

Extending Description Logics with Generic Concepts – the Case of Terminologies

Joshua Hirschbrunn¹, Yevgeny Kazakov¹

¹University of Ulm, James-Franck-Ring, 89081 Ulm, Germany
joshua.hirschbrunn@uni-ulm.de, yevgeny.kazakov@uni-ulm.de

Abstract

We propose an extension of Description Logics (DLs) with generic concepts and conditional axioms. Inspired by object-oriented languages, generic concepts allow a compact definition of concepts with similar structures. For example, one can define a generic concept $\text{Owner}[X]$ to describe objects that own another object from X , and later use a specific replacement of the parameter X , such as $\text{Owner}[\text{Pet}]$ representing pet owners. Conditional axioms can be used to set bounds on the values that replace the generic parameters. For example, we could restrict replacements of X in a concept $\text{Keeper}[X]$ to only subconcepts of Pet . As the set of possible parameter replacements can be infinite and even uncountable, the generic extensions are, in general, undecidable. To identify decidable generic DLs, we focus on the case of terminologies, requiring that variables are only used in definitions of generic concepts. We formulate restrictions that allow a reduction of generic entailment to classical entailment and further conditions that ensure decidability.

1 Introduction

Many large Description Logics (DLs) ontologies exhibit regularities in their syntactic structure (Mikroyannidi et al. 2011, 2012), and there have been several proposals to model such regularities within the languages so that ontologies are easier to maintain (Gangemi and Presutti 2009; He, Zheng, and Lin 2015; Skjæveland et al. 2018; Krieg-Brückner, Mossakowski, and Neuhaus 2019; Skjæveland et al. 2017; Borgida et al. 2012; Kindermann, Parsia, and Sattler 2019; Kindermann et al. 2018, 2024). One proposal is to apply the principles of generic programming for object-oriented languages (Garcia et al. 2003) to ontologies. Generic DLs (Hirschbrunn and Kazakov 2024) extend classical DLs with two new features: concept variables and parametrized concepts. *Concept variables* are placeholders that can be replaced with (ordinary) concepts. For example, a *generic concept* $\exists \text{owns}.X$ uses a concept variable X , which could be replaced with (ordinary) concepts like Pet or Car resulting in (ordinary) concepts $\exists \text{owns}.\text{Pet}$ and $\exists \text{owns}.\text{Car}$. *Parametrized concepts* are a generalized form of atomic concepts, whose meaning may depend on other concepts. For example, a parameterized concept $\text{Owner}[X]$ can be used to de-

scribe owners of objects from X , and could be defined using a *generic axiom* $\text{Owner}[X] \equiv \exists \text{owns}.X$. Thus, $\text{Owner}[\text{Pet}]$ and $\text{Owner}[\text{Car}]$ describe two different kinds of owners.

Generic axioms can be interpreted in two ways: using the schema semantics and using the second-order semantics (Hirschbrunn and Kazakov 2024). Under the *schema semantics*, the axiom $\text{Owner}[X] \equiv \exists \text{owns}.X$ is regarded *literally* as an abbreviation of (countably many) axioms $\text{Owner}[C] \equiv \exists \text{owns}.C$ obtained by replacing the concept variable X with all possible concepts C from the language. Under the *second-order semantics*, concept variables can be replaced with arbitrary subsets of the interpretation domain. Second-order semantics is, generally, *stronger* than the schema semantics because not every subset of a domain is an interpretation of some concept of the language. However, entailment under the schema semantics can be computed using standard DL algorithms by treating instances of parametrized concepts such as $\text{Owner}[\text{Pet}]$ and $\text{Owner}[\text{Car}]$ as distinct atomic concepts. Schema entailment, however, may depend on the *language* in which the replacement concepts C are constructed: replacing with \mathcal{EL} concepts may result in fewer entailments than replacing with \mathcal{ALC} concepts (and fewer than for the second-order entailment). Therefore, a central question for generic DLs is, when both semantics result in the same entailments.

The previous work on generic DLs (Hirschbrunn and Kazakov 2024) shows that it is possible to ensure that second-order entailment coincides with the schema entailment by limiting the base language to the DL \mathcal{EL} and applying further syntactic restrictions. In this paper, we use a different approach: Instead of restricting the type of *concept constructors* that can be used in ontologies, we restrict the *shape of axioms*. Specifically, we allow *definitions* of (generic) atomic concepts of the form $A[X_1, \dots, X_n] \equiv C$ where the left-hand side is a parametrized concept and the right-hand side C is an arbitrary (*SROIQ*) concept containing only the variables X_i ($1 \leq i \leq n$) present on the left. A *terminology* is a set of such concept definitions in which every concept is defined at most once. We can also allow *partial concept definitions* of the form $A[X_1, \dots, X_n] \sqsubseteq C$, where the partially defined atomic concept may appear in several such partial definitions, because they can be rewritten to concept definitions using concept conjunction and fresh atomic concepts. We also allow the ontology to con-

tain any number of *ground axioms*, i.e., axioms that do not contain concept variables.

Sometimes, it is useful to further restrict (partial) definitions by limiting the scope of parameters for which they can be used. For this purpose, we introduce a new type of axiom called *conditional axiom*. Similar to bounds in generic programming (Garcia et al. 2003), conditional axioms allow for restricting the concepts/domain-subsets that are considered for concept variables. A conditional axiom consists of a range of conditions and a target axiom: $\{\gamma_1, \dots, \gamma_n\} \Rightarrow \beta$, the conditions, as well as the target axiom, are classical axioms (potentially using generic concepts). For example, this allows us to specify different kinds of contents: $X \sqsubseteq \text{File} \Rightarrow \text{Contents}[X] \sqsubseteq \text{Data}$, $X \sqsubseteq \text{Food} \Rightarrow \text{Contents}[X] \sqsubseteq \text{Nutrients}$, $X \sqsubseteq \text{Law} \Rightarrow \text{Contents}[X] \sqsubseteq \text{Paragraphs}$. If a parameterized concept is applied to an argument that is not satisfied by the conditions, we consider it to be undefined, and it can be interpreted arbitrarily, which aligns with the idea of partial definitions. The terminological part of ontologies that we specify can also be cyclic. For example, our DL extension is able to capture recursive data definitions such as a node of a tree-structure: $\text{Node}[X] \equiv \forall \text{hasSuccessor. Node}[X] \sqcap \exists \text{hasValue.X}$

In this paper, we present the following results. First, we show that reasoning under classical ontologies extended with (only) conditional axioms can be reduced to reasoning with classical ontologies allowing for negated axioms (Section 4). Second, using the conditional axioms, we show a reduction of reasoning with ground generic ontologies to reasoning with classical ontologies (with conditional axioms) (Section 5). Third, we show that we can reduce reasoning with a non-ground generic terminology to reasoning with a ground generic ontology using a fixpoint approach; this approach allows us to extend a model of the ground part of a generic terminology to a model of the whole ontology (Section 6). Finally, we also explain how the semantic restrictions from Section 6 can be achieved syntactically in practice (Section 7).

2 Related Work

As mentioned in the introduction, our work is based on the existing generic extension of description logics proposed earlier (Hirschbrunn and Kazakov 2024). We differ from that work in the way we restrict the usage of generic features to keep decidability. Instead of restricting the generic DL to a fragment of the extension of \mathcal{EL} , we work with the extension of DLs up to \mathcal{SROIQ} , but require that axioms with variables are part of a terminology where each parameterized concept is defined at most once (multiple partial definitions are also allowed). Additionally, we introduce a new feature of the generic extension in this paper, namely, conditional axioms that allow us to restrict the range of concept variables using classical axioms.

Apart from generic extensions, our approach is related to several other areas. Ontology parts of similar syntactic structure are primarily studied in the field of Ontology Design Patterns (ODPs) (Gangemi and Presutti 2009; He, Zheng, and Lin 2015; Skjæveland et al. 2018; Krieg-Brückner,

Mossakowski, and Neuhaus 2019; Skjæveland et al. 2017; Borgida et al. 2012; Kindermann, Parsia, and Sattler 2019). Similar to our reduction of second-order to classical reasoning, this method employs variables to create axiom templates for deriving standard axioms tailored to particular applications. The key distinction from our approach is that ODPs lack true generic concepts like $\text{Owner}[X]$ and do not possess model-theoretic semantics. Rather, ODPs are generally a preliminary stage, substituting variables with concepts from predetermined sets of candidates to form a classical ontology that can subsequently be used in the standard way. There are also works in this area that are primarily concerned with finding repeating structures in ontologies (usually with the goal of testing and defining new ODPs) (Mikroyannidi et al. 2011, 2012; Lawrynowicz et al. 2018).

A related but different concept are the Generators introduced by Kindermann et al. (Kindermann et al. 2018). Those are a kind of rule language on top of DLs, consisting of rules called generators that have axiom templates as conditions and targets. If a certain replacement of variables in the conditions results in an axiom that is entailed by the ontology, then the target is added as an axiom to the ontology for the same variable replacement. This is somewhat similar to our conditional axioms, with the most important difference being that usage of these generators requires the (manual) specification of a *language* of replacement concepts, while in our case, the second-order semantics considers arbitrary subsets of the domain as replacements of the concept variables.

Additionally, in the broader context of DL research, the idea of making axioms depend on other axioms in the DL itself, like we do with conditional axioms, is also not completely new. One example are so-called context description logics (Klarman and Gutiérrez-Basulto 2010), which allow axioms to be dependent on a concrete context, which is itself also formulated as a DL axiom. They differ from conditional axioms in that contexts are defined outside the ontology in a special context ontology; there is no direct connection between elements of the context axiom and the classical axiom targeted, as we can establish by sharing variables; and finally, context DLs are a kind of multi-modal DL allowing to formulate a form of possibility and necessity on contexts, which we do not support. We also differ from DL rules (e.g., (Krötzsch, Rudolph, and Hitzler 2008)) because we do not leave the DL language to formulate rules in First-Order Logic (FOL), but formulate conditional axioms directly in the DL of the ontology itself and we quantify over *concept* variables, not variables for individuals.

3 Generic Extension

We start by formally defining the syntax of generic DLs with parameterized concepts, concept variables, and conditional axioms.

Definition 1 (Syntax, extended from (Hirschbrunn and Kazakov 2024)). *The syntax of generic DLs consists of disjoint and countably infinite sets N_C of concept names, each with an assigned arity $\text{ar}(A) \in \mathbb{N}$ ($A \in N_C$), N_R of role names, and N_X of concept variables. Given a base DL L*

that is a fragment of *SRIOQ*, we define by \mathcal{LX} its corresponding generic extension, adding parameterized (atomic) concepts, concept variables, and conditional axioms. Specifically, the set of \mathcal{LX} -concepts is the smallest set containing concept variables $X \in N_X$, classical atomic concepts $A \in N_C$ with $\text{ar}(A) = 0$, concept terms $A[C_1, \dots, C_n]$, where $A \in N_C$, $n = \text{ar}(A) \geq 1$ and C_1, \dots, C_n are \mathcal{LX} -concepts, and which is closed under the concept constructors of \mathcal{L} . The set of \mathcal{LX} -axioms is the smallest set containing all axioms built from \mathcal{LX} -concepts using the axiom constructors of \mathcal{L} , as well as axioms of the form $\Gamma \Rightarrow \beta$ (conditional axioms), where β is a non-conditional \mathcal{LX} axiom and Γ is a set of such axioms. An \mathcal{LX} -ontology is a (possibly infinite) set \mathcal{K} of \mathcal{LX} -axioms.

For a conditional axiom $\alpha = (\Gamma \Rightarrow \beta)$, we call all elements $\gamma \in \Gamma$ conditions of α and β the target of α . All axioms that are not conditional, i.e., Γ is empty, we call *unit* axioms. Note that all conditions and the target can only be unit axioms; conditional axioms can not be nested.¹ We introduce a range of special notations to facilitate the discussion about \mathcal{LX} concepts and axioms.

Definition 2 (Adapted from (Hirschbrunn and Kazakov 2024)). *Let the expression ex be either a \mathcal{LX} -concept, a \mathcal{LX} -axiom, or a \mathcal{LX} -ontology. We denote by $\text{sub}(ex)$ (all) subconcepts of ex , i.e., substrings of the expression that are valid concepts. For \mathcal{LX} -concepts and \mathcal{LX} -axioms (that are not using \Rightarrow), we split $\text{sub}(ex)$ into $\text{sub}^+(ex)$ and $\text{sub}^-(ex)$ the set of concepts that occur positively, respectively negatively, in ex , $\text{sub}(ex) = \text{sub}^+(ex) \cup \text{sub}^-(ex)$.² We denote by $\text{vars}(ex) = \text{sub}(ex) \cap N_X$ the set of concept variables occurring in ex . We say that ex is ground if $\text{vars}(ex) = \emptyset$.*

A (concept variable) substitution is a partial mapping $\theta = [X_1/C_1, \dots, X_n/C_n]$ that assigns concepts C_i to concept variables X_i ($1 \leq i \leq n$). We denote by $\theta(ex)$ the result of applying the substitution to ex , defined in the usual way.

In the remainder of the paper, we differentiate between different versions of a DL as follows: If concepts, axioms, and ontologies include concept terms, conditional axioms, and concept variables, we call them *generic*. If this is not the case, we call these *classical*.

As described in Section 1, we adapt the existing second-order semantics for generic extensions (Hirschbrunn and Kazakov 2024) for usage with conditional axioms:

Definition 3 (Second-Order Semantics, adapted from (Hirschbrunn and Kazakov 2024)). *A (second-order) interpretation for a \mathcal{LX} is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty set called the domain of \mathcal{I} and $\cdot^{\mathcal{I}}$ is an interpretation function, which assigns to every $A \in N_C$ with arity*

¹If conditional axioms were allowed to be nested, we would be able to express all boolean combinations over axioms.

²For less expressive DLs occurring positively (negatively) usually simply corresponds to occurring on the right side of the axiom under an even (odd) number of nested negations or on the left side under an odd (even) number of nested negations, for more expressive DLs this can be more difficult, see e.g., (Simancik 2012) for *SRIOQ*.

$n = \text{ar}(A)$ a function $A^{\mathcal{I}} : (2^{\Delta^{\mathcal{I}}})^n \rightarrow 2^{\Delta^{\mathcal{I}}}$ and to every $r \in N_R$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. A valuation for \mathcal{I} (also called a variable assignment) is a mapping η that assigns to every variable $X \in N_X$ a subset $\eta(X) \subseteq \Delta^{\mathcal{I}}$.

The interpretation of \mathcal{LX} -concepts $C^{\mathcal{I}, \eta} \subseteq \Delta^{\mathcal{I}}$ is recursively defined by $X^{\mathcal{I}, \eta} = \eta(X)$ for $X \in N_X$, $A[C_1, \dots, C_n]^{\mathcal{I}, \eta} = A^{\mathcal{I}}(C_1^{\mathcal{I}, \eta}, \dots, C_n^{\mathcal{I}, \eta})$, and is extended to other \mathcal{LX} -concepts in the usual way. Satisfaction of unit axioms $\mathcal{I} \models_{\eta}^2 \beta$ under \mathcal{I} and η is determined from the interpretation of \mathcal{LX} -concepts in β in the standard way. For example, $\mathcal{I} \models_{\eta}^2 C \sqsubseteq D$ iff $C^{\mathcal{I}, \eta} \subseteq D^{\mathcal{I}, \eta}$. The interpretation of conditional axioms follows naturally, i.e., $\mathcal{I} \models_{\eta}^2 \Gamma \Rightarrow \beta$, iff $\exists \gamma \in \Gamma : \mathcal{I} \not\models_{\eta}^2 \gamma$ or $\mathcal{I} \models_{\eta}^2 \beta$. We write $\mathcal{I} \models^2 \alpha$ if $\mathcal{I} \models_{\eta}^2 \alpha$ for every valuation η . Finally, for an ontology \mathcal{K} , we write $\mathcal{I} \models^2 \mathcal{K}$ if $\mathcal{I} \models^2 \alpha$ for every $\alpha \in \mathcal{K}$, and we write $\mathcal{K} \models^2 \alpha$ if $\mathcal{I} \models^2 \mathcal{K}$ implies $\mathcal{I} \models^2 \alpha$.

We add a few remarks about this definition: First, for a classical ontology, every second-order model is a classical model and vice versa, as the second-order interpretation only differs from a classical interpretation in its treatment of atomic concepts as functions, which is not relevant if we only have atomic concepts with zero arity, as in the case of classical ontologies. Second, notice that for our conditional axioms, the same η is considered for the conditions, as for the target, i.e., the $\forall \eta$ quantification is outside the implication. This is important as we want conditions to restrict the choice of subsets of the domain that are considered for the variables in the target axiom. For example, all usages of X in $\{X \sqsubseteq \text{Pet}\} \Rightarrow \text{Keeper}[X] \equiv \exists \text{owns}.X \sqcap \exists \text{feeds}.X$, must be the same, and describe some kind of pet. Finally, we can easily reduce the entailment of a ground axiom $\Gamma \Rightarrow \beta$, i.e., $\mathcal{K} \models \Gamma \Rightarrow \beta$, to the unsatisfiability of \mathcal{K} extended with Γ and the conditional axiom $\{\beta\} \Rightarrow \top \sqsubseteq \perp$: Clearly, $\mathcal{K} \models \Gamma \Rightarrow \beta$ iff $\mathcal{K} \cup \Gamma \models \beta$ and if $\mathcal{K} \cup \{\{\beta\} \Rightarrow \top \sqsubseteq \perp\}$ is unsatisfiable, then $\mathcal{I} \models \beta$ holds for every model \mathcal{I} of \mathcal{K} , therefore $\mathcal{K} \models \Gamma \Rightarrow \beta$ iff $\mathcal{K} \cup \Gamma \cup \{\{\beta\} \Rightarrow \top \sqsubseteq \perp\}$ is unsatisfiable.

In the remainder of the paper, we sometimes use the following ordering on (second-order) interpretations: For two interpretations \mathcal{I}_1 and \mathcal{I}_2 over the same domain Δ ($\Delta^{\mathcal{I}_1} = \Delta^{\mathcal{I}_2} = \Delta$), we define: $\mathcal{I}_1 \preceq \mathcal{I}_2$ iff $\forall r \in N_R : r^{\mathcal{I}_1} = r^{\mathcal{I}_2}$ and $\forall A \in N_C, \forall M_1, \dots, M_n \subseteq \Delta : A(M_1, \dots, M_n)^{\mathcal{I}_1} \subseteq A(M_1, \dots, M_n)^{\mathcal{I}_2}$.

4 Conditional Axioms

We start our analysis by considering the new feature introduced in this paper, i.e., conditional axioms, on their own. That is, we consider the extension of classical DLs (only) with conditional axioms (not yet concept terms or concept variables). For example, we allow axioms such as $\{A \sqsubseteq B, A \sqsubseteq C\} \Rightarrow \exists r.A \sqsubseteq \top$. The classical satisfaction relation can easily be extended to these conditional axioms with classical concepts: $\mathcal{I} \models \Gamma \Rightarrow \beta$, iff $\exists \gamma \in \Gamma : \mathcal{I} \not\models \gamma$ or $\mathcal{I} \models \beta$.

Reasoning with a classical ontology with conditional axioms can be nondeterministically reduced to reasoning with negated axioms by choosing for each conditional axiom either the target axiom or the negation of some condition ax-

iom. Then the satisfiability of our ontology with conditions coincides with the satisfiability of (at least) one of these constructed ontologies.

Theorem 1. *There is a non-deterministic algorithm that reduces in polynomial time the satisfiability of a classical ontology with conditional axioms to the satisfiability of an ontology potentially including negated axioms.*

Proof. Let \mathcal{K} be a classical ontology with conditional axioms. Let \mathcal{K}' be obtained from \mathcal{K} by adding (1) all unit axioms to \mathcal{K}' and (2) for each (conditional) axiom $\{\gamma_1, \dots, \gamma_n\} \Rightarrow \beta \in \mathcal{K}$, adding non-deterministically either β or one of $\neg\gamma_i$ for some $1 \leq i \leq n$. If \mathcal{K} is satisfiable, then for some of these choices \mathcal{K}' is satisfiable.

Given $\mathcal{I} \models \mathcal{K}$, we can construct one such \mathcal{K}' by doing for each $\alpha \in \mathcal{K}$ the following: If α is a unit axiom, then $\alpha \in \mathcal{K}'$ otherwise $\alpha = \{\gamma_1, \dots, \gamma_n\} \Rightarrow \beta$, and as $\mathcal{I} \models \alpha$ either $\exists j (1 \leq j \leq n) : \mathcal{I} \not\models \gamma_j$ and we add $\neg\gamma_j$ to \mathcal{K}' or $\mathcal{I} \models \beta$, and we add β to \mathcal{K}' . Then $\mathcal{I} \models \mathcal{K}'$ and because \mathcal{K}' contains the non-conditional axioms of \mathcal{K} and for each conditional axiom, either one of the negated conditions or the target is in \mathcal{K}' , \mathcal{K}' fulfills the construction described above.

Conversely, if \mathcal{K}' is satisfiable then \mathcal{K} is satisfiable in the same interpretation: Take \mathcal{I} such that $\mathcal{I} \models \mathcal{K}'$, we show that $\mathcal{I} \models \mathcal{K}$. Take $\alpha \in \mathcal{K}$ either α is non-conditional and $\alpha \in \mathcal{K}'$ or $\alpha = \{\gamma_1, \dots, \gamma_n\} \Rightarrow \beta$ and either $\exists \gamma_j : \mathcal{I} \models \neg\gamma_j$ then $\mathcal{I} \models \alpha$ or $\mathcal{I} \models \beta$ and $\mathcal{I} \models \alpha$.

This gives us a non-deterministic polynomial-time reduction. \square

The reduction described in Theorem 1 can be used for DLs, which can express negation of the axioms appearing as conditions. For example, negations of concept inclusion axioms $C \sqsubseteq D$ can be expressed as $\{C(a), (\neg D)(a)\}$ with a a fresh individual. Of course, for a less expressive DL like \mathcal{EL} , this raises the complexity of reasoning, as effectively we are using \mathcal{ALC} reasoning.

5 Ground Ontologies

The approach described in the previous section allows us to remove conditional axioms from ontologies to get back to the classical case. With this result, only two features still make it difficult to consider a generic ontology under standard classical interpretations. The first are variables, the second are concept terms. We can leave variables aside for now by considering only ground generic ontologies. To deal with concept terms, a naive way to interpret them under classical interpretations is to simply consider them as new atomic concept names.³ Unfortunately, this has the side effect that we do not account for equivalent axioms anymore, i.e., using classical interpretations in this way for an ontology such as $\{C \equiv D\}$ we do not get $A[C] \equiv A[D]$ as a consequence because $A[C]$ and $A[D]$ are two independent atomic concepts. On the other hand, clearly, for second-order semantics, we get $A[C] \equiv A[D]$ as the function $A^{\mathcal{J}}$ applied to the same set $M = C^{\mathcal{I}} = D^{\mathcal{I}}$ twice, gives the same result. To still be able

³This was done in the existing work on generic extensions, leading to syntactic restrictions (Hirschbrunn and Kazakov 2024).

to reduce second-order entailment to classical entailment using this approach, we transform the given ontology using a closure that moves this treatment of equal concepts from the semantics to explicit (conditional) axioms in the ontology:

Definition 4 (Congruence Closure). *A congruence axiom is a (conditional) axiom of the form: $\bigwedge_{i=1}^n C_i \equiv D_i \Rightarrow A[C_1, \dots, C_n] \equiv A[D_1, \dots, D_n]$ where $n = \text{ar}(A)$ and C_i, D_i , \mathcal{LX} -concepts ($1 \leq i \leq n$). The congruence closure of a ground ontology \mathcal{K} is the extension of \mathcal{K} with all congruence axioms for which $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{K})$ and $A[D_1, \dots, D_n] \in \text{sub}(\mathcal{K})$.*

Clearly, all congruence axioms are tautologies under the second-order semantics:

Lemma 2. *Let α be a congruence axiom and \mathcal{I} a second-order interpretation. Then $\mathcal{I} \models^2 \alpha$.*

Proof. Let α be a congruence axiom (Definition 4), \mathcal{I} a second-order interpretation, and η a variable assignment. If $C_i^{\mathcal{I}, \eta} \neq D_i^{\mathcal{I}, \eta}$ for some i ($1 \leq i \leq n$), then, trivially, $\mathcal{I} \models_{\eta}^2 \alpha$. Otherwise $(A[C_1, \dots, C_n])^{\mathcal{I}, \eta} = A^{\mathcal{I}}(C_1^{\mathcal{I}, \eta}, \dots, C_n^{\mathcal{I}, \eta}) = A^{\mathcal{I}}(D_1^{\mathcal{I}, \eta}, \dots, D_n^{\mathcal{I}, \eta}) = A[D_1, \dots, D_n]^{\mathcal{I}, \eta}$, which, likewise, implies $\mathcal{I} \models_{\eta}^2 \alpha$. Since η was arbitrary, we proved $\mathcal{I} \models^2 \alpha$. \square

Lemma 3. *Let \mathcal{K} be a ground ontology and \mathcal{K}' the congruence closure of \mathcal{K} (see Definition 4). Then \mathcal{K} is satisfiable under second-order semantics iff \mathcal{K}' is (classically) satisfiable.*

Proof. (\Rightarrow) Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a second-order interpretation such that $\mathcal{I} \models^2 \mathcal{K}$. Define the classical interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ with $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and $A[C_1, \dots, C_n]^{\mathcal{J}} = A^{\mathcal{I}}(C_1^{\mathcal{I}}, \dots, C_n^{\mathcal{I}})$ for every $A \in N_C$, $n = \text{ar}(A)$ and C_i ground \mathcal{LX} -concepts ($1 \leq i \leq n$), and $r^{\mathcal{J}} = r^{\mathcal{I}}$ for every $r \in N_R$. Note that this definition implies that $D^{\mathcal{J}} = D^{\mathcal{I}}$ for every ground \mathcal{LX} -concept since the extension of interpretation under concept constructors is defined in \mathcal{I} and \mathcal{J} in the same way. Likewise, $\mathcal{I} \models^2 \alpha$ iff $\mathcal{J} \models \alpha$ for every ground \mathcal{LX} -axiom α . Hence, from $\mathcal{I} \models^2 \mathcal{K}$, we obtain $\mathcal{J} \models \mathcal{K}$. Further, by Lemma 2, $\mathcal{I} \models^2 \alpha$ for every congruence axiom $\alpha \in \mathcal{K}'$. Hence $\mathcal{J} \models \mathcal{K}'$.

(\Leftarrow) Let \mathcal{J} be a classical interpretation such that $\mathcal{J} \models \mathcal{K}'$. Define the second-order interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$, $A^{\mathcal{I}}(M_1, \dots, M_n) = A[C_1, \dots, C_n]^{\mathcal{J}}$ if $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{K})$ and $M_i = C_i^{\mathcal{J}}$ ($1 \leq i \leq n$), and $A^{\mathcal{I}}(M_1, \dots, M_n) = \emptyset$ in the remaining cases, and $r^{\mathcal{I}} = r^{\mathcal{J}}$ for every $r \in N_R$. Notice that the interpretation of $A^{\mathcal{J}}(M_1, \dots, M_n)$ is well-defined, i.e., it does not depend on the choice of the atom $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{K})$ such that $C_i^{\mathcal{J}} = M_i$ ($1 \leq i \leq n$). Indeed, for every other choice $A[D_1, \dots, D_n] \in \text{sub}(\mathcal{K})$ such that $D_i^{\mathcal{J}} = M_i$ ($1 \leq i \leq n$), by Definition 4, the congruence axiom belongs to \mathcal{K}' , and since $\mathcal{J} \models \mathcal{K}'$ and $C_i^{\mathcal{J}} = D_i^{\mathcal{J}}$ ($1 \leq i \leq n$), we obtain $A[C_1, \dots, C_n]^{\mathcal{J}} = A[D_1, \dots, D_n]^{\mathcal{J}}$. Since $A[C_1, \dots, C_n]^{\mathcal{J}} = A^{\mathcal{I}}(C_1^{\mathcal{I}}, \dots, C_n^{\mathcal{I}})$ for every $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{K})$, similarly like in the case (\Rightarrow),

it follows that $\mathcal{I} \models^2 \alpha$ iff $\mathcal{J} \models \alpha$ for every $\alpha \in \mathcal{K}$. Since $\mathcal{J} \models \mathcal{K}'$ and $\mathcal{K} \subseteq \mathcal{K}'$, it follows that $\mathcal{I} \models^2 \mathcal{K}$. \square

Note that \mathcal{K}' can be computed in polynomial time in the size of \mathcal{K} since the number of atoms $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{K})$ is linear in \mathcal{K} . Therefore, we get the following result.

Theorem 4. *Second-order satisfiability of ground ontologies with conditional axioms can be reduced in polynomial time to the satisfiability of ground ontologies with conditional axioms under classical semantics.*

Proof. Let \mathcal{K} be a ground ontology with conditional axioms, and \mathcal{K}' its congruence closure according to Definition 4. Note that \mathcal{K}' can be computed in polynomial time in the size of \mathcal{K} since the number of atoms $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{K})$ is linear in \mathcal{K} . The statement of the theorem now follows directly from Lemma 3. \square

6 Terminologies

Following the results regarding ground ontologies, in this section, we extend our results to the non-ground case. The goal of this section is to reduce the (second-order) satisfiability of generic non-ground ontologies to the (second-order) satisfiability of generic ground ontologies. With the results from the previous sections, this gives us the ability to reduce reasoning with generic ontologies to classical reasoning (with negated axioms).

Our approach works under the assumption that a given generic ontology consists of two parts: a *ground part* containing arbitrary ground axioms and a *terminological part* consisting of generic concept definitions. Our main result shows that, under certain semantic conditions, an arbitrary model of the ground part can be extended to a model of the terminological part by using a fixpoint operator reminiscent of defining the least fixpoint semantics for (cyclic) \mathcal{EL} terminologies (Baader 2003).

Definition 5 (Generic Terminology). *A (generic) concept definition is a conditional axiom α of the form $\Gamma \Rightarrow A[X_1, \dots, X_n] \equiv D$ (called a complete concept definition) or $\Gamma \Rightarrow A[X_1, \dots, X_n] \sqsubseteq D$ (called a partial concept definition), where $n = \text{ar}(A) \geq 1$, X_1, \dots, X_n are distinct variables and $\text{vars}(\Gamma) \cup \text{vars}(D) \subseteq \{X_1, \dots, X_n\}$. We say that α defines A (completely or partially, respectively), call A the defined concept name of α and D its description. A (generic) terminology is a set \mathcal{T} of concept definitions, such that a concept is either defined in one complete definition or in one or more partial definitions, but not both.*

For a given generic terminology \mathcal{T} , we denote completely defined concept names as $\Sigma_{\text{def}}^{\mathcal{T}}$, and partially defined concept names as $\Sigma_{\text{part}}^{\mathcal{T}}$. All other concept names occurring in the terminology are called *primitive concept names*, denoted $\Sigma_{\text{prim}}^{\mathcal{T}}$. It should be noted that we permit cyclic dependencies among the defined concept names. We do not consider an axiom as a proper concept definition if a variable occurs in the conditional axiom, but not as an argument of the defined concept name. In this case, the definition would be ambiguous; for

example, $A[X] \equiv X \sqcap \exists r.Y$ does not clearly define how $A[C]$ should be interpreted.

As said above, we want to take a model of the ground part of an ontology and extend it to a model of the whole ontology (including the non-ground but terminological part). This means that given a model of the non-ground part \mathcal{I} , we can only change the interpretation of parameterized concepts for arguments that do not occur in the ground part, e.g., if $A[C]$ is a concept in the ground ontology, we may not change the interpretation of $A^{\mathcal{I}}(M)$ for the argument $M = C^{\mathcal{I}}$ in order to ensure that our resulting interpretation still is a model. What we can change is the interpretation for “unknown” M s. We formalize these allowed changes in the following definition.

Definition 6 (Terminological Expansions). *Let \mathcal{G} be a ground generic ontology, \mathcal{T} a generic terminology, and \mathcal{I} a model of \mathcal{G} , i.e., $\mathcal{I} \models \mathcal{G}$. We call a set $M \subseteq \Delta^{\mathcal{I}}$ known if there is a $C \in \text{sub}(\mathcal{G})$ such that $C^{\mathcal{I}} = M$, otherwise it is unknown. A subset of $\mathcal{P}(\Delta^{\mathcal{I}})$ is unknown if at least one member is unknown. Then a terminological expansion of \mathcal{I} is an interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$, such that $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$, $\forall r \in N_R : r^{\mathcal{J}} = r^{\mathcal{I}}$, $\forall C \in \text{sub}(\mathcal{G}) : C^{\mathcal{J}} = C^{\mathcal{I}}$, for $A \in \Sigma_{\text{part}}^{\mathcal{T}}$ and $M_1, \dots, M_n \subseteq \Delta^{\mathcal{I}}$ unknown, $A(M_1, \dots, M_n)^{\mathcal{J}} = \emptyset$, and $\forall B \in \Sigma_{\text{prim}}^{\mathcal{T}} : B^{\mathcal{J}} = B^{\mathcal{I}}$. By $\text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ we denote the set of all terminological expansions of \mathcal{I} . We omit \mathcal{G} and \mathcal{T} if they are irrelevant or clear from the context.*

Note that, for the concepts appearing in \mathcal{T} and \mathcal{G} , a $\mathcal{J} \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ differs from \mathcal{I} only in the interpretation of defined concept names when those are applied to unknown arguments, i.e., to subsets of the domain that are not “represented” by any concept that occurs in \mathcal{G} . This makes sure that for all concepts in \mathcal{G} , \mathcal{J} and \mathcal{I} coincide, i.e., $\mathcal{J} \models^2 \mathcal{G}$.

We choose to interpret concept terms using partially defined concept names applied to unknown arguments as the empty set in the expansions. The advantage of this is that we immediately know that their definition is entailed by every $\mathcal{J} \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$.

Definition 7. *An $L\mathcal{X}$ -ontology \mathcal{K} is said to be admissible if $\mathcal{K} = \mathcal{G} \cup \mathcal{T}$, where:*

- A1. \mathcal{G} is a ground ontology,
- A2. \mathcal{T} is a (generic) terminology,
- A3. If A is defined by some $\alpha \in \mathcal{T}$ with arguments X_1, \dots, X_n , then for every substitution θ such that $\theta(A[X_1, \dots, X_n]) \in \text{sub}(\mathcal{G})$ it holds that $\mathcal{G} \models^2 \theta(\alpha)$,
- A4. If D is the description of some completely defined concept name in \mathcal{T} , \mathcal{I} a model of \mathcal{G} and $\mathcal{J}_1, \mathcal{J}_2 \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ such that $\mathcal{J}_1 \preceq \mathcal{J}_2$, then $D^{\mathcal{J}_1, \eta} \subseteq D^{\mathcal{J}_2, \eta}$ for every valuation η , i.e., D behaves monotonically.

The notion of an *admissible ontology* ensures that an extension of a model of the ground part to a model of the terminological part is always possible. Condition 3 prevents a clash of the knowledge of the ground and the terminological parts, e.g., having an axiom $\top \sqsubseteq A[B]$ in \mathcal{G} and an axiom $A[X] \equiv \perp$ in \mathcal{T} violates this condition. Furthermore, Condition 3 ensures that for known arguments, definitions in \mathcal{T} are entailed by every $\mathcal{J} \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$. This means we do

only need to choose a $\mathcal{J} \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ that also entails complete definitions for unknown arguments to get a model of \mathcal{K} . To find this \mathcal{J} , we use an approach that (starting from \mathcal{I}) changes the interpretation of completely defined concept names step-by-step to get closer to their definition, until a fixpoint is reached.

For such a fixpoint to exist, we need to make sure that the interpretation only increases from step to step. Because we allow defined concept names in the descriptions in \mathcal{T} , this can only be ensured if descriptions are always upward monotonic in all concept terms using fully defined concept names. For example, if we had $A[X] \equiv \neg B[X]$, this would not hold, as if we assume that $B[X]$ increases with every step of our expansion, then $A[X]$ would at the same time decrease. Indeed, if this monotonicity were not required, we would be able to express General Concept Inclusions (GCIs) in our terminology. The reason for this is similar to absorption (Horrocks and Tobies 2000): We can express a GCI $C[X] \sqsubseteq D[X]$ as $\top \equiv \neg C[X] \sqcup D[X]$ and, because this is not allowed as a terminological axiom (as \top is not a parameterized concept), we use $A[X] \equiv \neg A[X] \sqcap \neg B[X]$ to be able to use $B[X] \equiv \neg C[X] \sqcup D[X]$ instead of \top . To prevent such cases, we use Condition 4 in Definition 7.

Definition 8 (One Step Expansion). *Let $\mathcal{K} = \mathcal{G} \cup \mathcal{T}$ be an admissible ontology according to Definition 7 and \mathcal{I} a model of \mathcal{G} . The one-step expansion is a function $1\text{Exp}_{\mathcal{K}, \mathcal{I}} : \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I}) \rightarrow \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ such that $1\text{Exp}_{\mathcal{K}, \mathcal{I}}(\mathcal{J})$ is the interpretation $\mathcal{J}' \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ defined by changing the interpretation of completely defined concept names in the following way: Let $A \in \Sigma_{\text{def}}^{\mathcal{T}} M_1, \dots, M_n \subseteq \Delta^{\mathcal{I}}$ unknown, if A is defined by $\Gamma \Rightarrow A[X_1, \dots, X_n] \equiv D \in \mathcal{T}$ then for $\eta = \{X_1/M_1, \dots, X_n/M_n\} : A^{\mathcal{J}'}(M_1, \dots, M_n) = D^{\mathcal{J}, \eta}$.*

Intuitively, the one-step expansion $1\text{Exp}(\mathcal{J})$ is the result of updating the interpretation of completely defined concept names (when applied to unknown arguments) by using their description. Note that the conditions of Definition 5 ensure that one-step expansion of \mathcal{J} is well-defined. In particular, the definition is unambiguous because every concept name is completely defined in \mathcal{T} at most once, and all concept variables appearing in this definition must be parameters of this concept name. In this procedure, we do not take the conditions Γ into consideration, because if we make sure that in the extended model the definition $A[X_1, \dots, X_n] \equiv D$ holds for every choice of X_i , then it also holds in the cases where Γ is also entailed. Disregarding Γ can also not lead to contradictions: A contradiction with another axiom in \mathcal{T} is not possible, because every concept name is only completely defined once (regardless of whether conditions are present or not); And a contradiction with \mathcal{G} is not possible because we only change the interpretation for unknown arguments.

We are now ready to show the final result of this section. We use here that the one-step expansion we defined is a monotonic function on the set of terminological expansions, giving us the guaranteed existence of a fixpoint. This fixpoint is our new model of the whole ontology \mathcal{K} .

Theorem 5. *Let $\mathcal{K} = \mathcal{G} \cup \mathcal{T}$ be an admissible ontology and \mathcal{G} second-order satisfiable, then \mathcal{K} is second-order satisfiable.*

Proof. Given a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of \mathcal{G} , we show that a fixpoint of $1\text{Exp}_{\mathcal{K}, \mathcal{I}}$ exists such that it is a model of \mathcal{K} .

We start by showing the monotonicity of the one-step expansion, i.e., for $\mathcal{J}_1, \mathcal{J}_2 \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$, $\mathcal{J}_1 \preceq \mathcal{J}_2$ implies $1\text{Exp}(\mathcal{J}_1) \preceq 1\text{Exp}(\mathcal{J}_2)$. By Definition 8, we need to show $A^{1\text{Exp}(\mathcal{J}_1)}(M_1, \dots, M_n) \subseteq A^{1\text{Exp}(\mathcal{J}_2)}(M_1, \dots, M_n)$ for all unknown $M_1, \dots, M_n \subseteq \Delta^{\mathcal{I}}$. Let $\eta = \{X_i \rightarrow M_i\}$ then by Definition 7 Case A4, $D^{\mathcal{J}_1, \eta} \subseteq D^{\mathcal{J}_2, \eta}$ and $A(M_1, \dots, M_n)^{1\text{Exp}(\mathcal{J}_1)} = D^{\mathcal{J}_1, \eta} \subseteq D^{\mathcal{J}_2, \eta} = A^{1\text{Exp}(\mathcal{J}_2)}(M_1, \dots, M_n)$.

One can easily see that \preceq is a complete lattice on $\text{Tx}(\mathcal{I})$. Then by Tarski's Fixpoint Theorem (Lloyd 1987), $1\text{Exp}_{\mathcal{K}, \mathcal{I}}$ has a fixpoint, let \mathcal{J} denote such a fixpoint. As $\mathcal{J} \in \text{Tx}_{\mathcal{G}, \mathcal{T}}(\mathcal{I})$ we know $\mathcal{J} \models^2 \mathcal{G}$.

We now show that $\mathcal{J} \models^2 \mathcal{T}$. Take $\Gamma \Rightarrow \beta \in \mathcal{T}$ and some η , we show that $\mathcal{J} \models_{\eta}^2 \Gamma \Rightarrow \beta$. Assume that β defines A for arguments X_1, \dots, X_n . Case 1: If we have $A[C_1, \dots, C_n] \in \text{sub}(\mathcal{G})$ and $\eta(X_i) = C_i^{\mathcal{J}} (1 \leq i \leq n)$, then by Definition 7 Case A3, $\mathcal{G} \models^2 [X_1/C_1, \dots, X_n/C_n](\Gamma \Rightarrow \beta)$ and as $\mathcal{J} \models^2 \mathcal{G}$, we get $\mathcal{J} \models_{\eta}^2 \Gamma \Rightarrow \beta$. Case 2: Assume that $\mathcal{J} \not\models_{\eta}^2 \Gamma$, then trivially $\mathcal{J} \models_{\eta}^2 \Gamma \Rightarrow \beta$, therefore in the remainder we assume that $\mathcal{J} \models_{\eta}^2 \Gamma$ and proof $\mathcal{J} \models_{\eta}^2 \beta$. Case 3: Assume that $\beta = A[X_1, \dots, X_n] \sqsubseteq D$, then by Definition 6, $A[X_1, \dots, X_n]^{\mathcal{J}, \eta} = \emptyset$ and therefore $\mathcal{J} \models^2 \beta$. Case 4: Assume that $\beta = A[X_1, \dots, X_n] \equiv D$, then because \mathcal{J} is a fixpoint of the one-step expansion, we know that applying the one-step expansion to \mathcal{J} does not change the interpretation of β . Then we know that for the interpretation of $A[X_1, \dots, X_n]$ this means that $A[X_1, \dots, X_n]^{\mathcal{J}, \eta} = D^{\mathcal{J}, \eta}$ and $\mathcal{J} \models_{\eta}^2 \beta$.

Therefore, we have shown that there is a model \mathcal{J} such that $\mathcal{J} \models \mathcal{K}$. \square

7 Ensuring Admissibility

In the previous section, we have shown that the satisfiability of a generic ontology with an arbitrary ground part and a terminological non-ground part can be reduced to the satisfiability of the ground part only. The requirement for this result is, that the given ontology is *admissible* (Definition 7), i.e., fulfills certain restrictions: First, for a defined concept name, the definition must already be entailed for known arguments by the ground part (A3), second, the descriptions of complete concept definitions need to be (upward) monotonic in the contained concept terms using completely defined concept names, i.e., increase if the interpretation of subterms increases (A4). These are both semantic restrictions; in this section, we discuss how these restrictions can be achieved in practice, i.e., turned into the following syntactic restrictions.

Definition 9. *An $L\mathcal{X}$ -ontology \mathcal{K} is said to be practically admissible if $\mathcal{K} = \mathcal{G} \cup \mathcal{T}$, where:*

- B1. \mathcal{G} is a ground ontology,
- B2. \mathcal{T} is a (generic) terminology,
- B3. If D is a description in \mathcal{T} , and $A[C_1, \dots, C_n] \in \text{sub}(D)$ then C_1, \dots, C_n are variables or 0-ary atomic concepts.
- B4. For each $A[X_1, \dots, X_n] \equiv D \in \mathcal{T}$: If $B[C_1, \dots, C_m] \in \text{sub}(D)$ such that $B[X_1, \dots, X_m] \equiv D' \in \mathcal{T}$, then $B[C_1, \dots, C_m] \in \text{sub}^+(D)$.

Definition 10 (Ground Expansion). *Given a generic ontology \mathcal{K} , consisting of a ground part \mathcal{G} and a generic terminology \mathcal{T} , i.e. $\mathcal{K} = \mathcal{G} \cup \mathcal{T}$, we define the ground expansion $\text{Exp}(\mathcal{G})$ as the set of axioms achieved in the following way: All axioms in \mathcal{G} are in $\text{Exp}(\mathcal{G})$. Then we repeatedly check if for $A[C_1, \dots, C_n] \in \text{sub}(\text{Exp}(\mathcal{G}))$ such that A is defined by α in \mathcal{T} , we have $\theta(\alpha) \in \text{Exp}(\mathcal{G})$ for $\theta = \{X_1/C_1, \dots, X_n/C_n\}$. If this is not the case, we add $\theta(\alpha)$ to $\text{Exp}(\mathcal{G})$. We repeat this until no new axioms are added to $\text{Exp}(\mathcal{G})$.*

To get from condition A3 to B3, we use the ground expansion procedure. This procedure does not terminate in every case. To achieve termination, it is important that there do not occur complex concepts or concept terms as arguments (of concept terms) in \mathcal{T} . If this were the case and we additionally have a cyclic dependency (which we allow), we could get non-termination: Consider, e.g. $A[X] \equiv A[B[X]] \in \mathcal{T}$, then any replacement of X with a concept C from $A[C] \in \text{sub}(\text{Exp}(\mathcal{G}))$ results in a new concept $A[B[C]] \in \text{sub}(\text{Exp}(\mathcal{G}))$, which again results in a new concept $A[B[B[C]]] \in \text{sub}(\text{Exp}(\mathcal{G}))$ and so on. This would result in an infinite $\text{Exp}(\mathcal{G})$. If we do not have complex concepts or concept terms as arguments, calculating the expansion of \mathcal{G} takes at most exponential time (even in the presence of cycles). This is because, in the worst case, we have to ground every terminological axiom with every set of concepts occurring as arguments of concept terms in \mathcal{K} .

Clearly, if the original \mathcal{K} is satisfiable, then the resulting $\text{Exp}(\mathcal{G})$ is also satisfiable, as we only add instances of axioms in \mathcal{T} . Furthermore, it is easy to see that now $\mathcal{K}' = \text{Exp}(\mathcal{G}) \cup \mathcal{T}$ fulfills Condition A3. Therefore, instead of checking A3, we can instead check B3 and apply the procedure from Definition 10.

Similarly, it is possible to turn the semantic condition A4 into the syntactic condition B4. A4 requires the descriptions of complete concept definitions to behave monotonically if the interpretation of other completely defined concept names (applied to unknown arguments) is extended. (The interpretation of other concepts is already fixed in $\text{Tx}(\mathcal{I})$, compare Definition 6 and Definition 7). It is well known that positive polarity of subterms results in the concept being upward monotonic in the subterm (see e.g., (Blackburn, de Rijke, and Venema 2001)). Using a suitable definition of positive polarity for *SRIQ* (e.g., (Simancik 2012)) this can be shown in general using induction on the structure of the concept description. In our case, a simple syntactic condition (B4) that is sufficient to achieve monotonicity, is to require that concept terms using completely defined concept names may only occur positively in a complete concept description.

Therefore, instead of checking the admissibility of an \mathcal{LX} -ontology according to the semantic Definition 7, we can instead use the syntactic practical admissibility in Definition 9.

Corollary 6. *If an ontology $\mathcal{K} = \mathcal{G} \cup \mathcal{T}$ is practically admissible according to Definition 9, then the ontology $\mathcal{K}' = \text{Exp}(\mathcal{G}) \cup \mathcal{T}$ is admissible according to Definition 7 and \mathcal{K} is second-order satisfiable iff \mathcal{K}' is second-order satisfiable.*

Taking these remarks together with our earlier results, we obtain the following reduction as a consequence of Theo-

rems 1, 4, and 5:

Corollary 7. *Given a generic ontology \mathcal{K} satisfying Definition 9, the satisfiability of \mathcal{K} under second-order semantics can be reduced in exponential time to the satisfiability of classical ontologies with negated axioms.*

8 Discussion and Conclusion

Generic description logics were introduced to efficiently handle collections of similar axioms in ontologies, offering advantages akin to those of generic classes in programming: A concept name’s definition can be applied in various contexts, minimizing the necessity for duplicating and altering intricate concept structures. This method supports modular ontology construction and aids in preventing mistakes that may occur during axiom refactoring. Unfortunately, existing generic extensions (Hirschbrunn and Kazakov 2024) were limited to fragments of the extension of \mathcal{EL} .

In this paper, we lift that restriction and demonstrate the decidability of generic extensions of expressive DLs up to *SRIOQ*. We achieve this by requiring that axioms with variables are only used to define concept names, while they can be used freely when ground. This is a reasonable restriction, as this captures the initial idea of generic concepts, namely, being a way to combine the definition of many similar concepts into one place. It also corresponds to the historic development of DLs, which also started with terminologies.

We also introduce a new feature of generic extensions, namely, conditional axioms. These allow us to formulate conditions under which an axiom should hold, while in interpretations where these conditions do not hold, the axiom can be ignored. Conditional axioms are a natural addition to generic DLs, akin to bounds in generic programming. They can be used as a check on variable replacements in concept terms, allowing to select one (or more) of potentially many partial definitions given for a concept name in an ontology. Furthermore, conditional axioms are also an advantage for complete definitions, for example, we can formulate that the definition of $\text{Keeper}[X]$ “makes sense” only when X describes some set of pets, i.e., $\{X \sqsubseteq \text{Pet}\} \Rightarrow \text{Keeper}[X] \equiv \exists \text{owns}.X \sqcap \exists \text{feeds}.X$. This prevents modeling errors, where $\text{Keeper}[\cdot]$ is used with some wrong argument, such as $\text{Keeper}[\text{Car}]$.

Planned future work includes implementing the approach described here, studying how existing ontologies might benefit from generic extensions, specifically, determining the extent to which inherent complexity can be reduced, and developing a tool for automatically translating existing ontologies to use the generic extension.

In summary, the findings in this paper demonstrate that it is possible to get generic extensions of expressive description logics that are still decidable, provided certain reasonable restrictions are applied. Additionally, the introduction of conditional axioms allows the use of generic concepts in a more targeted way by restricting the replacement of parameters. This is a valuable addition to the area of generic description logics, as well as to the broader research area that deals with exploiting syntactic regularities in ontologies.

References

- Baader, F. 2003. Terminological cycles in a description logic with existential restrictions. In *IJCAI*, volume 3, 325–330.
- Blackburn, P.; de Rijke, M.; and Venema, Y. 2001. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press. ISBN 978-1-10705088-4.
- Borgida, A.; Horkoff, J.; Mylopoulos, J.; and Rosati, R. 2012. Experiences in Mapping the Business Intelligence Model to Description Logics, and the Case for Parametric Concepts. In Kazakov, Y.; Lembo, D.; and Wolter, F., eds., *Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*, volume 846 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Gangemi, A.; and Presutti, V. 2009. Ontology Design Patterns. In Staab, S.; and Studer, R., eds., *Handbook on Ontologies*, 221–243. Springer Berlin Heidelberg. ISBN 978-3-540-70999-2 978-3-540-92673-3.
- Garcia, R.; Jarvi, J.; Lumsdaine, A.; Siek, J. G.; and Willcock, J. 2003. A comparative study of language support for generic programming. In *Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 115–134.
- He, Y.; Zheng, J.; and Lin, Y. 2015. Ontorat: Automatic generation and editing of ontology terms. In Couto, F. M.; and Hastings, J., eds., *Proceedings of the International Conference on Biomedical Ontology, ICBO 2015, Lisbon, Portugal, July 27-30, 2015*, volume 1515 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Hirschbrunn, J.; and Kazakov, Y. 2024. Extending Description Logics with Generic Concepts – the Tale of Two Semantics. In Marquis, P.; Ortiz, M.; and Pagnucco, M., eds., *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024*.
- Horrocks, I.; and Tobies, S. 2000. Reasoning with Axioms: Theory and Practice. *CoRR*, cs.LO/0005012.
- Kindermann, C.; George, A.-M.; Parsia, B.; and Sattler, U. 2024. Minimal Macro-Based Rewritings of Formal Languages: Theory and Applications in Ontology Engineering (and Beyond). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10581–10588. Issue: 9.
- Kindermann, C.; Lupp, D. P.; Sattler, U.; and Thorstensen, E. 2018. Generating Ontologies from Templates: A Rule-Based Approach for Capturing Regularity. In Ortiz, M.; and Schneider, T., eds., *Proceedings of the 31st International Workshop on Description Logics co-located with 16th International Conference on Principles of Knowledge Representation and Reasoning (KR 2018), Tempe, Arizona, US, October 27th - to - 29th, 2018*, volume 2211 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Kindermann, C.; Parsia, B.; and Sattler, U. 2019. Comparing Approaches for Capturing Repetitive Structures in Ontology Design Patterns. In Janowicz, K.; Krisnadhi, A. A.; Poveda-Villalón, M.; Hammar, K.; and Shimizu, C., eds., *Proceedings of the 10th Workshop on Ontology Design and Patterns (WOP 2019) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 27, 2019*, volume 2459 of *CEUR Workshop Proceedings*, 17–31. CEUR-WS.org.
- Klarman, S.; and Gutiérrez-Basulto, V. 2010. ALC_{ALC} : A Context Description Logic. In Janhunen, T.; and Niemelä, I., eds., *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings*, volume 6341 of *Lecture Notes in Computer Science*, 208–220. Springer.
- Krieg-Brückner, B.; Mossakowski, T.; and Neuhaus, F. 2019. Generic Ontology Design Patterns at Work. In Barton, A.; Seppälä, S.; and Porello, D., eds., *Proceedings of the Joint Ontology Workshops 2019 Episode V: The Styrian Autumn of Ontology, Graz, Austria, September 23-25, 2019*, volume 2518 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2008. Description Logic Rules. In Ghallab, M.; Spyropoulos, C. D.; Fakotakis, N.; and Avouris, N. M., eds., *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008. Proceedings*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, 80–84. IOS Press.
- Lawrynowicz, A.; Potoniec, J.; Robaczyk, M.; and Tudorache, T. 2018. Discovery of emerging design patterns in ontologies using tree mining. *Semantic Web*, 9(4): 517–544.
- Lloyd, J. W. 1987. *Foundations of logic programming*. Berlin ; New York : Springer-Verlag. ISBN 978-0-387-18199-8.
- Mikroyannidi, E.; Iannone, L.; Stevens, R.; and Rector, A. 2011. Inspecting Regularities in Ontology Design Using Clustering. In Aroyo, L.; Welty, C.; Alani, H.; Taylor, J.; Bernstein, A.; Kagal, L.; Noy, N.; and Blomqvist, E., eds., *The Semantic Web – ISWC 2011*, volume 7031, 438–453. Springer Berlin Heidelberg. ISBN 978-3-642-25072-9 978-3-642-25073-6. Series Title: Lecture Notes in Computer Science.
- Mikroyannidi, E.; Manaf, N. A. A.; Iannone, L.; and Stevens, R. 2012. Analysing Syntactic Regularities in Ontologies. In *OWLED*, volume 849.
- Simancik, F. 2012. Elimination of Complex RIAs without Automata. In Kazakov, Y.; Lembo, D.; and Wolter, F., eds., *Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*, volume 846 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Skjæveland, M. G.; Forssell, H.; Klüwer, J. W.; Lupp, D. P.; Thorstensen, E.; and Waaler, A. 2017. Reasonable Ontology Templates: APIs for OWL. In Nikitina, N.; Song, D.; Fokoue, A.; and Haase, P., eds., *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, volume 1963 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Skjæveland, M. G.; Lupp, D. P.; Karlsen, L. H.; and Forssell, H. 2018. Practical Ontology Pattern Instantiation, Discovery, and Maintenance with Reasonable Ontology Tem-

plates. In Vrandečić, D.; Bontcheva, K.; Suárez-Figueroa, M. C.; Presutti, V.; Celino, I.; Sabou, M.; Kaffee, L.; and Simperl, E., eds., *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, 477–494. Springer.