Stable Model Semantics for Description Logic Terminologies

Federica Di Stefano[1], Mantas Šimkus[1,2]

[1]Institute of Logic and Computation, TU Wien, Austria
[2]Department of Computing Science, Umeå University, Sweden
federica.stefano@tuwien.ac.at, simkus@dbai.tuwien.ac.at

Abstract

This paper studies a *stable model semantics* for *Description Logic (DL)* knowledge bases (KBs) and for (possibly cyclic) terminologies, ultimately showing that terminologies under the proposed semantics can be equipped with effective reasoning algorithms. The semantics is derived using *Quantified Equilibrium Logic*, and—in contrast to the usual semantics of DLs based on classical logic—supports default negation and allows to combine the open-world and the closed-world assumptions in a natural way. Towards understanding the computational properties of this and related formalisms, we show a strong undecidability result that applies not only to KBs under the stable model semantics, but also to the more basic setting of minimal model reasoning. Specifically, we show that concept satisfiability in minimal models of an $\mathcal{ALCIO}$ KB is undecidable. We then turn our attention to (possibly cyclic) DL terminologies, where ontological axioms are limited to definitions of concept names in terms of complex concepts. This restriction still yields a very rich setting. We show that standard reasoning problems, like concept satisfiability and subsumption, are ExpTime-complete for terminologies expressed in $\mathcal{ALCI}$ under the stable model semantics.

Introduction

Description Logics (DLs) is a prominent family of languages in the area of *Knowledge Representation and Reasoning*, allowing to model a domain of interest by formalizing relationships between *concepts*, which are written in a convenient yet rich logic-based syntax, and semantically denote classes of objects (Baader et al. 2017). Specifically, DLs underlie the W3C standard OWL for writing *ontologies* in the Semantic Web (Grau et al. 2008), and they are used, e.g., in formalizing and reasoning about complex terminologies in healthcare (Elkin 2023); see (Schneider and Šimkus 2020) for a survey on ontologies and data management.

DLs are often seen as fragments of the classical first-order logic, equipped with a syntax that is more convenient for knowledge representation. In particular, this means that most DLs nowadays make the *open-world assumption (OWA)*, in which, intuitively, everything that is not forbidden is considered possible. However, it is acknowledged that supporting the *closed-world assumption (CWA)* is also important in order to enable commonsense reasoning in DLs; see, e.g., some works based on *circumscription* in (Bonatti, Lutz, and Wolter 2009; Bonatti, Faella, and Sauro 2011; Di Stefano, Ortiz, and Šimkus 2023). This is also witnessed, e.g., by works that combine DL knowledge bases (KBs) and rules with default negation (see, e.g., (Motik and Rosati 2010; Bajraktari, Ortiz, and Šimkus 2018; Lukumbuzya, Ortiz, and Šimkus 2020) and the references therein). The reconciliation of OWA and CWA specifically in DL *terminologies* is a problem whose relevance is boosted by the new W3C SHACL standard for expressing constraints over RDF graphs (Knublauch and Kontokostas 2017). SHACL is syntactically very close to DL terminologies, but its semantics has not been fully established yet (but it clearly leans towards CWA).

A DL terminology $\mathcal{T}$ consists of statements of the form $A := C$, where a concept name A is *defined* using a complex concept expression C. A terminology may contain *terminological cycles*, where a definition of some concept name may be recursive, as in the following terminology:

$$\begin{aligned} \mathsf{BasicUser} &:= User \sqcap \neg \mathsf{PrivilegedUser} \\ \mathsf{PrivilegedUser} &:= Admin \sqcup \exists promotedBy.\mathsf{PrivilegedUser} \end{aligned}$$

Here $\mathsf{BasicUser}$ and $\mathsf{PrivilegedUser}$ are (intensional or *defined*) concept names defined using (extensional) base predicates $User$, $Admin$, and $promotedBy$. A concrete semantics for terminologies tells us how to interpret the defined concept names given an extension for the base predicates. Suppose the base predicates correspond to the following facts:

$$User(a) \quad User(b) \quad promotedBy(a, b) \quad promotedBy(b, a)$$

The standard (*descriptive*) semantics sees ":=" as a logical equivalence; in our example, it produces two possible extensions for the defined concept names:

(i) $\mathsf{BasicUser}(a), \mathsf{BasicUser}(b)$
(ii) $\mathsf{PrivilegedUser}(a), \mathsf{PrivilegedUser}(b)$

While the extension (i) is natural and expected, the extension (ii) is questionable: the membership of a and b in $\mathsf{PrivilegedUser}$ is not well-founded (there is only a self-supported justification). Thus it makes sense to seek a semantics that would reject (ii), but keep (i) as an intended structure. A relevant alternative semantics here is the *least fixpoint semantics* of Baader (1990), which however is too strong in this example: it rejects both (i) and (ii). In (Baader 1990), a given interpretation of *extensional* predicates is extended to a model of the terminology. These extensions can be ordered with a *preference* relation between interpretations and the *preferred* model of a terminology $\mathcal{T}$ is given by the model assigning the minimal possible extension to defined concepts (Baader and Nutt 2003). Such a model might not exist, as in the case above where the extensions in (i) and (ii) are incomparable.

Motivated by examples as above, in this paper we study a new semantics for DL KBs and terminologies that is inspired by logic programming with default negation under the stable model semantics (known as *Answer Set Programming*). The *stable model semantics* of DLs proposed here is stronger than the classical semantics, in the sense that some classical models of a KB or terminology will be rejected as implausible (like (ii) in our example above). This semantics enables default negation and allows to combine the open-world and the closed-world assumptions in a natural way. Furthermore, in the case of DL terminologies in $\mathcal{ALCI}$, reasoning under the stable model semantics is not more expensive than reasoning under the classical semantics.

In a nutshell, the contributions of this paper are as follows:

◦ We define a semantics for general DL knowledge bases using *Quantified Equilibrium Logic (QEL)* (Pearce and Valverde 2008). This yields a new definition of stable models that is not only elegant but also has some other advantages. Specifically, our semantics avoids the process of Skolemization as used in (Gottlob et al. 2021). Moreover, the use of QEL in defining the semantics allows us to elegantly support *fixed predicates*, which are needed for modeling (extensional) base predicates in terminologies. Since standard DLs do not support disjunction of roles, enabling fixed predicates using other frameworks (e.g., via (Ferraris, Lee, and Lifschitz 2011)) would be cumbersome.

◦ We provide a strong undecidability result for reasoning in DLs in the presence of predicate minimization. First, it shows that reasoning in general DL KBs under the proposed semantics is undecidable. However, the proof is given for KBs without negation, which means that it carries over and applies to much simpler settings. Specifically, it shows that concept satisfiability is undecidable in *circumscribed* $\mathcal{ALCIO}$ KBs where *all* predicates are set to be minimized. This complements the negative results in (Bonatti, Lutz, and Wolter 2009), which rely on the use of *varying* predicates.

◦ We define a stable model semantics for DL terminologies. To achieve this, we instantiate our stable model semantics for general KBs. Intuitively, for a given terminology $\mathcal{T}$, we require all base concept and role names of $\mathcal{T}$ to be interpreted as fixed predicates, i.e. they are not subject to minimization. This is natural given the nature of base predicates (see the example above). In addition, we provide two alternative definitions of stable models, which are all equivalent, but are useful to illustrate and analyze different aspects of the proposed semantics. Specifically, we provide definitions based on *level mappings* and *fixpoint computation*.

◦ We show a worst-case optimal complexity result for reasoning in DL terminologies under the proposed semantics. Specifically, we study the case of $\mathcal{ALCI}$ terminologies and prove EXPTIME-completeness of the following problems: (i) deciding the existence of a stable model of a terminology, (ii) deciding concept satisfiability in a stable model of a terminology, and (iii) checking concept subsumption over all stable models of a terminology. This is achieved by proving a *tree-model property* and employing 2-way alternating tree automata (Vardi 1998). Thus, in terms of computational complexity, the stable model semantics for $\mathcal{ALCI}$ terminologies is not more expensive than the classical semantics.

Preliminaries

We recall here $\mathcal{ALCIO}$ 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 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) the expression $\{o\}$, where $o \in N_I$, is a concept;
(d) 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 $\mathcal{A}$.

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 extended to all concept expressions and all roles in the standard way (see, e.g., (Baader et al. 2017)). The "$\models$" relation, and thus the notion of a *(classical) model* of a TBox, ABox, or a KB, are as usual.

Reasoning tasks. Assume a KB $\mathcal{K}$ and concepts C, D. The *satisfiability problem* is to check if $\mathcal{K}$ has a model. The *concept subsumption problem (w.r.t. $\mathcal{K}$)* is to check if $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all models of $\mathcal{I}$ of $\mathcal{K}$. The *concept satisfiability problem (w.r.t. $\mathcal{K}$)* is to check if $\mathcal{K}$ has a model with $C^\mathcal{I} \neq \emptyset$.

Terminologies. A *concept definition* is an expression of the form $A := C$, where $A \in N_C$ and C is a concept. A *terminology* $\mathcal{T}$ is a finite set of concept definitions, where additionally $\{A := C_1, A := C_2\} \subseteq \mathcal{T}$ implies $C_1 = C_2$, i.e. any concept name A can have at most one definition. We say an interpretation $\mathcal{I}$ is a *classical model* of a terminology $\mathcal{T}$ (in symbols, $\mathcal{I} \models \mathcal{T}$), if $A^\mathcal{I} = C^\mathcal{I}$ for all definitions $A := C$ in $\mathcal{T}$. We let $\mathsf{def}(\mathcal{T}) = \{A \mid A := C \in \mathcal{T}\}$ and $\mathsf{base}(\mathcal{T}) = (N_C \cup N_R \cup \{\top\}) \setminus \mathsf{def}(\mathcal{T})$.

Minimal Models (with Fixed Predicates). We recall here a simplified version of circumscription, in which all predicates are either minimized or fixed. Assume a KB $\mathcal{K}$ and a set $F \subseteq N_C \cup N_R$ of predicates. For a pair of interpretations $\mathcal{I}, \mathcal{J}$ sharing the same domain, we write $\mathcal{I} \subseteq_F \mathcal{J}$, if

(i) $a^{\mathcal{I}} = a^{\mathcal{J}}$, for all $a \in N_I$,

(ii) $q^{\mathcal{I}} = q^{\mathcal{J}}$, for all $q \in F$,

(iii) $p^{\mathcal{I}} \subseteq p^{\mathcal{J}}$, for all $p \notin F$.

We write $\mathcal{I} \subset_F \mathcal{J}$, if $\mathcal{I} \subseteq_F \mathcal{J}$ and $\mathcal{J} \not\subseteq_F \mathcal{I}$. We say $\mathcal{J}$ is a *minimal model of $\mathcal{K}$ with fixed predicates* F, if $\mathcal{J} \models \mathcal{K}$ and there is no $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \subset_F \mathcal{J}$. The set of such models is denoted $MM_F(\mathcal{K})$. When a terminology $\mathcal{T}$ is considered instead of a KB $\mathcal{K}$, the above notion is adapted to $MM_F(\mathcal{T})$ in the obvious way (by replacing $\mathcal{K}$ with $\mathcal{T}$).

If $F = \emptyset$, we omit F from the subscript in all cases above, using $MM(\mathcal{K})$ instead of $MM_\emptyset(\mathcal{K})$, and $\subseteq$ instead of $\subseteq_\emptyset$.

Equilibrium Logic and DLs

Equilibrium Logic (EL) (Pearce 1996) is a powerful formalism that allows, e.g., extending the stable model semantics of *Answer Set Programming (ASP)* to arbitrary theories. EL is built upon the logic of *Here-and-There (HT)* with an additional minimality requirement. *Quantified Equilibrium Logic (QEL)* has been introduced in (Pearce and Valverde 2008) as a generalization of EL from the propositional to the first-order setting.

We now introduce an HT semantics for DL KBs, which will later allow to obtain a brief definition of stable models. In contrast to the classical case, an interpretation in the logic HT consists of a pair of structures $(\mathcal{I}, \mathcal{J})$ sharing the same domain, where $\mathcal{I}$ is the 'here' world and $\mathcal{J}$ is the 'there' world. Following the standard nomenclature, we call *assumed* everything that is true 'there', and *founded* everything that is true 'here'. The two worlds are related by the inclusion relation: 'here' is included in 'there'. Formally, HT interpretations and the evaluation of complex concepts in such interpretations are defined as follows.

Definition 1. *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.*

In the HT logic, the implication is intuitionistic: in jargon it needs to be 'founded', meaning that the HT interpretation must model it, and 'assumed', meaning that the 'there' world must model it. In DLs, the universally quantified concept of the form $\forall r.C$ can be translated in FOL as $\forall y((r(x, y) \to C(y))$. Thus the interpretation must align with the interpretation of implication in quantified HT. As a matter of fact, concept inclusions are also affected by this double nature of implication, as they are 'explicit' implications in DLs.

Definition 2. *Assume a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ 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 DL KB.

Definition 3 (Stable model)**.** *Given $F \subseteq N_C \cup N_R$, an interpretation $\mathcal{J}$ is a* stable model *of a KB $\mathcal{K}$* under fixed predicates *F, if*

(i) the HT interpretation $(\mathcal{J}, \mathcal{J})$ 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 $\mathcal{I} \subset_F \mathcal{J}$.

We denote with $SM_F(\mathcal{K})$ the set of all stable models for $\mathcal{K}$ with fixed predicates F. If $F = \emptyset$, we drop the subscript F and write $SM(\mathcal{K})$. We write $\mathcal{K} \models_{sm} C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all $\mathcal{I} \in SM(\mathcal{K})$.

In the semantics introduced above, the negation $\neg$ behaves as *negation as failure* or *default negation* in logic programs. Given an HT model, the 'there' is a classical model and a concept A true at some domain element in the 'there' can be thought of as 'to be justified'. An HT model is not stable if the truth of an atom in the 'there' cannot be proved. Intuitively, the truth of $\neg A$ at a domain element d in a stable model amounts to 'we cannot justify A at d'. Since negation is not classical, knowledge bases that are equivalent under classical semantics might not be equivalent under the stable model semantics.

Example 1. *Assume a graph $G = (V, E)$. For each vertex v_i, we introduce an individual i and a concept V_i. Consider the ABox $\mathcal{A} = \{C(r), C(g), C(b)\} \cup \bigcup_{v_i \in V} \{V_i(i)\}$. For each v_i we now introduce a role p_i to intuitively give to each vertex a color assignment A_i. Let $\mathcal{T}$ be the TBox below:*

$$
\begin{array}{rl}
V_i \sqsubseteq \exists p_i & \textit{for each } v_i \in V \\
\exists p_i^- \sqcap \exists p_j^- \sqsubseteq \bot & \textit{for each } i, j \textit{ s.t. } (v_i, v_j) \in E \\
\exists p_i^- \sqcap \neg C \sqsubseteq \bot & \textit{for each } i \textit{ s.t. } v_i \in V
\end{array}
$$

Let us call $\mathcal{K} = (\mathcal{A}, \mathcal{T})$. It is easy to show that G is 3-colorable iff there exists $\mathcal{I} \in SM(\mathcal{K})$. Intuitively, the last axiom requires that each assignment must correspond to one of the three colors r, g, b, stated in the ABox, as the 'founded' elements of C are only the elements of the ABox.

The example above is a variation of the reduction proposed in (Ngo, Ortiz, and Šimkus 2016) for showing NP-hardness of KB satisfiability in DL-Lite$_{core}$ with closed predicates. The effect of default negation is indeed the same as closed predicates as it forces the set of colors to be restricted to the individuals in the ABox. Under the classical semantics, the KB in Example 1 is equivalent to a KB in DL-Lite$_{core}$. Under the stable model semantics, we cannot replace $\neg C$ on the left-hand side with positive occurrences of C on the right-hand side while preserving the semantics. The latter is underlined by the following example on *access policies*, adapted from (Di Stefano, Ortiz, and Šimkus 2023).

Example 2. *The following KB $\mathcal{K}$ describes the scenario in which classified files can only be read by users holding permission to do so. We want to require that the reading permission* must *be granted by an administrator. We show that under the stable model semantics, we can properly model such a policy. Let $\mathcal{K}$ be as follows:*

$$\mathrm{Classified\_Document}(f_1) \quad \mathrm{User}(John) \quad \mathrm{read}(John, f_1)$$

$$\exists \mathrm{access\_granted\_by.Admin} \sqsubseteq \mathrm{Has\_Read\_Perm}$$

$$\mathrm{Classified\_Document} \sqcap \neg\forall \mathrm{read}^-.\mathrm{Has\_Read\_Perm} \sqsubseteq \bot$$

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

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

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

Figure 1: HT semantics for DLs.

Assume that the predicates Admin *and* $\mathrm{access\_granted\_by}$ *are fixed, i.e. classical. The default negation in the last inclusion implies that every classified document must be read only by users who have permission to do so, thus* John *has permission to read the file* f_1*. However, in any stable model this permission must be 'justified', i.e. an admin is granting it. Thus there exists an* Admin *who gave John access to the classified document. Observe that if the last inclusion is replaced by*

$$\mathrm{Classified\_Document} \sqsubseteq \forall(\mathrm{read})^-.\mathrm{Has\_Read\_Permission}$$

we derive (counterintuitively) that every user reading a classified document automatically acquires permission to do so.

Reductions and Undecidability Results

We discuss here some relationships between reasoning problems and present some negative decidability results. For the sake of simplicity of presentation, we discuss these results assuming the set of fixed predicates F to be empty. All the results here also hold in case $F \neq \emptyset$.

First, we show that for any pair of reasoning tasks, one can be reduced to the other or to the complement of it.

Proposition 1. *Assume a KB* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*, concepts* C, D*, and let* A, r *and* o *be a concept name, role name and individual not occurring in* $\mathcal{K}$*. The following are true:*

(a) $\mathcal{K} \models_{sm} C \sqsubseteq D$ *iff* A *is unsatisfiable w.r.t.* $\mathcal{K}' = (\mathcal{T} \cup \{C \sqcap \neg D \sqsubseteq A\}, \mathcal{A})$*, under the stable model semantics;*

(b) C *is satisfiable w.r.t.* $\mathcal{K}$ *iff*

$$\mathcal{K}' = (\mathcal{T} \cup \{\neg C \sqcap \neg A \sqcap \{o\} \sqsubseteq A\}), \mathcal{A})$$

has a stable model;

(c) $\mathcal{K}$ *has a stable model iff* $\mathcal{K} \not\models_{sm} \top \sqsubseteq \bot$*;*

Identifying a stable model of a KB requires checking minimality (see (ii) in Definition 3), which is computationally difficult. We shall see that it causes undecidability of basic reasoning tasks under the stable model semantics, e.g. for deciding the existence of a stable model. However, the undecidability proof that we provide next is quite strong as it applies to the much more basic setting: specifically, when we are simply interested in classical models that are minimal w.r.t. the $\subseteq$ relation. We show undecidability of checking concept satisfiability in minimal models of an $\mathcal{ALCIO}$ KB. Technically, this setting corresponds to circumscription in DLs (Bonatti, Lutz, and Wolter 2009) where *all* predicates are minimized and no *priorities* among minimized predicates are assumed. This result is interesting in its own right and it complements the undecidability results on circumscribed DLs, relying on the use of *varying* predicates (Bonatti, Lutz, and Wolter 2009).

Theorem 1. *The following problem is undecidable: given a* $\mathcal{ALCIO}$ *KB* $\mathcal{K}$ *and a concept name* A*, check if there exists* $\mathcal{I} \in MM(\mathcal{K})$ *such that* $A^{\mathcal{I}} \neq \emptyset$*.*

Proof (Sketch). We provide a reduction from the undecidable *domino tiling* problem (Berger 1966). Let $P = (T, V, H)$ be an instance of the tiling problem, where T is a finite set of tiles, and $H, V \subseteq T \times T$ are the horizontal and vertical compatibility conditions. For P, we construct a KB $\mathcal{K}_P = (\mathcal{T}_P, \emptyset)$ with TBox $\mathcal{T}_P$ defined as follows:

$$\top \sqsubseteq \exists h.\top \sqcap \exists v.\top \sqcap \exists r^-.\{a\} \tag{1}$$

$$\top \sqsubseteq \bigsqcup_{t \in T} A_t, \qquad A_t \sqcap A_{t'} \sqsubseteq \bot, \text{ for all } t \neq t' \tag{2}$$

$$A_t \sqsubseteq \forall h. \bigsqcup_{(t,t') \in H} A_{t'} \sqcap \forall v. \bigsqcup_{(t,t') \in V} A_{t'} \tag{3}$$

$$\{a\} \sqsubseteq \exists r.X \tag{4}$$

$$X \sqcap \exists h.\exists v.\exists h^-.\exists v^-.X \sqsubseteq \forall r^-.(G \sqcap \forall r.X) \tag{5}$$

It is not too difficult to check that P has a solution iff there is $\mathcal{I} \in MM(\mathcal{T}_P)$ with $G^{\mathcal{I}} \neq \emptyset$. Intuitively, if $\mathcal{I}$ is not a proper grid, then there exists a domain element x that cannot reach itself via an hvh^-v^--path. An interpretation $\mathcal{J} \subset \mathcal{I}$ can be obtained by reducing the extension of X to x only, and G to the empty set. It is easy to prove that $\mathcal{J}$ is a model of $\mathcal{T}_P$, as it still satisfies axiom (4) and (5), deriving a contradiction. Thus $\mathcal{I}$ has to be a proper grid and a solution for P can be easily defined. Conversely, given a solution for P, a minimal model $\mathcal{I}$ of $\mathcal{T}_P$ such that $G^{\mathcal{I}} \neq \emptyset$ is easy to define.

We remark here that this construction can be seen as an application of the *saturation technique* (Eiter and Gottlob 1995) known from disjunctive logic programming to prove a result on DLs. □

Observe that $\mathcal{T}_P$ that we used for Theorem 1 does not use negation '$\neg$' at all. We see next that for $\neg$-free KBs, the minimal model and stable model semantics coincide.

Proposition 2. *For any KB* $\mathcal{K}$ *that does not use '*$\neg$*', and any* $F \subseteq N_C \cup N_R$*, we have* $SM_F(\mathcal{K}) = MM_F(\mathcal{K})$*.*

In Example 1 using the fact that $\neg$ behaves as *default negation* we can enforce the closed world assumption. A similar idea can be used to reduce reasoning w.r.t. KBs in $\mathcal{ALCIO}$ to reasoning w.r.t. KBs in $\mathcal{ALCI}$ under the equilibrium semantics.

Proposition 3. *For an $\mathcal{ALCIO}$ KB $\mathcal{K}$ we can build in polynomial time an $\mathcal{ALCI}$ KB $\mathcal{K}'$ such that: (a) a stable model $\mathcal{I}$ of $\mathcal{K}$ can be extended to a stable model $\mathcal{I}'$ of $\mathcal{K}'$, and (b) if $\mathcal{I}$ is a stable model of $\mathcal{K}'$, then the restriction of $\mathcal{I}$ to the signature of $\mathcal{K}$ is a stable model of $\mathcal{K}$.*

Proof (Sketch). Assume a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ in $\mathcal{ALCIO}$. For all nominals o occurring in $\mathcal{T}$, we introduce two fresh concept names N_o, N_o'. We obtain $\mathcal{K}'$ from $\mathcal{K}$ by replacing each occurrence of $\{o\}$ in $\mathcal{T}$ with N_o', adding the assertions $N_o(o), N_o'(o)$ and adding the inclusion $N_o' \sqcap \neg N_o \sqsubseteq \bot$. The effect of this transformation is a simulation of the nominals o: in every stable model $\mathcal{I}$ the extensions of N_o and N_o' are reduced to a unique element, $o^{\mathcal{I}}$. Note that the transformation requires the addition of ABox assertions. $\square$

As a consequence of the above result, the reductions in Proposition 1 also apply for $\mathcal{ALCI}$ KBs.

Theorem 2. *Under the stable model semantics, a standard reasoning problem in $\mathcal{ALCIO}$ KBs can be reduced in polynomial time to the same problem in $\mathcal{ALCI}$.*

The following theorem follows from Theorem 1, Propositions 2 and 3.

Theorem 3. *Standard reasoning problems in $\mathcal{ALCI}$ under the stable model semantics are undecidable.*

Stable Models for Terminologies

In the previous sections we have seen that reasoning under the stable model semantics in general DL KBs is undecidable. Here we turn our attention to terminologies, which consist of (possibly recursive) definitions of concept names in terms complex concept expressions. A key feature of terminologies is the separation of the predicates of a terminology $\mathcal{T}$ into two sets: $\mathsf{def}(\mathcal{T})$ are the *intensional* predicates that are defined using concept definitions based on the *extensional* predicates in $\mathsf{base}(\mathcal{T})$. In the context of stable model semantics here (or other contexts, like circumscription) it is thus natural to not require minimization of the predicates in $\mathsf{base}(\mathcal{T})$, i.e. the extensions of these predicates should remain fixed during the minimization process. Based on this observation, a stable models semantics for terminologies $\mathcal{T}$ can be immediately obtained by instantiating Definition 3, which covers general KBs: (a) view every concept definition $A := C \in \mathcal{T}$ as an inclusion $C \sqsubseteq A$, and (b) use $F = \mathsf{base}(\mathcal{T})$ as the set of fixed predicates. This leads to a setting that is not only intuitive from the knowledge representation perspective, but that is also decidable and—in terms of complexity theory—not more expensive that reasoning under the usual semantics.

Example 3. *Consider an example, where we reason about the risks of using different components in some production scenario. Consider the following terminology $\mathcal{T}$:*

$$\begin{aligned}\mathsf{RiskyComponent} &:= \neg\mathsf{ValidatedComponent} \sqcup \\ &\quad\ \exists\mathsf{depends}.\mathsf{RiskyComponent} \\ \mathsf{ValidatedComponent} &:= \exists\mathsf{hasCertification}.\top \sqcup \\ &\quad\ \exists\mathsf{supercedes}.\mathsf{RiskyComponent}\end{aligned}$$

We have base roles $\mathsf{depends}$, $\mathsf{hasCertification}$, $\mathsf{supercedes}$. *Specifically,* $\mathsf{supercedes}$ *tells us that one component/part is a new (improved) version of an older one. A component is deemed to be safe as a unit if it has a certification, or if it supercedes an older component that is deemed to be risky. A component is deemed to be risky, if we cannot validate the component, or it depends on a risky component.*

Consider the structure $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{c_1, c_2\}$ and such that $\mathsf{RiskyComponent}^{\mathcal{I}} = \{c_1\}$, $\mathsf{ValidatedComponent}^{\mathcal{I}} = \{c_2\}$, and $\mathsf{hasCertification}^{\mathcal{I}} = \emptyset$, $\mathsf{depends}^{\mathcal{I}} = \{(c_1, c_1)\}$ and $\mathsf{supercedes}^{\mathcal{I}} = \{(c_2, c_1)\}$. This interpretation is a classical model of $\mathcal{T}$, but it is not entirely intuitive. There was no well-founded justification to infer that c_1 is a risky component, and as a consequence there was no reason to validate c_2. The stable model semantics rejects $\mathcal{I}$. If we set $\mathsf{depends}^{\mathcal{I}} = \emptyset$, we obtain an interpretation that is indeed a stable model of the terminology.

We next formalize the above intuition in a stand-alone definition of stable models for $\mathcal{ALCI}$ terminologies. In addition, we provide two alternative definitions, one based on the so-called *level mappings* and another one based on *fixpoint computation*. All three definitions describe the same stable models but the two alternatives provide complementary insights and are useful for obtaining reasoning algorithms.

Stable Models via Equilibrium Logic. Assume a concept definition $A := C$, a terminology $\mathcal{T}$ and an HT interpretation $(\mathcal{I}, \mathcal{J})$. We write:

- $(\mathcal{I}, \mathcal{J}) \models A := C$, if $C^{(\mathcal{I},\mathcal{J})} \subseteq A^{(\mathcal{I},\mathcal{J})}$ and $C^{\mathcal{J}} \subseteq A^{\mathcal{J}}$;
- $(\mathcal{I}, \mathcal{J}) \models \mathcal{T}$, if $(\mathcal{I}, \mathcal{J}) \models A := C$ for all $A := C \in \mathcal{T}$.

We can now define the stable models of $\mathcal{T}$ as follows:

Definition 4. *Assume a terminology $\mathcal{T}$ and let $F = \mathsf{base}(\mathcal{T})$. An interpretation $\mathcal{I}$ is called a* stable model *of $\mathcal{T}$, if*

(i) $(\mathcal{J}, \mathcal{J}) \models \mathcal{T}$, and
(ii) there is no $\mathcal{I}$ s.t. $(\mathcal{I}, \mathcal{J}) \models \mathcal{T}$ and $\mathcal{I} \subset_F \mathcal{J}$.

We note that since all role names are fixed predicates, the semantics of a concept of the form $(\forall R.C)^{(\mathcal{I},\mathcal{J})}$ in Figure 1 simplifies to the equality $(\forall R.C)^{(\mathcal{I},\mathcal{J})} = \{e \in \Delta^{\mathcal{I}} \mid \forall e' : (e, e') \in R^{\mathcal{I}} \text{ implies } C^{(\mathcal{I},\mathcal{J})}\}$.

Stable Models via Level Mappings. The first alternative definition is based on *level mappings*, which, intuitively speaking, ensure a well-founded justification for the membership of objects in defined concept names. The existence of a level mapping in a classical model $\mathcal{I}$ of a terminology $\mathcal{T}$ guarantees that $\mathcal{I}$ is in fact a stable model. Similar characterizations are quite common in ASP (see, e.g., (Janhunen 2004)) and have been recently used in the context of SHACL (Andresel et al. 2020).

Definition 5. *Assume an interpretation $\mathcal{I}$ and a terminology $\mathcal{T}$. We use $\mathsf{conc}(\mathcal{T})$ to denote the set of all concept expressions that appear in $\mathcal{T}$, including the concepts that appear as subconcepts of larger concepts.*

A level mapping *is a strict partial order $\prec$ over $\Delta^{\mathcal{I}} \times \mathsf{conc}(\mathcal{T})$ such that:*

(i) if $e \in C^{\mathcal{I}}$, then $(e, C) \prec (e, A)$ for all $A := C \in \mathcal{T}$,
(ii) if $e \in (C_1 \sqcap C_2)^{\mathcal{I}}$, then $(e, C_i) \prec (e, C_1 \sqcap C_2)$ for all $i \in \{1, 2\}$
(iii) if $e \in (C \sqcup D)^{\mathcal{I}}$, then $(e, C) \prec (e, C \sqcup D)$ or $(e, D) \prec (e, C \sqcup D)$
(iv) if $e \in (\exists r.C)^{\mathcal{I}}$, then $(e', C) \prec (e, \exists r.C)$ for some e' such that $(e, e') \in r^{\mathcal{I}}$
(v) if $e \in (\forall r.C)^{\mathcal{I}}$, then $(e', C) \prec (e, \forall r.C)$ for all e' such that $(e, e') \in r^{\mathcal{I}}$

A level mapping $\prec$ is well-founded if it contains no infinite chain $(e_1, C_1) \succ (e_2, C_2) \succ \cdots$.

We can now define stable models as follows:

Definition 6. *An interpretation $\mathcal{I}$ is a* stable model *of a terminology $\mathcal{T}$, if*

(i) $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all $A := C \in \mathcal{T}$, and
(ii) there exists a well-founded level mapping $\prec$ for $\mathcal{I}$.

Intuitively condition (ii) has a twofold purpose: it ensures that the truth of a defined concept at a given domain element (1) can always be *justified*, decomposing the complex concept defining it, and (2) the chains tracking this justification eventually terminate. Observe that negation "$\neg$" does not even appear in the above definition. Basically, the condition (ii) in Definition 6 checks that $\mathcal{I}$ is the minimal model of $\mathcal{T}$ that is "reduced" w.r.t. $\mathcal{I}$ (in the sense of Gelfond Lifshitz reduct).

Stable Models via Fixpoint Computation. We next characterize stable models in terms of the least fixpoints of immediate consequences operators. Given a terminology $\mathcal{T}$ and an interpretation $\mathcal{I}$, we will define an operator

$$T_{\mathcal{I},\mathcal{T}} \colon 2^{\Delta^{\mathcal{I}} \times \mathsf{conc}(\mathcal{T})} \to 2^{\Delta^{\mathcal{I}} \times \mathsf{conc}(\mathcal{T})}$$

with the intuitive meaning as follows: if S is a set of pairs (e, C) such that the membership $e \in C^{\mathcal{I}}$ is *justified*, then $T_{\mathcal{I},\mathcal{T}}(S)$ produces further pairs (e', D) such that the membership $e' \in D^{\mathcal{I}}$ is also *justified* by S and the terminology $\mathcal{T}$. The operator $T_{\mathcal{I},\mathcal{T}}$ is formally defined as follows:

$$\begin{aligned}
T_{\mathcal{I},\mathcal{T}}(S) = S \cup &\\
&\cup \{(e, C) \mid C \in \mathsf{base}(\mathcal{T}) \wedge e \in C^{\mathcal{I}}\}\\
&\cup \{(e, \neg C) \mid e \in (\neg C)^{\mathcal{I}}\}\\
&\cup \{(e, C_1 \sqcap C_2) \mid (e, C_i) \in S\}\\
&\cup \{(e, \forall r.D) \mid (e', D) \in S \text{ for all } e' \text{ s.t. } (e, e') \in r^{\mathcal{I}}\}\\
&\cup \{(e, \exists r.C) \mid \text{ there is } (e', C) \in S \text{ s.t. } (e, e') \in r^{\mathcal{I}}\}\\
&\cup \{(e, C_1 \sqcup C_2) \mid (e, C_1) \in S \text{ or } (e, C_2) \in S\}\\
&\cup \{(e, A) \mid A := C \in \mathcal{T} \wedge (e, C) \in S\}
\end{aligned}$$

Observe that $T_{\mathcal{I},\mathcal{T}}$ is *monotonic*, i.e., $T_{\mathcal{I},\mathcal{T}}(S) \subseteq T_{\mathcal{I},\mathcal{T}}(S')$ whenever $S \subseteq S'$, and $(2^{\Delta^{\mathcal{I}} \times \mathsf{conc}(\mathcal{T})}, \subseteq)$ is a complete lattice. Thus $T_{\mathcal{I},\mathcal{T}}$ has a least fix-point, reached at some limit ordinal α (Lloyd 1987).

Let us denote with $\mathit{lfp}(T_{\mathcal{I},\mathcal{T}})$ the least fix point of $T_{\mathcal{I},\mathcal{T}}$.

Definition 7. *An interpretation $\mathcal{I}$ is a* stable model *of a terminology $\mathcal{T}$, if*

(i) $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all $A := C \in \mathcal{T}$, and
(ii) $\mathit{lfp}(T_{\mathcal{I},\mathcal{T}}) = \{(e, C) \in \Delta^{\mathcal{I}} \times \mathsf{conc}(\mathcal{T}) \mid e \in C^{\mathcal{I}}\}$.

Theorem 4. *The above three definitions of stable models of terminologies are equivalent.*

Computational Complexity of Reasoning

We provide here our main complexity result for reasoning over terminologies under the introduced semantics.

Theorem 5. *For $\mathcal{ALCI}$ terminologies under the stable model semantics, the problems of satisfiability, concept subsumption, and concept satisfiability are* ExpTime*-complete.*

Note that the reductions between reasoning tasks in Proposition 1 can be easily reformulated for the case of $\mathcal{ALCI}$ terminologies[1]. Hence to prove the above result we focus in the rest of this section on the problem of deciding the existence of a stable model for a given $\mathcal{ALCI}$ terminology. We first observe that this problem is as hard as deciding the existence of a classical model of a $\mathcal{ALCI}$ TBox, which is an ExpTime-complete problem (Schild 1991). Indeed, if $\mathcal{T}$ is a TBox in $\mathcal{ALCI}$, then $\mathcal{T}$ has a (classical) model iff $\mathcal{T}' = \{A := \neg A \sqcap \neg C_{\mathcal{T}}\}$ has a stable model, where $C_{\mathcal{T}} = \sqcap_{C \sqsubseteq D \in \mathcal{T}} (\neg C \sqcup D)$.

To illustrate one of the challenging aspects of obtaining reasoning algorithms, we note that the stable model semantics leads to the loss of the *finite model property* that is enjoyed under the classical semantics of $\mathcal{ALCI}$ (Baader et al. 2017). This is illustrated via the following example:

Example 4. *Let $\mathcal{T}$ consist of the following:*

$$A := \neg\neg\exists r.A \qquad (A)$$
$$B := \forall r^-.B \qquad (B)$$
$$C := \neg B \sqcap \neg C \qquad (C)$$

Observe that $\mathcal{T}$ has an infinite stable model with $A^{\mathcal{I}} \neq \emptyset$. Indeed, take $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{1, 2, 3, \ldots\}$ and such that $A^{\mathcal{I}} = B^{\mathcal{I}} = \Delta^{\mathcal{I}}$, and $C^{\mathcal{I}} = \emptyset$. Suppose a finite stable model $\mathcal{J}$ of $\mathcal{T}$ exists such that $A^{\mathcal{J}} \neq \emptyset$. Due to (A), there is an infinite sequence $e_0 \in A^{\mathcal{J}}, e_1 \in A^{\mathcal{J}}, \ldots$ of elements such that $(e_i, e_{i+1}) \in r^{\mathcal{J}}$ holds for all $i \geq 0$. Since $\mathcal{J}$ is finite, it contains a cycle "formed" by the directed r-edges. Specifically, there exist $0 \leq j_1 < j_2$ such that $e_{j_1} = e_{j_2}$. Due to (B), we have that none of the vertices of this cycle can participate in $B^{\mathcal{J}}$, otherwise $\mathcal{J}$ is not stable as each occurrence of B in the cycle does not have a well-founded

[1]Since terminologies do not allow nominals or ABoxes, the only interesting case is point (b) in Proposition 1. Observe that checking satisfiability of a concept C w.r.t. a terminology $\mathcal{T}$ reduces to checking if $\mathcal{T}' = \mathcal{T} \cup \{B := \neg\exists r.C \sqcap \neg B\}$ has a stable model, where r is a fresh role name, and B is a fresh concept name.

justification, i.e. no well-founded level mapping for $\mathcal{J}$ can be found. The last concept definition (C) is simply a constraint that tells us that B must be proven at every element of a stable model. Putting these two observations together we get that $\mathcal{J}$ cannot be a stable model of $\mathcal{T}$.

To check the existence of a stable model for a terminology $\mathcal{T}$ in deterministic exponential time, we apply similar techniques as in (Vardi 1998; Sattler and Vardi 2001) for μ-calculus (Kozen 1983). The key insight here is the *tree model property*: we show that if $\mathcal{T}$ has a stable model, then $\mathcal{T}$ has a stable model that is shaped like a tree. We then define a two-way alternating automaton over infinite trees (Muller and Schupp 1987) that accepts exactly the trees corresponding to tree-shaped stable models of a given terminology.

Given a concept C, we denote with $nnf(C)$, the *negation normal form* of C.

Definition 8 (Types). *Given a terminology $\mathcal{T}$, we denote with $cl(\mathcal{T})$ the smallest set of concepts that contains $nnf(C)$ and $nnf(\neg C)$ for each concept C that appears in $\mathcal{T}$ (possibly as a subconcept of a larger concept). A* type *for $\mathcal{T}$ is any set $\tau \subseteq cl(\mathcal{T})$ such that:*

(a) for all $A := C \in \mathcal{T}$, $A \in \tau$ iff $nnf(C) \in \tau$;
(b) for all $C \in cl(\mathcal{T})$, either $C \in \tau$ or $nnf(\neg C) \in \tau$;
(c) for all $C \sqcup D \in cl(\mathcal{T})$, $C \sqcup D \in \tau$ iff $C \in \tau$ or $D \in \tau$;
(d) for all $C \sqcap D \in cl(\mathcal{T})$, $C \sqcap D \in \tau$ iff $C \in \tau$ and $D \in \tau$.

We now define the notion of *pre-models*, which essentially characterizes classical models of a terminology.

Definition 9 (Pre-models). *A* pre-model *of a terminology $\mathcal{T}$ is a pair $(\mathcal{I}, \pi)$ of an interpretation $\mathcal{I}$ and a mapping π that assigns to every $e \in \Delta^{\mathcal{I}}$ some type $\pi(e)$ for $\mathcal{T}$ such that:*

- *for all concept names A, $e \in A^{\mathcal{I}}$ iff $A \in \pi(e)$;*
- *for all $\exists r.C \in cl(\mathcal{T})$, if $\exists r.C \in \pi(e)$ then there exists e' such that $(e, e') \in r^{\mathcal{I}}$ and $C \in \pi(e')$;*
- *for all $\forall r.C \in cl(\mathcal{T})$, if $\forall r.C \in \pi(e)$ then for all e' with $(e, e') \in r^{\mathcal{I}}$ we have $C \in \pi(e')$.*

We next define the notion of *choice function*, which will help us to keep track of justifications of defined concepts.

Definition 10 (Choice function). *A choice function ch for a pre-model $(\mathcal{I}, \pi)$ of $\mathcal{T}$ is a partial function*

$$ch : \Delta^{\mathcal{I}} \times cl(\mathcal{T}) \to \Delta^{\mathcal{I}} \cup cl(\mathcal{T})$$

such that:

(i) if $C \sqcup D \in \pi(e)$, then $ch(e, C \sqcup D) \in \pi(e) \cap \{C, D\}$;
(ii) if $\exists r.C \in \pi(e)$, then $ch(e, \exists r.C) = e'$ for some e' such that $(e, e') \in e^{\mathcal{I}}$ and $C \in \pi(e')$.

A triple $(\mathcal{I}, \pi, ch)$ is an adorned *pre-model for $\mathcal{T}$.*

Intuitively, pre-models are candidates for stable models. To check that an adorned pre-model is stable we use a further relation that tracks justifications of positive concepts.

Definition 11. *Assume an adorned pre-model $(\mathcal{I}, \pi, ch)$ for $\mathcal{T}$. For this structure we define the binary* derivation relation *$\rightsquigarrow$ over $\mathsf{conc}(\mathcal{T}) \times \Delta^{\mathcal{I}}$ such that:*

- *if $A \in \pi(e)$ and $A := C \in \mathcal{T}$, then $(A, e) \rightsquigarrow (C, e)$;*
- *if $C \sqcup D \in \pi(e) \cap \mathsf{conc}(\mathcal{T})$, then $(C \sqcup D, e) \rightsquigarrow (ch(C \sqcup D, e), e)$;*
- *if $C \sqcap D \in \pi(e) \cap \mathsf{conc}(\mathcal{T})$, then $(C \sqcap D, e) \rightsquigarrow (C, e)$ and $(C \sqcap D, e) \rightsquigarrow (D, e)$;*
- *if $\forall r.C \in \pi(e) \cap \mathsf{conc}(\mathcal{T})$, then $(\forall r.C, e) \rightsquigarrow (C, e')$ for all e' with $(e, e') \in r^{\mathcal{I}}$;*
- *if $\exists r.C \in \pi(e)$, then $(\exists r.C, e) \rightsquigarrow (C, ch(\exists r.C, e))$.*

We say $\rightsquigarrow$ is well-founded *if it contains no infinite chain $\ell_1 \rightsquigarrow \ell_2 \rightsquigarrow \cdots$. In this case, $(\mathcal{I}, \pi, ch)$ is also* well-founded.

The above-defined "decorations" of an interpretation allow us to recognize stable models of a terminology. Observe that there is a clear correspondence between the notion of level mapping and the derivation relation. In a stable model $\mathcal{I}$ the well-founded level mapping can be used to identify a well-founded derivation relation; vice versa given a well-founded adorned pre-model $(\mathcal{I}, \pi, ch)$, the derivation relation can be used to define a well-founded level mapping. The latter intuition is exploited in the following result.

Lemma 1. *An interpretation $\mathcal{I}$ is a stable model of a terminology $\mathcal{T}$ iff there exist π, ch s.t. $(\mathcal{I}, \pi, ch)$ is an adorned pre-model of $\mathcal{T}$ whose derivation relation is well-founded.*

For an interpretation $\mathcal{I}$, its *graph* $G_{\mathcal{I}}$ is the graph whose nodes are $\Delta^{\mathcal{I}}$ and such that there is an edge between v and v' if $(v, v') \in r^{\mathcal{I}}$ for some role name r. An interpretation $\mathcal{I}$ is tree-shaped if $G_{\mathcal{I}}$ is a tree.

Theorem 6. *If a terminology $\mathcal{T}$ has a well-founded adorned pre-model then it has a* tree-shaped *well-founded adorned pre-model with branching degree bounded by the size of $\mathcal{T}$.*

We prove the result following the technique used in (Sattler and Vardi 2001): a given model $\mathcal{I}$ is *unraveled* into a tree-shaped one using π to keep the branching degree bounded by the size of $\mathcal{T}$ and the function ch to preserve the well-foundedness of the derivation relation.

Theorem 6 shows that to decide the existence of a stable model for $\mathcal{T}$ it suffices to search for a tree-shaped well-founded adorned pre-model for $\mathcal{T}$.

Theorem 7. *Given a terminology $\mathcal{T}$, we can construct a 2ATA $\mathbf{A}$ (with Büchi acceptance condition) whose number of states is polynomial in the size of $\mathcal{T}$ and such that $\mathcal{T}$ has a tree-shaped well-founded adorned pre-model iff $\mathbf{A}$ is not empty, i.e. $\mathbf{A}$ accepts a tree.*

The automaton $\mathbf{A}$ is the intersection of two automata $\mathbf{A}_M$ and $\mathbf{A}_F$ that operate on labeled trees where, roughly speaking, a node stores the concept names satisfied by the object together with the roles that connect it to the parent (if it is not the root). The first automaton $\mathbf{A}_M$ checks that the input tree is a classical model of the terminology, it can be constructed in the usual way (Calvanese, Eiter, and Ortiz 2007). The second automaton $\mathbf{A}_F$ tracks the justifications for defined concepts. It can be constructed in a similar way as in (Sattler and Vardi 2001) and uses the characterization of stable models given by well-founded adorned pre-models given by Lemma 1, that is indeed tightly related to *level mappings*. Both $\mathbf{A}_M$ and $\mathbf{A}_F$ require only polynomially many states in the size of $\mathcal{T}$.

The emptiness of 2ATA $\mathbf{A}$ can be checked in exponential time in the number of states, (Vardi 1998). Thus, from

Lemma 1, Theorem 6 and Theorem 7, checking the existence of a stable model for a terminology $\mathcal{T}$ can be done in deterministic exponential time in the size of $\mathcal{T}$.

Related Work

Fixpoint-based Approaches. In the last three decades, essentially three semantics have been proposed to deal with terminologies: the *least fixpoint semantics*, the *greatest fixpoint semantics*, and the *descriptive semantics*. The *descriptive semantics* (Nebel 1991) is nowadays the classical semantics for DLs, deemed as the most natural semantics. Baader (1990) advocates fixpoint semantics to overcome some weaknesses of the descriptive semantics. Intuitively, the main difference between the latter and our approach is as follows: the approach of Baader (1990) takes an interpretation $\mathcal{J}$ of the base predicates and then selects the 'best' extension of $\mathcal{J}$ modeling the terminology. The latter is selected as the least fixpoint of an operator, that might not exist if the TBox is not monotonic; our approach instead takes a model of terminology (thus the extension of *all* predicates is given) and checks that occurrences of the defined concept are justified. For the terminologies in $\mathcal{FL}_0$, the semantics of (Baader 1990) and our coincide, i.e. we select the same stable models. De Giacomo and Lenzerini (1997) proposed an extension of the syntax with fixpoint operators, based on μ-calculus. A drawback of the latter, shared with the approaches proposed in (Baader 1990; Schild 1994), is the requirement on the *syntactic monotonicity* of terminologies.

Tuple Generating Dependencies (TGDs). (Gottlob et al. 2021) presents a stable model semantics that can be used for TGDs as well as for some DLs. The semantics there uses Skolemization, i.e., dedicated new objects are used in order to satisfy existential quantifiers. This approach brings benefits in terms of decidability, but it is not fully intuitive as argued in (Alviano, Morak, and Pieris 2017). The latter work advocates a definition based on a formula in second-order logic from (Ferraris, Lee, and Lifschitz 2011). The results for TGDs of (Alviano, Morak, and Pieris 2017) regard query answering, and use syntactic constructs that are not allowed in standard DLs. The semantics proposed in this work avoids the use of a translation of the terminologies into a richer framework, as second-order logic. This facilitated our analysis of the model-theoretic properties and computational complexity of the introduced formalism.

Other approaches. Similarly to Equilibrium Logic, the logic of *minimal knowledge and negation as failure (MKNF)* (Lifschitz 1994) generalizes the stable model semantics of logic programs. Based on MKNF, extensions of expressive DLs have been proposed in (Donini, Nardi, and Rosati 2002) and in (Motik and Rosati 2010). In contrast, our non-monotonic extension of DLs directly adopts the semantics of QEL, which leads to a self-contained formalism that does not rely on a translation into a more expressive logic. Furthermore, the authors in (Donini, Nardi, and Rosati 2002) apply different syntactic restrictions to ensure decidability, and as a consequence, different reasoning algorithms are used. Differently from (Motik and Rosati 2010), in the approach we proposed in this work the non-monotonicity is not carried by the integration of rules but occurs already at the level of the knowledge base. The latter aspect also differentiates our approach from the one in (Levy and Rousset 1998). In (Heymans, Nieuwenborgh, and Vermeir 2006), the authors consider the so-called *conceptual logic programs*, where only unary and binary predicates are allowed, under the *open answer set* semantics. They impose decidability-ensuring restrictions on the programs that are similar to those used for terminologies. The proposed formalism can capture DLs under the classical semantics however does not directly extend the stable model semantics to them.

SHACL constraints. DL terminologies are related to constraints over RDF graphs expressed using the recent SHACL standard of W3C (Knublauch and Kontokostas 2017): defined concept names correspond to shape names, basic predicates are classes or properties, depending on the arity. In particular, (Andresel et al. 2020) presents a stable model semantics for SHACL using a definition based on level mappings. A large fragment of SHACL constraints from (Andresel et al. 2020) can be directly translated into $\mathcal{ALCIO}$ terminologies, while essentially preserving a correspondence between stable models. We say "essentially" because in this paper (as customary in DL research) infinite interpretations are supported, while this is not the case in (Andresel et al. 2020). Based on this correspondence, our ExpTime membership result here can be applied for reasoning about SHACL constraints under the stable model semantics. Specifically, checking the existence of a (possibly infinite) RDF graph that satisfies a given shape name under a given set of SHACL constraints (corresponding to $\mathcal{ALCI}$ concepts) can be performed in single exponential time.

Conclusion

In this paper, we have investigated a *stable model semantics* for general DL KBs and cyclic DL terminologies. Among our insights is a positive complexity results for $\mathcal{ALCI}$ terminologies as well as some negative results that apply even to the more basic settings of minimal model reasoning.

For future work, we expect that our result for $\mathcal{ALCI}$ can be extended to $\mathcal{ALCIO}$ using ideas from (Sattler and Vardi 2001). *Finite model reasoning* in terminologies under the stable model semantics is also a relevant open problem: it has the potential to provide new insights into, e.g., the complexity of static analysis problems for SHACL. Another natural direction (also relevant for SHACL) is to study terminologies that support *regular expressions* over roles, which enable recursive navigation of paths in an interpretation.

Acknowledgements

This work was partially supported by the Austrian Science Fund (FWF) project P30873 and by the Wallenberg AI, Autonomous Systems, and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

References

Alviano, M.; Morak, M.; and Pieris, A. 2017. Stable Model Semantics for Tuple-Generating Dependencies Revisited. In *Proc. of PODS 2017*, 377–388. ACM.

Andresel, M.; Corman, J.; Ortiz, M.; Reutter, J. L.; Savkovic, O.; and Šimkus, M. 2020. Stable Model Semantics for Recursive SHACL. In *Proc. of WWW 2020*, 1570–1580. ACM / IW3C2.

Baader, F. 1990. Terminological Cycles in KL-ONE-based Knowledge Representation Languages. In *Proc. of AAAI 1990*, 621–626. AAAI Press / The MIT Press.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Baader, F.; and Nutt, W. 2003. Basic Description Logics. In *Description Logic Handbook*, 43–95. Cambridge University Press.

Bajraktari, L.; Ortiz, M.; and Šimkus, M. 2018. Combining Rules and Ontologies into Clopen Knowledge Bases. In *Proc. of AAAI 2018*, 1728–1735. AAAI Press.

Berger, R. 1966. The undecidability of the domino problem. Number 66 in Memoirs of the American Mathematical Society. *The American Mathematical Society*, 202.

Bonatti, P. A.; Faella, M.; and Sauro, L. 2011. Defeasible Inclusions in Low-Complexity DLs. *J. Artif. Intell. Res.*, 42: 719–764.

Bonatti, P. A.; Lutz, C.; and Wolter, F. 2009. The Complexity of Circumscription in Description Logic. *J. Artif. Intell. Res.*, 35(1): 717–773.

Calvanese, D.; Eiter, T.; and Ortiz, M. 2007. Answering Regular Path Queries in Expressive Description Logics: An Automata-Theoretic Approach. In *Proc. of AAAI 2007*, 391–396. AAAI Press.

De Giacomo, G.; and Lenzerini, M. 1997. A Uniform Framework for Concept Definitions in Description Logics. *J. Artif. Intell. Res.*, 6: 87–110.

Di Stefano, F.; Ortiz, M.; and Šimkus, M. 2023. Description Logics with Pointwise Circumscription. In *Proc. of IJCAI 2023*, 3167–3175. ijcai.org.

Donini, F. M.; Nardi, D.; and Rosati, R. 2002. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2): 177–225.

Eiter, T.; and Gottlob, G. 1995. On the Computational Cost of Disjunctive Logic Programming: Propositional Case. *Ann. Math. Artif. Intell.*, 15(3-4): 289–323.

Elkin, P. L., ed. 2023. *Terminology, Ontology and their Implementations*. Springer Cham.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2011. Stable models and circumscription. *Artif. Intell.*, 175(1): 236–263.

Gottlob, G.; Hernich, A.; Kupke, C.; and Lukasiewicz, T. 2021. Stable Model Semantics for Guarded Existential Rules and Description Logics: Decidability and Complexity. *J. ACM*, 68(5): 35:1–35:87.

Grau, B. C.; Horrocks, I.; Motik, B.; Parsia, B.; Patel-Schneider, P.; and Sattler, U. 2008. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6(4): 309–322. Semantic Web Challenge 2006/2007.

Heymans, S.; Nieuwenborgh, D. V.; and Vermeir, D. 2006. Conceptual logic programs. *Ann. Math. Artif. Intell.*, 47(1-2): 103–137.

Janhunen, T. 2004. Representing Normal Programs with Clauses. In *Proc. of ECAI 2004*, ECAI'04, 358–362. NLD: IOS Press. ISBN 9781586034528.

Knublauch, H.; and Kontokostas, D. 2017. Shapes constraint language (W3C SHACL). https://www.w3.org/TR/shacl/. Accessed: 2024-01-24.

Kozen, D. 1983. Results on the Propositional mu-Calculus. *Theor. Comput. Sci.*, 27: 333–354.

Levy, A. Y.; and Rousset, M. 1998. Combining Horn Rules and Description Logics in CARIN. *Artif. Intell.*, 104(1-2): 165–209.

Lifschitz, V. 1994. Minimal belief and negation as failure. *Artif. Intell.*, 70(1): 53–72.

Lloyd, J. W. 1987. *Foundations of Logic Programming, 2nd Edition*. Springer.

Lukumbuzya, S.; Ortiz, M.; and Šimkus, M. 2020. Resilient Logic Programs: Answer Set Programs Challenged by Ontologies. In *Proc. of AAAI 2020*, 2917–2924. AAAI Press.

Motik, B.; and Rosati, R. 2010. Reconciling description logics and rules. *J. ACM*, 57(5): 30:1–30:62.

Muller, D. E.; and Schupp, P. E. 1987. Alternating Automata on Infinite Trees. *Theor. Comput. Sci.*, 54: 267–276.

Nebel, B. 1991. Terminological Cycles: Semantics and Computational Properties. In Sowa, J. F., ed., *Principles of Semantic Networks - Explorations in the Representation of Knowledge*, The Morgan Kaufmann Series in representation and reasoning, 331–361. Morgan Kaufmann.

Ngo, N.; Ortiz, M.; and Šimkus, M. 2016. Closed Predicates in Description Logics: Results on Combined Complexity. In *Proc. of KR 2016*, 237–246. AAAI Press.

Pearce, D. 1996. A New Logical Characterisation of Stable Models and Answer Sets. In *NMELP*, volume 1216 of *Lecture Notes in Computer Science*, 57–70. Springer.

Pearce, D.; and Valverde, A. 2008. Quantified Equilibrium Logic and Foundations for Answer Set Programs. In *Proc. of ICLP 2008*, volume 5366 of *Lecture Notes in Computer Science*, 546–560. Springer.

Sattler, U.; and Vardi, M. Y. 2001. The Hybrid μ-Calculus. In *Proc. of IJCAR 2001*, volume 2083 of *Lecture Notes in Computer Science*, 76–91. Springer.

Schild, K. 1991. A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI 1991*, 466–471. Morgan Kaufmann.

Schild, K. 1994. Terminological Cycles and the Propositional μ-Calculus. In *Proc. of KR 1994*, 509–520. Morgan Kaufmann.

Schneider, T.; and Šimkus, M. 2020. Ontologies and Data Management: A Brief Survey. *Künstliche Intell.*, 34(3): 329–353.

Vardi, M. Y. 1998. Reasoning about The Past with Two-Way Automata. In *Proc. of ICALP 1998*, volume 1443 of *Lecture Notes in Computer Science*, 628–641. Springer.