\forall r.b \vdash Y}\ (l\neg\forall) \qquad \frac{X' \vdash \neg b, Y'}{X \vdash \neg\exists r.b, Y}\ (r\neg\exists)$$

where $X' = \{a \mid \forall r.a \in X\} \cup \{\neg a \mid \neg\exists r.a \in X\}$, and
$Y' = \{a \mid \exists r.a \in Y\} \cup \{\neg a \mid \neg\forall r.a \in Y\}$

Termination axioms

$$X, a \vdash a, Y \ (=) \qquad X, \neg a \vdash \neg a, Y \ (=)$$

$$X, a, \neg a \vdash Y \ (\mathrm{l}\uparrow) \qquad X \vdash a, \neg a, Y \ (\mathrm{r}\uparrow)$$

$$X, \perp \vdash Y \ (\mathrm{l}\perp) \qquad X \vdash \top, Y \ (\mathrm{l}\top)$$

Figure 12: Rules of Sequent Calculus for $\mathcal{ALC}$ [22].

Received \jreceived