Semantics for Description Logics via Assumption-Based Argumentation: Preliminary Results

Giovanni Buraglio, Federica Di Stefano and Mantas Šimkus

Institute of Logic and Computation, TU Wien

Abstract

Description Logic (DL) terminologies are collections of rules in which a class is defined in terms of a complex DL concept. Various semantics for terminologies have been proposed over the past decades. However, the tight relation with the recent W3C SHACL standard for expressing constraints over RDF graphs motivated the need to enrich terminologies with forms of commonsense reasoning. In this paper, we explore how reasoning over DL terminologies can be captured in argumentative terms, by establishing a translation into Assumption-Based Argumentation (ABA), a rule-based formalism comprising defeasible assumptions, where argumentation semantics are used to retrieve sets of jointly acceptable assumptions. We focus on the problem of graph completion, where a labeled graph is extended by assigning additional labels only to existing nodes while satisfying a given terminology. Building on the existing correspondence between logic programs and ABA, we show how a DL terminology and a labeled graph can be encoded as an ABA framework such that different completions of the graph arise from different criteria for selecting sets of assumptions. We present results for DL terminologies in $\mathcal{ALCI}$ under the classical semantics, the minimal model semantics of Circumscription, and the stable model semantics of Quantified Equilibrium Logic. Furthermore, we provide preliminary results for fragments of the DL-Lite family, where the problem of graph completion is extended to allow the introduction of new domain elements, enabling full model construction.

Keywords

Description Logics, Terminologies, Assumption-Based Argumentation, Graph Completions

1. Introduction

Description Logics (DLs) [1] are a well-established family of formalisms in the realm of Knowledge Representation and Reasoning (KR). They allow for reasoning in highly expressive contexts and enjoy numerous applications in several different scenarios, e.g. in medical ontologies [2]. Over the years, much research has focused on enriching DLs with forms of nonmonotonic reasoning, in the attempt to provide formal methods that allow for common-sense reasoning [3, 4, 5, 6]. In parallel, computational models of argumentation have been developed as an unifying formalism for different forms of nonmonotonic reasoning, comprising logic programming, default logic, and autoepistemic logic, to name a few. Boosted by the seminal paper of Dung [7], various argumentation systems have been introduced, interpreting different forms on nonmonotonic inference as dialectical reasoning [8]. In particular, Assumption-Based Argumentation (ABA) [9] enables argumentative reasoning over rule-based knowledge bases, so-called ABA frameworks (ABAFs), bringing together ideas from abstract argumentation and logic programming. Indeed, while ABAFs syntactically resemble logic programs, consisting of a set of defeasible sentences (called *assumptions*) and inference rules, the reasoning engine of ABA is defined in an argumentative fashion. Although previous works explored connections of argumentation systems with DLs and related formalisms [10, 11, 12, 13], their relationship remains largely unexplored.

In this paper, we continue this line of research, with the aim of providing an argumentative characterization of reasoning in relevant fragments of Description Logics. In particular, we start our investigation by establishing a connection between DLs and Assumption-Based Argumentation in the context of so-called *terminologies* in $\mathcal{ALCI}$. Terminologies are collections of possibly recursive rules, where a class is defined in terms of a complex DL concept that describes the properties of its elements. In

24th International Workshop on Nonmonotonic Reasoning, July 17–19, 2026, Lisbon, Portugal

giovanni.buraglio@tuwien.ac.at (G. Buraglio); fstefano@tuwien.ac.at (F. Di Stefano); mantas.simkus@tuwien.ac.at (M. Šimkus)

the past decades, different semantics based on fix-point computations have been proposed [14, 15]. The need to reconcile the open-world assumption and the closed-world assumption in the context of DL terminologies has recently gained attention, due to the tight relation with the W3C SHACL [16] standard for expressing constraints over RDF graphs. In this context, a nonmonotonic semantics for DL terminologies based on Quantified Equilibrium Logic [17] has been proposed [18]. We focus here on the problem of *graph completion*, which naturally arises in the context of DL terminologies as a classification problem. Specifically, given a labeled graph in input (representing factual information about objects and their relations) together with a terminology that encodes high-level knowledge for the problem, the task is to complete the graph by labeling its nodes in a coherent manner. We consider the problem of graph completions under three different semantics: the classical DL semantics inherited by first-order logic, the minimal model semantics of Circumscription [19, 5] and the stable model semantics of Quantified Equilibrium Logic. We show how the problem of computing graph completions can be carried into ABA, using logic programs as a common link between the two formalisms. For a given graph G and a terminology $\mathcal{T}$, we associate a logic program, obtained by grounding the terminology using G as the universe for the elements. We then exploit the correspondence result between ABA and LPs [20] to associate an ABAF framework to G and $\mathcal{T}$, and retrieve different graph completions based on various argumentation semantics.

Subsequently, we take a step further and extend our investigation by considering general KBs in the DL-Lite family [21]. We shift our attention from the task of graph completions to the task of checking satisfiability of KBs in DL-Lite$_{not}$, an extension of standard DL-Lite$_{Horn}$. By exploiting the good model-theoretic properties of DLs in the DL-Lite family, in particular the small model property [18], we extend the translation given in the context of graph completions, obtaining models by means of extensions of ABA frameworks. Overall, the outcome of our research consists in an argumentative reading of different tasks in the setting of DLs, which allows us to import conflict resolution techniques which are native of argumentation into the realm of Description Logics. On the way to obtaining our results, we contribute to the definition of novel reasoning techniques in assumption-based argumentation by defining native variants of the classical and minimal model semantics.

2. Background

We recall preliminary notions regarding description logics, terminologies and assumption-based argumentation.

Description Logics: Classical, Minimal and Stable Semantics. We recall here $\mathcal{ALCI}$ concept expressions, terminologies and general knowledge bases together with their classical semantics. We assume countably infinite mutually disjoint sets N_C, N_R, N_I of *concept names*, *role names*, and *individuals*, respectively. If $r \in N_R$, then r and the expression r^- are *roles*. We use N_R^+ to denote the set of all roles, i.e. $N_R^+ = \{r, r^- \mid r \in N_R\}$. We simply call *predicate* any element in $N_C \cup N_R$. We define *(complex) concepts* inductively as follows:

(a) the symbols $\top$ and $\bot$ are concepts;
(b) each concept name $A \in N_C$ is a concept;
(c) if C, D are concepts, and r is a role, then $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall r.C$, and $\exists r.C$ are also concepts.

A *concept inclusion* is an expression of the form $C \sqsubseteq D$, where C, D are concepts. A *TBox* $\mathcal{T}$ is any finite set of concept inclusion axioms. *Assertions* are expressions of the forms $A(c)$ or $r(c, d)$, where $A \in N_C$, $r \in N_R$ and $c, d \in N_I$. An *ABox* $\mathcal{A}$ is a finite set of assertions. A *knowledge base* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Given a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$, we denote with $N_I(\mathcal{K}), N_C(\mathcal{K})$ and $N_R(\mathcal{K})$ the set of individual names, concept names and role names occurring in $\mathcal{K}$, respectively. We define the *signature* of $\mathcal{K}$ as the set $sig(\mathcal{K})$ of all predicate symbols (concept and role names) occurring in $\mathcal{K}$. An *interpretation* is a tuple $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set (the *domain*), and $\cdot^\mathcal{I}$ is a function that assigns to every $o \in N_I$ some element $o^\mathcal{I} \in \Delta^\mathcal{I}$, assigns to every $A \in N_C$ some $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and to every $r \in N_R$ some binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The interpretation function $\cdot^\mathcal{I}$ is

$$
\begin{array}{lll}
a^{(\mathcal{I},\mathcal{J})} = a^{\mathcal{I}} & A^{(\mathcal{I},\mathcal{J})} = A^{\mathcal{I}} & r^{(\mathcal{I},\mathcal{J})} = r^{\mathcal{I}} \\
(r^{-})^{(\mathcal{I},\mathcal{J})} = \{(e,e') \mid (e',e) \in r^{\mathcal{I}}\} & & (\neg C)^{(\mathcal{I},\mathcal{J})} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{J}} \\
(C_1 \sqcap C_2)^{(\mathcal{I},\mathcal{J})} = C_1^{\mathcal{I},\mathcal{J}} \cap C_2^{\mathcal{I},\mathcal{J}} & & (C_1 \sqcup C_2)^{(\mathcal{I},\mathcal{J})} = C_1^{\mathcal{I},\mathcal{J}} \cup C_2^{\mathcal{I},\mathcal{J}}
\end{array}
$$

$$
(\exists R.C)^{(\mathcal{I},\mathcal{J})} = \{e \in \Delta^{\mathcal{I}} \mid \exists e' : (e,e') \in R^{(\mathcal{I},\mathcal{J})} \wedge e' \in C^{(\mathcal{I},\mathcal{J})}\}
$$

$$
(\forall R.C)^{(\mathcal{I},\mathcal{J})} = \left\{ e \in \Delta^{\mathcal{I}} \mid \forall e' : \begin{array}{l} (e,e') \in R^{(\mathcal{I},\mathcal{J})} \text{ implies } e' \in C^{(\mathcal{I},\mathcal{J})} \text{ and} \\ (e,e') \in R^{\mathcal{J}} \text{ implies } e' \in C^{\mathcal{J}} \end{array} \right\}
$$

Figure 1: Here-and-There semantics for DLs, with $A \in N_C$, $r \in N_R$, $o \in N_I$ and $R \in N_R^+$.

extended to all concept expressions and all roles in the standard way (see, e.g., [1]). The "$\models$" relation, and thus the notion of a *(classical) model* of a TBox, ABox, or a KB, are as usual.

Definition 1. *Given two interpretations $\mathcal{I}$ and $\mathcal{J}$ and set of predicates F, we write $\mathcal{I} \subseteq_F \mathcal{J}$ if (i) $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and $a^{\mathcal{I}} = a^{\mathcal{J}}$, for all $a \in N_I$, (ii) $p^{\mathcal{I}} \subseteq p^{\mathcal{J}}$, for all $p \notin F$, and (iii) $q^{\mathcal{I}} = q^{\mathcal{J}}$, for all $q \in F$.*

We write $\mathcal{I} \subset_F \mathcal{J}$, if $\mathcal{I} \subseteq_F \mathcal{J}$ and there exists a predicate p such that $p^{\mathcal{I}} \subset p^{\mathcal{J}}$. If $F = \emptyset$, we simply write $\mathcal{I} \subseteq \mathcal{J}$ (resp. $\subset$). We call F the set of *fixed* predicates.

Definition 2 (Minimal models)**.** *Given a KB $\mathcal{K}$, a set of fixed predicates F, a model $\mathcal{I}$ of $\mathcal{K}$ is F-minimal if there exists no $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \subset_F \mathcal{I}$.*

The notion of F-minimal model is a special case of the notion of minimal model in *circumscribed* DLs [5]. We recall the stable model semantics for DLs [18, 6] based on Quantified Equilibrium Logic [17].

Definition 3. *A* Here-and-There (HT) interpretation *is a pair $(\mathcal{I}, \mathcal{J})$ of interpretations with $\mathcal{I} \subseteq \mathcal{J}$. We define an* interpretation function $\cdot^{(\mathcal{I},\mathcal{J})}$ *using the equations in Figure 1.*

Definition 4. *Assume a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ and an HT interpretation $(\mathcal{I}, \mathcal{J})$. We write:*
- $(\mathcal{I}, \mathcal{J}) \models C \sqsubseteq D$, *if* $C^{(\mathcal{I},\mathcal{J})} \subseteq D^{(\mathcal{I},\mathcal{J})}$ *and* $C^{\mathcal{J}} \subseteq D^{\mathcal{J}}$;
- $(\mathcal{I}, \mathcal{J}) \models \mathcal{T}$ *if* $(\mathcal{I}, \mathcal{J}) \models C \sqsubseteq D$ *for all* $C \sqsubseteq D \in \mathcal{T}$;
- $(\mathcal{I}, \mathcal{J}) \models \mathcal{A}$, *if* $\mathcal{I} \models \mathcal{A}$;
- $(\mathcal{I}, \mathcal{J}) \models \mathcal{K}$, *if* $(\mathcal{I}, \mathcal{J}) \models \mathcal{T}$ *and* $(\mathcal{I}, \mathcal{J}) \models \mathcal{A}$.

We can now define *stable models* of a Description Logic Knowledge Base.

Definition 5 (Stable models)**.** *For a set F of fixed predicates, an interpretation $\mathcal{J}$ is a F-*stable *model of a KB $\mathcal{K}$ if (i) $\mathcal{J}$ is a model of $\mathcal{K}$, and (ii) there is no $\mathcal{I}$ s.t. $(\mathcal{I}, \mathcal{J})$ is a model of $\mathcal{K}$ and $\mathcal{I} \subset_F \mathcal{J}$.*

Description Logic Terminologies. We call *terminology* a finite collection $\mathcal{T}$ of possibly recursive concept inclusions of the form $C \sqsubseteq A$, where C is a complex concept defined in the syntax of some DL $\mathcal{L}$ and $A \in N_C$. We remark here that previous works, e.g. [14, 15], define terminologies as a collection of *concept definitions* $C =: A$, where $=:$ is a meta-symbol intuitively standing for (logical) equivalence. In the following, we aim at connecting terminologies in $\mathcal{ALCI}$ with logic programs and hence we shall treat them as standard concept inclusions. Following [18], given a terminology $\mathcal{T}$, its signature, i.e. the set of concept names and role names occurring in $\mathcal{T}$ is partitioned into the set of *defined concepts*, i.e. all concept names A such that $C \sqsubseteq A \in \mathcal{T}$ and the set of *base predicates*, is given by all role names occurring in $\mathcal{T}$ and all concepts names that are not defined concepts. Formally, we denote with $\mathsf{def}(\mathcal{T}) = \{A | C \sqsubseteq A \in \mathcal{T}\}$ and $\mathsf{base}(\mathcal{T}) = (N_C(\mathcal{T}) \cup N_R(\mathcal{T}) \cup \{\top\}) \setminus \mathsf{def}(\mathcal{T})$.

Classical models of terminologies are defined as classical models of general KBs. When applying the minimal model semantics and the stable model semantics to DL terminologies, following [18], we assume that all base predicates are fixed. Base predicates are intuitively the building blocks that we use to define classes in terms of more complex concepts. For given terminology $\mathcal{T}$, an interpretation $\mathcal{I}$ is a *minimal* (resp. *stable*) model of $\mathcal{T}$ if $\mathcal{I}$ is an F-minimal (resp. F-*stable*) model of $\mathcal{T}$ according to the standard Definition 2 and Definition 5, where $F = \mathsf{base}(\mathcal{T})$.

Example 1 ([18]). *Consider the terminology $\mathcal{T}$ in $\mathcal{ALCI}$ defined as*

$$User \sqcap \neg \mathsf{PriviledgedUser} \sqsubseteq \mathsf{BasicUser}$$
$$\exists promotedBy.\mathsf{PriviledgedUser} \sqsubseteq \mathsf{PriviledgedUser}$$

where $\mathsf{def}(\mathcal{T}) = \{\mathsf{PriviledgedUser}, \mathsf{BasicUser}\}$ *and the rest of the occurring predicates belongs to* $\mathsf{base}(\mathcal{T})$*. Assume the ABox* $\mathcal{A} = \{User(a), User(b), promotedBy(a,b), promotedBy(b,a)\}$*. Consider the interpretation* $\mathcal{I}$ *with* $\Delta^{\mathcal{I}} = \{a,b\}$*, with* $a^{\mathcal{I}} = a$ *and* $b^{\mathcal{I}} = b$*,* $promotedBy^{\mathcal{I}} = \{(a,b),(b,a)\}$*,* $User^{\mathcal{I}} = \{a,b\}$*,* $\mathsf{PrivildgedUser}^{\mathcal{I}} = \{a,b\}$ *and* $\mathsf{BasicUser}^{\mathcal{I}} = \emptyset$*. It is easy to see that* $\mathcal{I} \models (\mathcal{A},\mathcal{T})$*. Furthermore,* $\mathcal{I}$ *is a minimal model of* $(\mathcal{A},\mathcal{T})$*. Indeed, the minimization of the predicate* $\mathsf{PriviledgedUser}$ *causes the violation of the first axiom. However* $\mathcal{I}$ *is not a stable model of* $(\mathcal{A},\mathcal{T})$*. Consider the interpretation* $\mathcal{J}$ *such that* $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$*,* $o^{\mathcal{J}} = o^{\mathcal{I}}$ *for all* $o \in \{a,b\}$*,* $promotedBy^{\mathcal{J}} = promotedBy^{\mathcal{I}}$*,* $User^{\mathcal{J}} = User^{\mathcal{J}}$*,* $\mathsf{PriviledgedUser}^{\mathcal{J}} = \emptyset$ *and* $\mathsf{BasicUser}^{\mathcal{J}} = \emptyset$*. Note that* $\mathcal{J} \subset_F \mathcal{I}$*. Furthermore,* $(\mathcal{J},\mathcal{I})$ *is an Here-and-There model of* $(\mathcal{A},\mathcal{T})$*.*

Assumption-Based Argumentation (ABA). Assumption-based argumentation frameworks (ABAFs) consist of a deductive system $(\mathcal{L},\mathcal{R})$, where $\mathcal{L}$ is a set of sentences, and $\mathcal{R}$ is a set of rules over $\mathcal{L}$. A rule $r \in \mathcal{R}$ has the form $\alpha_0 \leftarrow \alpha_1,\ldots,\alpha_n$ with $\alpha_i \in \mathcal{L}$, $body(r) = \{\alpha_1,\ldots,\alpha_n\}$ and $head(r) = \alpha_0$.

Definition 6 (ABA framework). *An ABAF* D *is a tuple* $(\mathcal{L},\mathcal{R},\mathcal{A},\overline{})$*, where* $(\mathcal{L},\mathcal{R})$ *is a deductive system,* $\mathcal{A} \subseteq \mathcal{L}$ *a set of assumptions, and* $\overline{} : \mathcal{A} \to \mathcal{L}$ *is a total mapping, called contrary function.*

For a set of assumptions $S \subseteq \mathcal{A}$, we use $\overline{S}$ to indicate the set of contraries of S. For a set of rules R, we fix $head(R) = \{head(r) \mid r \in R\}$, $body(R) = \{body(r) \mid r \in R\}$. We say that a sentence $\beta \in \mathcal{L}$ is tree-derivable from $S \subseteq \mathcal{A}$ and rules $R \subseteq \mathcal{R}$, denoted by $S \vdash^R \beta$, if there is a finite rooted labelled tree T where: the root of T is labelled with β; the set of labels for the leaves of T is equal to S or $S \cup \{\top\}$; and for every inner node v of T there is a rule $r \in R$ such that v is labelled with $head(r)$, and every successor of v is labelled with $\alpha \in body(r)$ or $\top$ if $body(r) = \emptyset$. We sometimes write $S \vdash \beta$ instead of $S \vdash^R \beta$ if it does not cause confusion.

Definition 7 (Attack). *Let* $D = (\mathcal{L},\mathcal{R},\mathcal{A},\overline{})$ *be an ABAF. A set* $S \subseteq \mathcal{A}$ *attacks* $T \subseteq \mathcal{A}$ *if* $S' \vdash \overline{\alpha}$ *for some* $S' \subseteq S$ *and* $\alpha \in T$*. A set* S *attacks an assumption* α *if* S *attacks the singleton* $\{\alpha\}$*. Moreover, for a set* $B \subseteq \mathcal{A}$ *we say that* S *attacks* B *if* S *attacks some* $b \in B$*. We use* $S^+_R = \{\alpha \in \mathcal{A} \mid S \vdash^R \overline{\alpha}\}$ *for the set of assumptions attacked by* S *and define the* range *of* S *w.r.t.* $R \subseteq \mathcal{R}$ *as* $S^{\oplus}_R = S \cup S^+_R$*.*

A set of assumptions E is conflict-free ($E \in cf(D)$) iff it does not attack itself, i.e. $E^+ \cap E = \emptyset$. Moreover, we say that a set of assumptions S *defends* an assumption α if it attacks each B attacking $\{\alpha\}$. A set of assumptions S is closed in D iff $S \vdash \alpha$ implies $\alpha \in S$. We refer to an ABAF being *flat* if every set S of assumptions is closed, and *non-flat* otherwise. Moreover, we say that an ABAF is *normal* iff each assumption is associated with exactly one contrary and $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ [20]. In the present paper, we consider only flat normal ABAFs.

Several argumentation semantics have been proposed for Assumption-Based Argumentation [9]. Among the existing ones, in this work we will focus on the stable semantics for ABA. For an ABAF D, a set of assumption $E \subseteq \mathcal{A}$ is a stable extension of D ($E \in stb(D)$) iff E attacks each $\{\beta\} \subseteq \mathcal{A} \setminus E$, i.e. $E^{\oplus} = \mathcal{A}$. In general, for a semantics σ, we call $\sigma(D)$ the set of σ-extensions of the ABAF D.

Logic Programs. A *normal logic program* (LP) P is a collection of rules of the form[1]

$$c \leftarrow a_1,\ldots,a_m,\neg b_1,\ldots,\neg b_n$$

For a program P, $Atom(P)$ is the set of all atoms occurring in P. The atoms b_i are called positive atoms and the atoms c_j are called negative atoms. The indices m, n can be equal to zero (that is, rules can have an empty body or only positive or negative atoms in the body). A rule r is called *fact* if $m = n = 0$.

The stable model semantics [22] for logic programs is defined via the so-called *GL-reduct*:

[1]Traditionally, negation in logic programs is denoted with not. Since we aim at considering LPs under different semantics, we use the standard symbol for negation.

Definition 8 (GL-reduct and Stable Models). *Let P be a normal logic program and $S \subseteq Atom(P)$ a set of atoms. The GL-reduct of P wrt S is the negation-free program P^S obtained from P by (i) deleting all rules $r \in P$ with $\neg b_j$ in the body for some $b_j \in S$, and (ii) deleting all negated atoms from the remaining rules. An interpretation S is a* stable model *of P iff S is the minimal model of P^S.*

In Definition 5, we defined stable models of Description Logics KBs, through the semantics of Quantified Equilibrium Logic [17], which generalizes to the first-order level the semantics of (propositional) Equilibrium Logic [23]. An equivalent characterization of stable models for logic programs can be obtained by means of equilibrium models, as in the setting of Description Logics KBs.

3. Classical and Minimal Model Reasoning in ABA

In this section, we lay the foundations for investigating the correspondence between the realm of Description Logics and Assumption-Based Argumentation. Since our study comprises classical, stable and minimal model semantics for DLs, we first need to introduce ABA semantics that capture classical and minimal model reasoning. To do so, it is convenient to look at ABA through the lenses of logic programming. Let us recall the standard translation from logic programs to ABA [20].

Definition 9 ([20]). *For a logic program P, we define the associated ABA framework $D_P = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ such that $\mathcal{L} = \{a | \, a \in Atom(P)\} \cup \{not\_a | a \in Atom(P)\}$, $\mathcal{A} = \{not\_a | a \in Atom(P)\}$, $\mathcal{R} = \{a \leftarrow b_1, \ldots, b_n, not\_c_1, \ldots, not\_c_n | \, a \leftarrow b_1, \ldots, b_n, \neg\, c_1, \ldots, \neg\, c_n \in P\}$ and $\overline{not\_a} = a$ for all $not\_a \in \mathcal{A}$.*

Via the above translation, several semantics for logic programs can be retrieved via different semantics of the ABAFs [20]. We aim at extending the existing results by retrieving other kinds of semantics for logic programs, under the standard translation of Definition 9 and considering other *modes* of selecting sets of assumptions. We start with the classical semantics, simply obtained by interpreting a given logic program P as a conjunction of propositional rules. To define this in the context of ABA, we extend the notion of conflict-free extension as follows.

Definition 10 (Classical extensions). *Given an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, a set of assumptions $E \subseteq \mathcal{A}$ is a* classical *extension of D ($E \in cls(D)$) if and only if, given $S = \mathcal{A} \setminus E$, $E \cup \overline{S} \nvdash \overline{\alpha}$ for any $\alpha \in E$.*

Given the translation of Definition 9, a set of assumptions E in D_P dually represents an interpretation for the atoms in the logic program P by telling us which atoms should be false. Intuitively speaking, the notion of a classical extension E ensures that when the atoms corresponding to assumptions in E are assumed to be false, by considering as true the remaining atoms, we obtain a set that is deductively closed w.r.t. $\vdash$. Moreover, this duality serves as a guide to define the ABA corresponding notion for minimal models of logic programs. Given a program P, an interpretation $M \subseteq Atom(P)$ is a minimal model for P if $M \models P$ and there exists no $M' \subset M$ such that $M' \models P$. Dually, this notion can be captured by considering the *maximal* classical extensions of the corresponding ABAF.[2]

Definition 11 (Maximally classical extension). *For an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, we say that E is a* maximally classical *extension of D iff $E \in cls(D)$ and there exists no $E' \in cls(D)$ such that $E \subset E'$.*

Example 2. *Consider the program $P = \{a \leftarrow b, \neg\, c\}$. Given the associated ABAF D_P (see Def. 9), the sets of assumptions $E_1 = \{not\_a, not\_b\}$, $E_2 = \{not\_a\}$ and $E_3 = \{not\_c\}$ are classical extensions of D_P. Let $M_i = \{a \in Atom(P) | not\_a \notin E_i\}$, for each $i \in \{1, 2, 3\}$. Each M_i is a classical model of P. Moreover, E_1 and E_3 are maximally classical, corresponding to the minimal models M_1 and M_3.*

Classical models of a logic program exactly correspond to classical extensions of the associated ABAF.

Proposition 1. *Consider a program P and let D_P be the associated ABAF. There is a one-to-one correspondence between the classical models of P and the classical extensions of D_P.*

[2] An alternative characterization for minimal models has been defined under a different translation in [9].

For the minimal model semantics, we leverage on the aforementioned duality of assumption-sets and interpretations, and show that maximally classical extensions correspond to minimal models of LPs.

Proposition 2. *Consider a program P and let D_P be the associated ABAF. There is a one-to-one correspondence between the minimal models of P and the maximally classical extensions of D_P.*

The next proposition, as a sanity check, shows that any stable extension is a classical extension, just as stable models of logic programs are classical models.

Proposition 3. *Let D be and ABAF and $E \subseteq \mathcal{A}$, if $E \in stb(D)$ then $E \in cls(D)$.*

4. From DL Terminologies to ABA: the Case of Graph Completions

In this section, we introduce the problem of *graph completions* w.r.t. terminologies in $\mathcal{ALCI}$, where an input finite labeled graph defines the extension of base predicates occurring in a terminology.

Given a terminology $\mathcal{T}$, a model of $\mathcal{T}$ is an assignment of the *defined concepts* in $\mathcal{T}$, for a certain extension of the *base predicates* (e.g. Example 1). For a terminology $\mathcal{T}$, we define $\mathsf{BC}(\mathcal{T}) = \mathsf{base}(\mathcal{T}) \cap N_C$ and $\mathsf{BR}(\mathcal{T}) = \mathsf{base}(\mathcal{T}) \cap N_R$. Observe that since all role names belong to $\mathsf{base}(\mathcal{T})$, then $\mathsf{BR}(\mathcal{T}) = N_R(\mathcal{T})$. If $\mathcal{T}$ is clear from the context, we write BC for $\mathsf{BC}(\mathcal{T})$ and BR for $\mathsf{BR}(\mathcal{T})$.

Definition 12 (Labeled graph)**.** *Given a terminology $\mathcal{T}$, a* labeled graph *for $\mathcal{T}$ is a structure $G = (V, E, \mathcal{L}_{\mathsf{BC}}, \mathcal{L}_{\mathsf{BR}})$ such that V is a finite set of vertices (or* data entries*), $\mathcal{L}_{\mathsf{BC}} \colon V \to 2^{\mathsf{BC}}$ and $\mathcal{L}_{\mathsf{BR}} \colon V \times V \to 2^{\mathsf{BR}}$ and $E = \{(v_i, v_j) \in V \times V \mid \mathcal{L}_{\mathsf{BR}}(v_i, v_j) \neq \emptyset\}$.*

Given a labeled graph G defined as above, we denote with $\mathcal{A}_G$ the ABox such that $B(v_i) \in \mathcal{A}_G$ iff $B \in \mathcal{L}_{\mathsf{BC}}(v_i)$ and $r(v_i, v_j) \in \mathcal{A}$ iff $r \in \mathcal{L}_{\mathsf{BR}}(v_i, v_j)$. Observe that every ABox $\mathcal{A}$ over $\mathsf{base}(\mathcal{T})$ of some terminology $\mathcal{T}$ can be seen as a labeled graph. We now define the notion of completion of a *labeled* graph w.r.t. a terminology $\mathcal{T}$.

Definition 13 (Graph completion)**.** *Assume a terminology $\mathcal{T}$ and a labeled graph $G = (V, E, \mathcal{L}_{\mathsf{BC}}, \mathcal{L}_{\mathsf{BR}})$ for $\mathcal{T}$. A* completion *of G w.r.t. $\mathcal{T}$ is an interpretation $\mathcal{I}$ such that $\Delta^{\mathcal{I}} = V$, $B^{\mathcal{I}} = \{v_i \mid B \in \mathcal{L}_{\mathsf{BC}}(v_i)\}$ for all $B \in \mathsf{BC}$, $r^{\mathcal{I}} = \{(v_i, v_j) \mid r \in \mathcal{L}_{\mathsf{BR}}(v_i, v_j)\}$ for all $r \in \mathsf{BR}$, and $\mathcal{I} \models \mathcal{T}$.*

From the above definition, it immediately follows that an interpretation $\mathcal{I}$ is a completion of G given a terminology $\mathcal{T}$ if and only if $\mathcal{I} \models (A_G, \mathcal{T})$.

Definition 14 (Minimal graph completion)**.** *Given a terminology $\mathcal{T}$ and a labeled graph G for $\mathcal{T}$, a completion $\mathcal{I}$ of G w.r.t. $\mathcal{T}$ is* minimal *if $\mathcal{I}$ is a minimal model of $(A_G, \mathcal{T})$.*

Definition 15 (Stable graph completion)**.** *Given a terminology $\mathcal{T}$ and a labeled graph G for $\mathcal{T}$, a completion $\mathcal{I}$ of G w.r.t. $\mathcal{T}$ is* stable *if $\mathcal{I}$ is a stable model of $(A_G, \mathcal{T})$.*

Grounding Terminologies in $\mathcal{ALCI}$. We first show that a terminology can be syntactically simplified by applying a *normalization procedure.* A terminology $\mathcal{T}$ in $\mathcal{ALCI}$ is in *normal form* if only axioms of the following shapes occur in $\mathcal{T}$:

$$\exists r.B \sqsubseteq A \qquad \forall r.B \sqsubseteq A \qquad B_1 \sqcap \cdots \sqcap B_i \sqcap \neg B_{i+1} \sqcap \cdots \sqcap \neg B_n \sqsubseteq A$$

where B, B_j and A are concept names and r is a role name or an inverse role. Any terminology $\mathcal{T}$ can be transformed into polynomial time into a terminology $\mathcal{T}'$ in normal form, applying standard normalization techniques [1]. We say that two terminologies $\mathcal{T}$ and $\mathcal{T}'$ are *equisatisfiable* under a certain semantics, if the existence of a model of $\mathcal{T}$ implies the existence of a model of $\mathcal{T}'$, and vice versa.

Proposition 4. *A terminology $\mathcal{T}$ can be transformed in polynomial time into a terminology $\mathcal{T}'$ in normal form such that $\mathcal{T}$ and $\mathcal{T}'$ are equisatisfiable under the classical, minimal and stable semantics.*

We now define the grounding of terminologies w.r.t. labeled graphs, which we can assume to be in normal form due to the proposition above. For a given terminology $\mathcal{T}$ and a labeled graph $G = (V, E, \mathcal{L}_{\mathsf{BC}}, \mathcal{L}_{\mathsf{BR}})$, we associate to the pair $(\mathcal{A}_G, \mathcal{T})$ a logic program $\mathcal{K}_G$, where facts are given by all the elements in $\mathcal{A}_G$ and rules are obtained by axioms in $\mathcal{T}$. First observe that $\mathcal{T}$ can be transformed into a first-order theory in the standard way [1], by interpreting concept names and role names as unary and binary predicates, and each of the DL logical connectives as its first-order counterpart. We define the grounding of $\mathcal{T}$ w.r.t. G as follows:[3]

(i) for each $\exists r.B \sqsubseteq A \in \mathcal{T}$, we consider the set of rules

$$\bigvee_{v' \in V} \Big(r(v, v') \land B(v') \Big) \rightarrow A(v) \qquad \text{for all } v \in V$$

(ii) for each $\forall r.B \sqsubseteq A \in \mathcal{T}$, we consider the set of rules

$$\bigwedge_{v' \in V} \Big(\neg r(v, v') \lor B(v') \Big) \rightarrow A(v) \qquad \text{for all } v \in V$$

(iii) for each $B_1 \sqcap \dots \sqcap B_i \sqcap \neg B_{i+1} \sqcap \dots \sqcap \neg B_n \sqsubseteq A$, we consider the set of rules

$$B_1(v) \land \dots \land B_i(v) \land \neg B_{i+1}(v) \land \dots \land \neg B_n(v) \rightarrow A(v) \qquad \text{for all } v \in V$$

Remark 1. *Observe that the theory above can be syntactically transformed into a normal logic program. Indeed, each rule of the form (i) can be transformed into $|V|$ rules of the form $r(v, v') \land B(v') \rightarrow A(v)$, for each $v \in V$. While for each rule r of the form (ii), we can introduce a fresh symbol $aux_{v,v'}^{\forall r.B}$ for each disjunct $\neg r(v, v') \lor B(v')$, and replace r with the following family of rules*

$$\bigwedge_{v' \in V} aux_{v,v'}^{\forall r.B} \rightarrow A(v) \qquad \neg r(v, v') \rightarrow aux_{v,v'}^{\forall r.B} \qquad B(v') \rightarrow aux_{v,v'}^{\forall r.B} \qquad \textit{for each } v, v' \in V$$

We denote with $\mathcal{T}_G$ the obtained theory. We then add to $\mathcal{T}_G$ the ABox $\mathcal{A}_G$ consisting of all facts represented in G, and denote with $\mathcal{K}_G$ the final propositional theory. When evaluating $\mathcal{K}_G$, intuitively true atoms give us instructions on how to construct interpretations of predicates occurring in $\mathcal{T}$. Given an interpretation M over the signature of $\mathcal{K}_G$, we can associate to M an interpretation $\mathcal{I}$ over the signature of $(\mathcal{A}_G, \mathcal{T})$ by defining $\Delta^{\mathcal{I}} = V$, $v^{\mathcal{I}} = v$ for all $v \in V$, for all for each $A \in N_C(\mathcal{T})$, we state $A^{\mathcal{I}} = \{v | A(v) \in M\}$ and, for each $r \in N_R(\mathcal{T})$, we state $r^{\mathcal{I}} = \{(v, v') | r(v, v')\}$.

4.1. Retrieving Graph Completions: Reaching ABA through Logic Programs

First, we show that we can retrieve completions of G w.r.t. $\mathcal{T}$ from models of $\mathcal{K}_G$. We start by classical semantics as it paves the way for the minimal and stable model semantics. Recall that a completion of G must interpret base symbols in $\mathsf{base}(\mathcal{T})$ as given by the labeling functions of G. However, the grounded theory $\mathcal{K}_G$ may contain a lot of spurious atoms, hence its models may not correspond to completions of G. We say that a model M of $\mathcal{K}_G$ *adheres* to $\mathcal{A}_G$ iff for each grounded base predicate $B(v)$ (resp. $r(v, v')$), $B(v) \in M$ iff $B(v) \in \mathcal{A}_G$ (resp. $r(v, v')$).

Example 3. *Consider the terminology $\mathcal{T} = \{\exists r.A \sqsubseteq A\}$ and assume the graph $G = (V, E, \mathcal{L}_{\mathsf{BC}}, \mathcal{L}_{\mathsf{BR}})$ with $V = \{v_1, v_2\}$, $E = \{(v_1, v_2), (v_2, v_1)\}$, $\mathcal{L}^{\mathsf{BR}}(v_i, v_j) = \{r\}$, for all $i \neq j$ and $\mathcal{L}^{\mathsf{BC}}(v_i) = \emptyset$. Applying the grounding defined above, we obtain a the grounded theory $\mathcal{K}_G$ given by the following formulas*

$$\begin{aligned} \mathcal{A}_G = \{ & r(v_1, v_2) & \qquad \mathcal{T}_G = \{ & (r(v_1, v_2) \land A(v_2)) \lor (r(v_1, v_1) \land A(v_1)) \rightarrow A(v_1), \\ & r(v_2, v_1)\} & & (r(v_2, v_1) \land A(v_1)) \lor (r(v_2, v_2) \land A(v_2)) \rightarrow A(v_2)\} \end{aligned}$$

[3] For the sake of conciseness, we omitted in the grounding the cases with inverse roles, allowed in $\mathcal{ALCI}$ terminologies. In such a case, in rules of the form (i) and (ii), each occurrence of $r(v, v')$ is simply replaced with $r(v', v)$.

For space reasons, we do syntactically modify $\mathcal{T}_G$ into a logic program. Consider the interpretation $M_1 = \{r(v_1, v_2), r(v_2, v_1), r(v_1, v_1), r(v_2, v_2)A(v_1), A(v_2)\}$. Observe that $M_1 \models \mathcal{K}_G$, however M_1 does not adhere to G since $r(v_i, v_i) \in M$. Let $\mathcal{I}$ be the interpretation associate to M_1, i.e. $\Delta^{\mathcal{I}} = \{v_1, v_2\}$, $A^{\mathcal{I}} = \{v_1, v_2\}$ and $r^{\mathcal{I}} = \{(v_i, v_j) | \text{for all } i, j = 1, 2\}$. It is easy to verify that $\mathcal{I} \models \mathcal{T}$ indeed for each $v_i \in \Delta^{\mathcal{I}}$, $v_i \in \exists r.A^{\mathcal{I}}$ and $v_i \in A^{\mathcal{I}}$. However $\mathcal{I}$ is not a completion of G. On the other hand, consider the interpretation $M_2 = \{r(v_1, v_2), r(v_2, v_1), A(v_1), A(v_2)\}$. It is easy to verify that $M \models \mathcal{K}_G$ and furthermore M adheres to G. Following the above case, it easy to see that the interpretation $\mathcal{J}$ associate to M is a model of $\mathcal{T}$ that is a completion for G.

The following proposition formally proves the correspondence between classical models of $\mathcal{K}_G$ and classical completions of G w.r.t. $\mathcal{T}$.

Proposition 5. *Assume a labeled graph G and a terminology $\mathcal{T}$ in $\mathcal{ALCI}$. From a graph completion $\mathcal{I}$ of G w.r.t. $\mathcal{T}$ we can construct a model M of $\mathcal{K}_G$ adhering to $\mathcal{A}_G$, and vice versa.*

The above result lifts to a correspondence between minimal and stable completions of G w.r.t. $\mathcal{T}$ and minimal and stable models of $\mathcal{K}_G$.

Example 4. *Consider the terminology $\mathcal{T}$ and the graph G as given in Example 3. The model M_2 of $\mathcal{K}_G$ adhering to $\mathcal{A}_G$ is not a minimal model of $\mathcal{K}_G$. Indeed, the interpretation $M' = \{r(v_1, v_2), r(v_2, v_1)\}$ is a model of $\mathcal{K}_G$ adhering to G and $M' \subset M$. Let $\mathcal{J}$ be the interpretation associated to M_2 (see Example 3). Consider the interpretation $\mathcal{I}'$ associated to M', i.e. $\Delta^{\mathcal{I}'} = \{v_1, v_2\}$, $r^{\mathcal{I}'} = \{(v_1, v_2), (v_2, v_1)\}$ and $A^{\mathcal{I}'} = \emptyset$ is a completion of G w.r.t. to $\mathcal{T}$ such that $\mathcal{I}' \subset \mathcal{J}$. Hence $\mathcal{I}$ is not a minimal completion of G w.r.t. to $\mathcal{T}$. Observe that M' is a minimal model of $\mathcal{K}_G$ and the associated interpretation $\mathcal{I}'$ is a minimal completion of G w.r.t. $\mathcal{T}$. Furthermore, since $\mathcal{K}_G$ and $\mathcal{T}$ are positive, M' is a stable model of $\mathcal{K}_G$ and $\mathcal{I}'$ is a stable completion of G w.r.t. $\mathcal{T}$.*

We formally prove the correspondence under the minimal and stable semantics.

Proposition 6. *Given a labeled graph G and a terminology $\mathcal{T}$ in $\mathcal{ALCI}$, from each minimal (resp. stable) completion of G w.r.t. $\mathcal{T}$, we construct a minimal (resp. stable) model of $\mathcal{K}_G$ adhering to $\mathcal{A}_G$, and vice versa.*

We are now in the position to make use of the results from Section 3 and show how graph completions can be obtained by looking at different semantics in ABA. Given a terminology $\mathcal{T}$ and a labeled graph G, we denote with $D_{(\mathcal{T},G)}$ the ABA framework obtained from the grounding $\mathcal{K}_G$. Recall that an extension of assumptions returns us a set of atoms that we can assume as false. As for models of $\mathcal{K}_G$, to ensure that we can construct graph completions of G w.r.t. $\mathcal{T}$, we select sets of assumptions that adhere to $\mathcal{A}_G$.

Definition 16. *Given a labeled graph G, a terminology $\mathcal{T}$ in $\mathcal{ALCI}$ and the ABAF $D_{(\mathcal{T},G)}$, we denote with $not\mathcal{A}_G = \{not\_a |\ a \in \mathcal{A}_G\}$. We say that a set of assumptions E* adheres *to $\mathcal{A}_G$ if*

$$(\mathcal{A} \setminus E) \cap \{not\_B(v), not\_r(v, v') | B, r \in base(\mathcal{T}),\ v, v' \in V\} = not\mathcal{A}_G$$

Example 5. *Recall the terminology $\mathcal{T}$ and the graph G defined in Example 3. Let $\mathcal{K}_G$ be the constructed grounding of $\mathcal{T}$ w.r.t. G. We obtain the ABAF $D_{(\mathcal{T},G)}$, with $\mathcal{L} = \{A(v), r(v, v') | A(v), r(v, v') \in Atom(\mathcal{K}_G)\} \cup \{not\_A(v), not\_r(v, v') | A(v), r(v, v') \in Atom(\mathcal{K}_G)\}$, $\mathcal{A} = \{not\_A(v), not\_r(v, v') | A(v), r(v, v') \in Atom(\mathcal{K}_G)\}$, the set of rules $\mathcal{R} = \mathcal{K}_G$ and $\overline{not\_p} = p$, with $p \in Atom(\mathcal{K}_G)$. In this case, $not\mathcal{A}_G = \{not\_r(v_1, v_2), not\_r(v_2, v_1)\}$. Consider the extension $E = \{not\_r(v_1, v_1), not\_r(v_2, v_2)\}$. It is easy to observe that E adheres to $\mathcal{A}_G$. Furthermore, $E \in cls(D_{(\mathcal{T},G)})$ since E is conflict-free and $\overline{\mathcal{A} \setminus E} \not\vdash \overline{\alpha}$ for all $\alpha \in E$. Consider the interpretation $M = \{p \in Atom(P) | not\_p \notin E\}$, i.e. $M = \{A(v_1), A(v_2), r(v_1, v_2), r(v_2, v_1)\}$. We already observed in Example 3 that $M \models \mathcal{K}_G$ and M adheres to $\mathcal{A}_G$. Consider now the extension $E' = \{not\_r(v_1, v_1)\}$. It is easy to observe that E' is a classical extension of $D_{(\mathcal{T},G)}$. Indeed, the corresponding interpretation $M' = \{A(v_1), A(v_2), r(v_1, v_2), r(v_2, v_1), r(v_2, v_1)$ is a model of $\mathcal{K}_G$. However, M' does not adhere to $\mathcal{A}_G$, as E' does not adhere to $\mathcal{A}_G$ since $not\_r(v_2, v_2) \notin not\mathcal{A}_G$ and $not\_r(v_2, v_2) \in \mathcal{A} \setminus E'$.*

Theorem 1. *From each classical (resp. minimal, stable) completion of a graph G w.r.t. to a terminology $\mathcal{T}$ in $\mathcal{ALCI}$, we can construct a classical extension (resp. maximally classical, stable) of $D_{(\mathcal{T},G)}$ adhering to $\mathcal{A}_G$, and vice versa.*

In the next section, we generalize the techniques above to compute arbitrary models of theories, going beyond graph completions. In particular, we do not restrict the domain of interpretations to the set of nodes of a given labeled graph. To do so, we weaken the language and consider theories in a relevant fragment in the DL-Lite family [21] that, in contrast to terminologies in $\mathcal{ALCI}$ [18], enjoys the *finite model property* under the classical, minimal and stable model semantics. We exploit the latter property to show that classical (resp. minimal, stable) models of theories correspond to classical (resp. maximally classical, stable) extensions of a *finite* ABAF, and vice versa.

5. From DL Terminologies to ABA: Beyond Graph Completions

We now lift the correspondence given in the previous section to the case of arbitrary models of KBs in DL-Lite$_{not}$. Similarly to the case of graph completions, we first construct a logic program, by grounding an input KB. Then, we demonstrate how to construct the associate ABAF, whose extensions correspond to models of the initial KB, under the classical, minimal and stable model semantics.

Given a collection of concept names N_C and role names N_R, a *basic concept* in DL-Lite is defined inductively by the grammar $B := \bot \mid \top \mid A \mid \exists R$, with $A \in N_C$ and $R \in \{r, r^- | r \in N_R\}$. We consider is DL-Lite$_{not}$, which extends standard DL-Lite$_{Horn}$ [21], by allowing concept inclusions of the form:

$$B_1 \sqcap \cdots \sqcap B_i \sqcap \neg B_i \sqcap \cdots \sqcap \neg B_n \sqsubseteq B$$

where B and each B_i, with $i \in \{1, \ldots, n\}$, are basic DL-Lite concepts. Observe that DL-Lite$_{not}$ KBs syntactically resemble normal logic programs and extend the terminologies by allowing concepts of $\exists R$ on the right-hand side of inclusions.

Reasoning in the DL-Lite family under the stable model semantics has been shown to retain good computational properties if all roles are assumed to be fixed [6]. In particular, it can be proved that under the latter assumption, DL-Lite$_{not}$ enjoys the *small model property*, as the following proposition states. In particular, if all roles are fixed, the property holds under the minimal model semantics as well.

We hint at the construction with the following example in the setting of the minimal model semantics.

Example 6. *Consider the KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ with $\mathcal{A} = \{A(a), A(b)\}$, $\mathcal{T} = \{A \sqcap \exists S \sqsubseteq C, \exists S^- \sqsubseteq B\}$. Consider the interpretation $\mathcal{I}$ with $\Delta^\mathcal{I} = \mathbb{N}$, with $a^\mathcal{I} = 1$, $b^\mathcal{I} = 2$, $A^\mathcal{I} = \{1,2\} = C^\mathcal{I}$, $S^\mathcal{I} = \{(n, 2n+1) | n \in \mathbb{N}\}$ and $B^\mathcal{I} = \{2n+1 | n \in \mathbb{N}\}$. First, observe that $\mathcal{I} \models \mathcal{K}$. Assume the set of fixed predicates $F = \{S\}$. We argue that $\mathcal{I}$ is a minimal model. First, we observe that each occurrence of A cannot be minimized without violating the ABox $\mathcal{A}$. The concepts C and B have minimal extensions as well, since each of their occurrences depends uniquely on the fixed role S, which cannot be minimized. We now* shrink *$\mathcal{I}$ by only select some domain elements. While doing so, we need to ensure that the retained elements see the same type of role connections. To do so, we select (at least) two elements m, m' in $\Delta^\mathcal{I}$, such that $m \in (\exists S^-)^\mathcal{I}$ (a* sink *for S) and $m' \in (\exists S)^\mathcal{I}$ (a* sink *for S^-). Let $m = 3$ and $m' = 2$ and observe that $(1,3) \in S^\mathcal{I}$, thus $3 \in (\exists S^-)^\mathcal{I}$, and $(2,5) \in S^\mathcal{I}$, thus $2 \in (\exists S)^\mathcal{I}$. Let $\mathcal{J}$ be the interpretation such that $\Delta^\mathcal{J} = \{1,2,3\}$, $a^\mathcal{J} = 1$, $b^\mathcal{J} = 2$, $A^\mathcal{J} = \{1,2\} = C^\mathcal{J}$, $B^\mathcal{J} = \{3\}$ and $S^\mathcal{J} = \{(1,3),(2,3)\}$. Observe that $\mathcal{J}$ is an F-minimal model of $\mathcal{K}$ of the desired size.*

The above example underlines a relevant aspect of DL-Lite$_{not}$ under the minimal and stable models semantics with all roles fixed: at each domain element e, the minimality of concept names uniquely depends on the role names involving e. Hence, from an interpretation $\mathcal{I}$, we can construct an interpretation $\mathcal{J}$ where we retain only a small number of elements: the individuals occurring in the ABox, to preserve satisfiability; while to preserve minimality, for each $\exists R$ in $\mathcal{K}$, we keep two elements x, y, with $x \in (\exists R^-)^\mathcal{I}$ and $y \in (\exists R)^\mathcal{I}$, which we use to reconstruct role links.

Proposition 7. *Given a KB $\mathcal{K}$ in DL-Lite$_{not}$, if there exists a model $\mathcal{I}$ of $\mathcal{K}$, then there exists a model $\mathcal{J}$ of $\mathcal{K}$ of size bounded by $|N_I(\mathcal{K})| + 2 \cdot |N_R(\mathcal{K})|$. Furthermore, assuming the set of fixed predicates $F \supseteq N_R(\mathcal{K})$, if $\mathcal{I}$ is F-minimal (resp. F-stable) then $\mathcal{J}$ is F-minimal (resp. F-stable).*

5.1. Grounding DL-Lite$_{not}$ Knowledge Bases

We aim at showing that each model of a DL-Lite$_{not}$ knowledge base $\mathcal{K}$ can be used to define a model of the associated grounded theory $T_{\mathcal{K}}$, and vice versa. First, notice that while Proposition 7 gives us a way to construct a model of $\mathcal{K}$ from a given model of $T_{\mathcal{K}}$, it does not immediately provide an intuition for the converse direction. On the one hand, given a KB $\mathcal{K}$ in DL-Lite$_{not}$, we can construct a grounded theory $T_{\mathcal{K}}$ using the finite sets of elements $N_I(\mathcal{K}) \cup P$, with $P = \{p_i | i \in I\}$ and I is an index set such that $|I| = 2 \cdot |N_R(\mathcal{K})|$. Subsequently, each atom in $T_{\mathcal{K}}$ is of the form $B(x)$ or $r(x, x')$, with $x, x' \in N_I(\mathcal{K}) \cup P$. On the other hand, from a model $\mathcal{I}$ of $\mathcal{K}$, we do not have a direct method for constructing a model of $T_{\mathcal{K}}$. Thanks to Proposition 7, we can safely assume the size of $\mathcal{I}$ to be bounded by $|N_I(\mathcal{K})| + 2 \cdot |N_R(\mathcal{K})|$. Furthermore, we can obtain out of $\mathcal{I}$ a collection of grounded predicates of the form $B(x)$ and $r(x, x')$, by injectively mapping elements of $\Delta^{\mathcal{I}}$ into $N_I(\mathcal{K}) \cup P$. However, these may not give us a *complete* interpretation of all atoms in T_K, as the domain of the interpretation $\mathcal{I}$ may be strictly smaller than the number of elements in $N_{\mathcal{I}}(\mathcal{K}) \cup P$. To overcome this issue, we strengthen the result of Proposition 7, showing that any satisfiable KB in DL-Lite$_{not}$ has a model of size exactly $|N_I(\mathcal{K})| + 2 \cdot |N_R(\mathcal{K})|$. Such an extension preserves the minimality and stability of $\mathcal{I}$ as well.

Proposition 8. *Given a KB $\mathcal{K}$ in DL-Lite$_{not}$, if there exists a model $\mathcal{I}$ of $\mathcal{K}$, then there exists a model $\mathcal{J}$ of $\mathcal{K}$ such that $|\Delta^{\mathcal{J}}| = |N_I(\mathcal{K})| + 2 \cdot |N_R(\mathcal{K})|$. Furthermore, assuming the set of fixed predicates $F = N_R(\mathcal{K})$, if $\mathcal{I}$ is F-minimal (resp. F-stable) then $\mathcal{J}$ is F-minimal (resp. F-stable).*

To show the above proposition, we exploit again the fact that in DL-Lite$_{not}$ with fixed roles, at each domain element, the minimality at node only depends on the set of roles in which the element is involved. Given the result of Proposition 7, we can extend the domain of an interpretation $\mathcal{I}$ until we reach the desired bound by cloning existing domain elements, including their role connections. We hint at this construction with the following example.

Example 7. *Recall the KB $\mathcal{K}$ given in Example 6 and let $\mathcal{J}$ be the constructed minimal model of $\mathcal{K}$. We augment the interpretation domain $\Delta^{\mathcal{J}}$ with $\Delta = \{x\}$, and we clone in x the element $3 \in \Delta^{\mathcal{J}}$. Let $\mathcal{J}'$ be the interpretation such that $\Delta^{\mathcal{J}'} = \Delta^{\mathcal{J}} \cup \Delta$ and the interpretation function such that $A^{\mathcal{J}'} = A^{\mathcal{J}}$, $C^{\mathcal{J}'} = C^{\mathcal{J}}$, $B^{\mathcal{J}'} = \{2, x\}$ and $S^{\mathcal{J}'} = S^{\mathcal{J}} \cup \{(1, x), (2, x)\}$. The interpretation $\mathcal{J}'$ is a F-minimal model of $\mathcal{K}$ of the desired size.*

Assume a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ in DL-Lite$_{not}$ and let $N_{\mathcal{I}}(\mathcal{K}) \cup P$ defined as above. We define the grounding of $\mathcal{K}$ w.r.t. $N_{\mathcal{I}}(\mathcal{K}) \cup P$ as the theory given by the set of facts given by the ABox $\mathcal{A}$ and the set of implications $B_1(x) \wedge \cdots \wedge B_i(x) \wedge \neg B_{i+1}(x) \wedge \cdots \wedge \neg B_{n-1}(x) \rightarrow B_n(x)$, with $x \in N_I(\mathcal{K}) \cup P$, where for each $j \in \{1, \ldots n\}$:

(a) $B_j(x) = A(x)$, if $B_j = A \in N_C$,
(b) $B_j(x) = \bigvee_{x' \in N_I(\mathcal{K}) \cup P} r(x, x')$, if $B_j = \exists r$, and $B_j(x) = \bigvee_{x' \in N_I(\mathcal{K}) \cup P} r(x', x)$ if $B_j = \exists r^-$.

The construction above is similar to the grounding we defined in the context of graph completions. However, in this case, we ground with respect to the set $N_I(\mathcal{K}) \cup P$, instead of considering the set of nodes of an input graph. The obtained theory can be transformed into a logic program by renaming the disjunctions appearing on the left-hand side (see Remark 1). We call $T_{\mathcal{K}}$ the resulting theory. The following proposition shows that a correspondence between models of $\mathcal{K}$ and models of $T_{\mathcal{K}}$.

Proposition 9. *Assume a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ in DL-Lite$_{not}$, from a model $\mathcal{I}$ of $\mathcal{K}$, we can construct a model M of $T_{\mathcal{K}}$, and vice versa.*

We now extend Proposition 9 to the case of the minimal and stable model semantics. Observe that, for the latter semantics, Proposition 8 relies on the assumption that all roles occurring in the KB are fixed. We shortly introduce the notion of fixed atoms in logic programs, which is along the same line of fixed predicates in DLs under the minimal and stable model semantics. Given a logic program P and a set $F \subseteq Atom(P)$, given two interpretation M and M', we write $M \subseteq_F M'$ if $M \subseteq M'$ and $M \cap F = M' \cap F$. While, we write $M \subset_F M'$ if $M \subset M$ and $M \cap F = M' \cap F$.

Definition 17. *Assume a logic program P and let $F \subseteq Atom(P)$ be a set of fixed atoms. An interpretation M is an F-minimal model of P if $M \models P$ and there exists no M' such that $M' \models P$ and $M' \subset_F M$.*

Similarly, we can extend the notion of stable models of logic programs, assuming that some atoms are fixed. Given a program P, fixed atoms behave *classically.*[4] Thus, when considering logic program with fixed atoms, intuitively speaking, we assume an extension of fixed atoms, and check whether we can derive (in a GL-reduct sense) the remaining atoms.

Definition 18. *Given a logic program P with fixed atoms in F, an interpretation M is a F-stable model if, given the GL-reduct P^M, $M \models P^M$ and there exists no $M' \subset_F M$ such that $M' \models P^M$.*

We are now ready to extend the result of Proposition 9 under the minimal model semantics and the stable model semantics.

Proposition 10. *Assume a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ in DL-Lite$_{not}$ and the set of fixed predicates $F \supseteq N_R(\mathcal{K})$. From an F-minimal (resp. stable) model of $\mathcal{K}$ we can construct F'-minimal (resp. stable) models of $T_\mathcal{K}$ where $F' = \{B(v), r(v, v')|B, r \in F\}$, and vice versa.*

Intuitively, when looking at a minimal or stable model of $T_\mathcal{K}$, we guess a truth assignment of fixed atoms, always containing grounded roles. The collection of all true atoms of the forms $B(v)$ and $r(v, v')$, with $B, r \in F$, can be seen as a labeled graph, on which the minimal and stable model semantics indicate how to assign the rest of the predicates.

Example 8. *Consider the KB $\mathcal{K}$ given in Example 6. By applying the grounding, we obtain the following theory $T_\mathcal{K}$ given by the set of facts $A(a)$ and $A(b)$ and:*

$$A(x) \wedge (S(x, a) \vee S(x, b) \vee S(x, p_1) \vee S(x, p_2)) \rightarrow C(x) \qquad x \in \{a, b, p_1, p_2\}$$
$$S(a, x) \vee S(b, x) \vee S(p_1, x) \vee S(p_2, x) \rightarrow B(x) \qquad x \in \{a, b, p_1, p_2\}$$

For space reasons, we do not syntactically modify it into a logic program. The interpretation $M = \{A(a), A(b), S(a, p_1), S(a, p_2), S(b, p_1), S(b, p_2), B(p_1), B(p_2), C(a), C(b)\}$ is an F'-stable model of $T_\mathcal{K}$, with $F' = \{S(x, y)|x, y \in \{a, b, p_1, p_2\}\}$. It is easy to check that, up to renaming domain elements, the interpretation associated to M corresponds to the interpretation $\mathcal{J}'$ given in Example 7.

5.2. Retrieving Models in DL-Lite$_{not}$: Reaching ABA through Logic Programs

We now extend the correspondence result to Assumption-Based Argumentation. We start by lifting the correspondence between logic programs and ABA, in the case of F-minimal and F-stable models. To do so, we first need to encode the meaning of fixed atoms in ABA. For this, we introduce the notion of *fixed assumptions.* Accordingly, our notion of classical maximality needs to be adjusted.

Definition 19 (Max. Classical with Fixed Assumptions)**.** *Given an ABAF D, and a set Φ of fixed assumptions, an extension E is Φ-maximally classical if $E \in cls(D)$ and there exists no $E' \in cls(D)$ such that $E' \cap \Phi = E \cap \Phi$ and $E \subset E'$.*

Assume a logic program P and a set $F \subseteq Atom(P)$ of fixed atoms. Given the associated ABA framework D_P, the set F induces the set of fixed assumptions $\Phi = \{not\_b|b \in F\}$. We show that F-minimal models of P are in correspondence with Φ-maximally classical extensions of D_P. First, we hint at this correspondence in the following example.

[4] Traditionally in logic programs, the classicality of an atom is encoded by adding the disjunction $p \vee \neg p$ to the program.

Example 9. *Assume a logic program* $P = \{a \leftarrow b, \neg c\}$ *with* $b, c \in F$*. The* F*-minimal models of* P *are:* $M_1 = \{a, b\}, M_2 = \{b, c\}, M_3 = \{c\}$ *and* $M_4 = \emptyset$*. Consider the corresponding ABAF* $D_P = \{\mathcal{L}_P, \mathcal{R}_P, \mathcal{A}_P, \overline{}\}$ *where* $\mathcal{A}_P = \{not\_a, not\_b, not\_c\}$, $\mathcal{R}_P = \{a \leftarrow b, not\_c\}$*, and* $\overline{not\_x} = x$ *for* $x \in \{a, b, c\}$*. Let* E_i *such that* $E_i = \{not\_x | x \notin M_i\}$*, i.e.,* $E_1 = \{not\_c\}$*,* $E_2 = \{not\_a\}$*,* $E_3 = \{not\_a, not\_b\}m$ *and* $E_4 = \{not\_a, not\_b, not\_c\}$*. First, observe that* $E_i \in cls(D_P)$*. Under the standard notion of maximally classical extensions,* E_4 *dominates* E_1*,* E_2*, and* E_3*. However, all* E_i *are all* Φ*-maximal, where* $\Phi = \{not\_b, not\_c\}$*. We show that this is the case for* E_1*. First, we look at the super-sets of* E_1*, in this case, only* E_4*. Then, we check if they agree on the fixed assumptions. Since* $E_1 \cap \Phi = \{not\_c\}$ *and* $E_4 \cap \Phi = \{not\_b, not\_c\}$*, we derive that* E_1 *is maximal. With a similar argument, it is easy to observe that given* i, j*, with* $i \neq j$ *and* $E_i \subseteq E_j$*, we have that* $E_i \cap \Phi \neq E_j \cap \Phi$*.*

Proposition 11. *Assume a logic program* P *with fixed atoms in* F*, given the corresponding ABAF* D_P *with fixed assumptions* $\Phi = \{not\_b | b \in F\}$*, from an* F*-minimal models of* P *we can construct a* Φ*-maximal classical extensions of* D_P*, and vice versa.*

Next, we define a notion of stable extension for an ABAF D with respect to a set of fixed assumptions Φ. Roughly speaking, we aim at capturing the idea that the stability of an extension should not regard fixed assumptions, which ones assumed act similarly to standard facts in the program. With the following definition, we introduce Φ-stable extensions in an ABA in which we generalized the notion of stable extension assuming a set of fixed assumptions Φ.

Definition 20 (Stable Extensions with Fixed Assumptions)**.** *Given an ABAF* D*, and a set* Φ *of fixed assumptions, an extension* E *is* Φ*-stable if it is conflict-free and, given* $\Phi_E = \{not\_b \in \Phi | not\_b \notin E\}$*, (i)* $E \cup \overline{\Phi_E} \nvdash \overline{\alpha}$*, for all* $\alpha \in E$*, and (ii)* $E \cup \overline{\Phi_E}$ *attacks all the assumptions in* $\mathcal{A} \setminus (E \cup \Phi)$*.*

Intuitively speaking, the set $\overline{\Phi_E}$ individuates a set of atoms in $\mathcal{L}$ which, if introduced as facts in the theory $\mathcal{R}$, due to (i) do not violate the conflict-freeness of E, and due to (ii) contribute to attack non-fixed assumption outside E. Observe that the notion of Φ-stability corresponds to the standard notion of stability if $\Phi = \emptyset$. Next, as in the case of maximally classical extensions, we show how F-stable models of a program P correspond to Φ-stable extensions of the corresponding ABAF D_P.

Example 10. *Consider again the logic program* $P = \{a \leftarrow b, \neg c\}$ *from Example 9, together with one fixed atom* $b \in F$*. The sets* $M_1 = \{a, b\}$ *and* $M_2 = \emptyset$ *are stable extensions of* P*. Let us consider the corresponding extensions in* D_P*,* $E_1 = \{not\_c\}$ *and* $E_2 = \{not\_a, not\_b, not\_c\}$*. For all* i*,* S_i *is a* Φ*-stable extension. We show the claim for* E_1*. First, observe that each* E_i *is conflict-free. Since* $\Phi_{E_1} = \{not\_b\}$*, we obtain that* $E_1 \cup \overline{\Phi_{E_1}} \nvdash c$ *and* $E_1 \cup \overline{\Phi_{E_1}} \vdash a$*, i.e.* E_1 *is* Φ*-stable.*

Proposition 12. *Let* P *be a logic program with fixed atoms in* F*, and* D_P *the corresponding ABAF with fixed assumptions in* $\Phi = \{not\_b | b \in F\}$*. From an* F*-stable model of* P *we can construct a* Φ*-stable extension of* D_P*, and vice versa.*

We show that if all assumptions are fixed, then the sets of classical, maximally classical and stable extensions coincide. This mirrors exactly the case of logic programs with only fixed atoms, where classical, minimal and stable model semantics coincide.

Proposition 13. *Given an ABAF* D*, the classical, max. classical and stable extensions coincide if* $\Phi = \mathcal{A}$*.*

Finally, we prove the main correspondence between ABA extensions and models in DL-Lite$_{not}$. Observe that the grounding of a KB $\mathcal{K}$ in DL-Lite$_{not}$ may result in a disjunctive program $T_{\mathcal{K}}$, as concepts of the form $\exists R$ may occur on the right-hand side of inclusions. The disjunctions produced from grounding these concepts are constituted by atoms of the form $r(x, x')$ (see (b)). We can rewrite any disjunctive rule r as $body(r) \wedge \bigwedge_{x \in N_I(\mathcal{K}) \cup P} \neg r(x, x') \rightarrow \bot$. This simple rewriting preserves the minimal and stable model semantics since each $r(x, x')$ is always fixed. The latter intuitively means that these atoms behave classically, and so does their negation. To simulate constraints, we can introduce an auxiliary atom bot to replace each occurrence of $\bot$. Let $T^{bot}_{\mathcal{K}}$ be the resulting (normal) logic program.

We call *consistent* any classical (resp. F-minimal, F-stable) model M of $T_{\mathcal{K}}^{bot}$ with $bot \notin M$. Note that by requiring the unsatisfaction of *bot*, we ensure that from M we can retrieve a consistent model of the program $T_{\mathcal{K}}$. Given the ABAF $D_{T_{\mathcal{K}}^{bot}}$ associated to $T_{\mathcal{K}}^{bot}$, we call *consistent* any classical (resp. Φ-maximally classical, Φ-stable) extension E such that $not\_bot \in E$.

Theorem 2. *Assume a KB $\mathcal{K}$ in DL-Lite$_{not}$, from each classical model of $\mathcal{K}$ we can construct a consistent classical extension of $D_{T_{\mathcal{K}}^{bot}}$, where $T_{\mathcal{K}}^{bot}$ is the grounding of $\mathcal{K}$ defined above w.r.t. $N_I(\mathcal{K}) \cup P = \{p_i | 0 \leq i \leq 2 \cdot |N_R(\mathcal{K})|\}$, and vice versa. Given a set of fixed predicates $F \supseteq N_R(\mathcal{K})$, from an F-minimal (resp. F-stable) model $\mathcal{I}$ of $\mathcal{K}$, we can construct construct a consistent Φ-maximally classical (resp. Φ-stable) extension of $D_{T_{\mathcal{K}}^{bot}}$, where $\Phi = \{not\_B(x), not\_r(x, x') | B, r \in F,\ x, x' \in N_I(\mathcal{K}) \cup P\}$, and vice versa.*

6. Conclusion and Future Work

In this paper, we bridged Description Logics and Assumption-Based Argumentation by showing how reasoning in relevant DLs can be obtained in argumentative terms. Our investigations focused on two specific reasoning tasks under the classical semantics and two non-monotonic semantics: the minimal semantics of Circumscription [19], and the stable model semantics of Quantified Equilibrium Logic [17]. First, we considered the task of computing *graph completions* w.r.t. terminologies in the popular $\mathcal{ALCI}$, i.e. possibly recursive collections of inclusions $C \sqsubseteq A$, where A is a concept name. Then, we turned our attention to the reasoning tasks of checking satisfiability of general KBs in DL-Lite$_{not}$, a fragment of the DL-Lite family syntactically resembling logic programs. A shared feature of the two settings is that reasoning is restricted to *finite* structures, allowing us to associate with the input theories a *finite* logic program. In the case of graph completions, we assumed a finite labeled graph which is completed into a model of an input terminology. In the case of satisfiability checking, we exploited the finite model property of DL-Lite$_{not}$ under the considered semantics. Building on the existing translation of LPs into ABA [20], we associate with an input DL theory a unique ABA framework, where different DL semantics are retrieved by uniquely shifting from one argumentative semantics to another.

Our findings open several directions for future research, particularly in integrating the strengths of DLs and ABA. First, we plan to extend our results to other semantics. A first step in this direction is to consider the well-founded semantics for $\mathcal{ALCI}$ terminologies, introduced in the context of SHACL [24]. In fact, through our translation in ABA and existing results for logic programs [20], we could obtain (tight) complexity bounds for DLs in the DL-Lite family. Furthermore, we aim at exploring how other argumentation semantics can be transferred in the context of DLs through our approach. This is particularly attractive in scenarios in which the input DL theory is unsatisfiable, while forms of inconsistency-tolerant reasoning can be adopted from ABA through weaker semantics, e.g., the *admissible*-based semantics, where satisfiability is always guaranteed [7]. Our interests also lean towards implementations of our reasoning tasks. For this, we aim at exploring how *splitting techniques* [25], dialogue-based *explanations* [26] and *abstraction* [27] developed in the context of ABA can be lifted in DLs and investigate their impact on the problem of graph completions.

Acknowledgments

This work was partially supported by the Austrian Science Fund (FWF) project 10.55776/PIN8884924, the Vienna Science and Technology Fund (WWTF) grant 10.47379/ICT25044, and the European Union's Horizon 2020 research and innovation programme (under grant agreement 101034440).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[2] S. Schulz, B. Suntisrivaraporn, F. Baader, M. Boeker, SNOMED reaching its adolescence: Ontologists' and logicians' health check, Int. J. Medical Informatics 78 (2009) S86–S94.

[3] K. Britz, G. Casini, T. Meyer, K. Moodley, U. Sattler, I. Varzinczak, Principles of klm-style defeasible description logics, ACM Trans. Comput. Log. 22 (2021) 1:1–1:46.

[4] L. Giordano, V. Gliozzi, N. Olivetti, G. L. Pozzato, A non-monotonic description logic for reasoning about typicality, Artif. Intell. 195 (2013) 165–202.

[5] P. A. Bonatti, C. Lutz, F. Wolter, The complexity of circumscription in dls, J. Artif. Intell. Res. 35 (2009) 717–773.

[6] F. Di Stefano, M. Simkus, Equilibrium description logics: Results on complexity and relations to circumscription, in: KR, 2024.

[7] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (1995) 321–358.

[8] P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, College Publications, 2018.

[9] A. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artif. Intell. 93 (1997) 63–101.

[10] M. Bienvenu, C. Bourgaux, Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation, in: KR, 2020, pp. 141–151.

[11] G. Alfano, S. Greco, C. Molinaro, F. Parisi, I. Trubitsyna, Extending abstract argumentation frameworks with knowledge bases, in: KR, 2025.

[12] Y. Mahmood, J. Virtema, T. Barlag, A. N. Ngomo, Computing repairs under functional and inclusion dependencies via argumentation, in: FoIKS, 2024, pp. 23–42.

[13] Y. Mahmood, M. Hecher, A. N. Ngomo, Dung's argumentation framework: Unveiling the expressive power with inconsistent databases, in: AAAI, 2025, pp. 15058–15066.

[14] F. Baader, Terminological cycles in kl-one-based knowledge representation languages, in: AAAI, AAAI Press / The MIT Press, 1990, pp. 621–626.

[15] G. De Giacomo, M. Lenzerini, A uniform framework for concept definitions in description logics, J. Artif. Intell. Res. 6 (1997) 87–110.

[16] H. Knublauch, D. Kontokostas, Shapes Constraint Language (W3C SHACL), https://www.w3.org/TR/shacl/, 2017. Accessed: 2026-06-15.

[17] D. Pearce, A. Valverde, Quantified equilibrium logic and foundations for answer set programs, in: ICLP, volume 5366 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 546–560.

[18] F. Di Stefano, M. Simkus, Stable model semantics for description logic terminologies, in: AAAI, AAAI Press, 2024, pp. 10484–10492.

[19] J. McCarthy, Circumscription - A Form of Non-Monotonic Reasoning, Artif. Intell. 13 (1980) 27–39.

[20] M. Caminada, C. Schulz, On the equivalence between assumption-based argumentation and logic programming, J. Artif. Intell. Res. 60 (2017) 779–825.

[21] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The dl-lite family and relations, J. Artif. Intell. Res. 36 (2009) 1–69.

[22] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: ICLP, MIT Press, 1988, pp. 1070–1080.

[23] D. Pearce, Equilibrium logic, Ann. Math. Artif. Intell. 47 (2006) 3–41.

[24] C. Okulmus, M. Simkus, SHACL validation under the well-founded semantics, in: KR, 2024.

[25] G. Buraglio, Splitting assumption-based argumentation frameworks, in: NMR, 2025, pp. 17–31.

[26] G. Buraglio, W. Dvorák, M. König, M. Ulbricht, Justifying argument acceptance with collective attacks: Discussions and disputes, in: IJCAI, 2024, pp. 3281–3288.

[27] I. Apostolakis, Z. G. Saribatur, J. P. Wallner, Abstraction in assumption-based argumentation, in: KR, 2024.