Lexicographical Inference over Inconsistent DL-based Ontologies

Jianfeng Du[1,2], Guilin Qi[3], and Yi-Dong Shen[1]

[1] State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences
[2] Graduate University of the Chinese Academy of Sciences
Beijing 100080, China
jfdu,ydshen@ios.ac.cn
[3] AIFB, Universität Karlsruhe, D-76128 Karlsruhe, Germany
gqi@aifb.uni-karlsruhe.de

Abstract. Logical inconsistency may often occur throughout the development stage of a DL-based ontology. We apply the *lexicographic inference* to reason over inconsistent DL-based ontologies without repairing them first. We address the problem of checking consequences in a $\mathcal{SHIQ}$ ontology that are classically inferred from every consistent (or coherent) subontology having the highest *lexicographic precedence*. We propose a method for compiling a $\mathcal{SHIQ}$ ontology to a propositional program so that the problem can be solved in polynomial calls to a SAT solver. We prove that this time complexity is worst-case optimal in data complexity. In order to make the method more scalable, we also present partition-based techniques to optimize the calling of SAT solvers.

1 Introduction

Ontologies play a core role for the success of the Semantic Web (SW) as they provide shared vocabularies for different domains. The Web Ontology Language (OWL) [25] is a standard language for modeling ontologies in the SW, which is based on Description Logics (DLs) [1]. The quality of ontologies is highly important for the SW technology. However, in practice it is difficult to construct an error free or logically consistent DL-based ontology. Logical inconsistency may often occur in different scenarios, such as ontology modeling, evolution, migration and merging [30, 11]. For example, if ontologies such as SUMO and CYC are directly merged into a single ontology, there will be misalignments of concepts that introduce logical inconsistency [29].

Given an inconsistent ontology, one may want to repair it so as to apply standard reasoners to access its (implicit) information. To fulfill this requirement, some methods (e.g. [30, 29, 24, 15, 6]) emerge. They repair inconsistent ontologies through debugging or diagnosing. Nevertheless, as pointed out by Haase *et al.* [11], in some cases consistency cannot be guaranteed at all and inconsistency cannot be repaired, still one wants to reason over ontologies in order to support information access and integration of new information. Hence, some other

methods (e.g. [13, 20, 19]) emerge to fulfill the latter requirement. They tolerate inconsistency and apply non-standard reasoning methods to obtain meaningful answers from an inconsistent ontology. Our method given in this paper belongs to the latter family of methods.

The notion of ordering plays a crucial role in handling inconsistency as it gives clues to tell which information is more important and should be kept. The well-known *lexicographic ordering* is defined for *stratified knowledge bases* in propositional logic [3], where a knowledge base, viewed as a set of formulas, is divided into a set of strata with priorities. A subbase is lexicographically preferable to another one if it contains more formulas in strata with higher priorities. A *lex-maximal consistent subbase* is defined as a subbase that has the highest lexicographic precedence. To reason with inconsistency, the *lexicographic inference* based on such ordering checks consequences that are classically inferred from every lex-maximal consistent subbase. Since many advantages of lexicographic inference have been shown in the literature [3, 4], such as the flexibility for utilizing priority information (e.g., priorities of different sources in the situation of ontology merging) and the conformity to the minimal-change point of view, we apply lexicographic inference to DLs.

The major challenge in applying lexicographic inference to DLs lies on the practicality of the computational aspect, because both reasoning in expressive DLs and lexicographic inference in propositional logic are already computationally hard. To take this challenge, we develop a method that is expected to work well on $\mathcal{SHIQ}$ [12] ontologies with simple terminologies and large ABoxes. The method checks *lex-consistent consequences* (resp. *lex-coherent consequences*) of a $\mathcal{SHIQ}$ ontology that are classically inferred from every *lex-maximal consistent* (resp. *coherent*[4]) *subontology*. Basically, the method first compiles the input $\mathcal{SHIQ}$ ontology into a propositional program, then performs the checking over the propositional program by polynomial calls to a SAT solver. We prove that this time complexity is worst-case optimal in *data complexity*, i.e. the complexity measured in the size of the ABox only. The compilation is based on an extension of the KAON2 transformation [14, 21] in which *decision atoms* are embedded, where decision atoms are new nullary atoms one-to-one corresponding to axioms in the original ontology. In order to make the method more scalable, we further adapt the partitioning technique in [6] to decompose the compiled propositional program, and develop a novel algorithm for using the partitioning results to check lex-consistent consequences. For the problem of checking a lex-coherent consequence, we first reduce it to the problem of checking a lex-consistent consequence, then solve it in the same way.

By now we have not fully tested the proposed method but a complete implementation is under way. Some relevant experimental results were reported in [6]. The method given in [6][5] similarly applies SAT solvers to compute consistent

[4] An ontology is called coherent if all atomic concept in it are satisfiable.

[5] The method employs the original KAON2 transformation and has a restriction that the terminology must be fixed and consistent. Such restriction is removed in the current work.

subontologies with certain maximality, such as those ones having the maximum number of ABox axioms[6]. It was shown [6] that for a $\mathcal{SHIQ}$ ontology, even though the ABox is large (i.e. has over tens of thousands of axioms), as long as the KAON2 transformation can reduce the terminology to over hundreds of DATALOG$^{\vee}$ [8] rules within which only a few are disjunctive or with equality, the subsequent partitioning step yields small propositional subprograms that can be efficiently handled by SAT solvers. Based on the fact that the extended KAON2 transformation spends almost all the time on the terminology (it directly translates atomic ABox axioms to ground clauses), the effectiveness of the partitioning step in decomposing a large propositional program into much smaller subprograms, as well as the efficiency of current powerful SAT solvers, we expect that the proposed method works well on $\mathcal{SHIQ}$ ontologies with simple terminologies and large ABoxes.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 gives some background on $\mathcal{SHIQ}$ and lexicographic inference. Section 4 formally defines lex-consistent and lex-coherent consequences for lexicographic inference in DLs. Section 5 presents our method for checking a lex-consistent (or lex-coherent) consequence of a stratified $\mathcal{SHIQ}$ ontology. Section 6 concludes.

2 Related Work

There exist some computational methods for lexicographic inference in DLs. In the work of Meyer *et al.* [20], the lexicographic inference and its refined version are respectively applied to $\mathcal{ALC}$ and its extension with *cardinality restrictions on concepts*. These inferences are computed through a *disjunctive DL knowledge base* (DKB for short) compiled from the original ontology. A *lex-consistent consequence* of the original ontology amounts to a consequence of the compiled DKB that is classically inferred from all disjuncts of the compiled DKB, where each disjunct is a DL-based ontology. In the work of Qi *et al.* [27], two other refined versions of lexicographic inference are proposed. The corresponding computational methods are also DKB-based. It should be noted that the DKB-based methods have a very high computational complexity. First, the compilation of a DKB needs up to exponential number of DL satisfiability tests wrt the number of axioms in the original ontology. Note that a satisfiability test in $\mathcal{SHIQ}$ is already NP-complete in data complexity [14]. Second, the checking of a consequence of a DKB is performed over all its disjuncts. Since the number of disjuncts can be exponential in the the number of axioms in the original ontology, the checking phase may need another exponential number of DL satisfiability tests. In contrast, our proposed method performs polynomial number of propositional satisfiability tests wrt the number of axioms in the original ontology in both

[6] They are actually lex-maximal consistent subontologies of a stratified ontology $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2)$, where $\mathcal{O}_1$, consisting of all terminological axioms, is the stratum with higher priority; and $\mathcal{O}_2$, consisting of all ABox axioms, is the stratum with lower priority.

the compiling phase and the checking phase. Each such satisfiability test is also NP-complete in data complexity and can further be optimized by our proposed partition-based techniques.

There exist other methods for reasoning over inconsistent DL-based ontologies [23, 13, 28, 18, 19]. As ours, most of them first specify the preferred consistent subontologies, then check consequences classically inferred from those subontologies. The method proposed in [13] first selects a consistent subontology based on a selection function, which is defined on the syntactic or semantic relevance, then reasons over the selected subontology. Such selected subontology is not always maximal, so the inference is less satisfactory from the minimal-change point of view. The methods given in [28] respectively extend *possibilistic* and *linear order* inferences in DLs by exploiting uncertainty degrees on DL axioms. Each extended inference selects a consistent subontology that keeps more DL axioms than the original one does, but the selected subontology is still often not maximal, so these extended inferences are also less satisfactory from the minimal-change point of view. The method given in [18] essentially checks consequences that are classically inferred from every maximal consistent subontology. It does not consider priority information on DL axioms and has a restriction that the terminology must be fixed and consistent. The reasoning methods proposed in [23] and [19] adopt a different idea. To tolerate inconsistency, they weaken an interpretation from two truth values to four truth values. Thus they result in a completely different reasoning mechanism for DL-based ontologies.

3 Preliminaries

3.1 $\mathcal{SHIQ}$ Syntax and Semantics

The $\mathcal{SHIQ}$ description logic [12] is highly related to OWL DL [25]. It semantically equals OWL DL without nominals and datatype specifications but with qualified number restrictions.

Given a set of role names N_R, a role is either some $R \in N_R$ or an inverse role R^- for $R \in N_R$. An *RBox* $\mathcal{O}_\mathcal{R}$ is a finite set of *transitivity axioms* $Trans(R)$ and *role inclusion axioms* $R \sqsubseteq S$, for R and S roles. For $R \in N_R$, we set $Inv(R) = R^-$ and $Inv(R^-) = R$, and assume that $R \sqsubseteq S \in \mathcal{O}_\mathcal{R}$ implies $Inv(R) \sqsubseteq Inv(S) \in \mathcal{O}_\mathcal{R}$ and $Trans(R) \in \mathcal{O}_\mathcal{R}$ implies $Trans(Inv(R)) \in \mathcal{O}_\mathcal{R}$. A role R is called *transitive* if $Trans(R) \in \mathcal{O}_\mathcal{R}$; *simple* if it has not any transitive subrole. Given a set of concept names N_C, the set of $\mathcal{SHIQ}$ concepts is the minimal set such that each $A \in N_C$ is a $\mathcal{SHIQ}$ concept (called an *atomic concept*) and, for C and D $\mathcal{SHIQ}$ concepts, R a role, S a simple role, and n a positive integer, $\top, \bot, \neg C, C \sqcap D, C \sqcup D, \exists R.C, \forall R.C, \leq n\ S.C$ and $\geq n\ S.C$ are also $\mathcal{SHIQ}$ concepts. A *TBox* $\mathcal{O}_\mathcal{T}$ is a finite set of *concept inclusion axioms* $C \sqsubseteq D$, where C and D are $\mathcal{SHIQ}$ concepts. An *ABox* $\mathcal{O}_\mathcal{A}$ is a set of *concept membership axioms* $C(a)$, *role membership axioms* $R(a, b)$, and *(in)equality axioms* $a \approx b, a \not\approx b$, where C is a $\mathcal{SHIQ}$ concept, R a role, and a and b individuals. The axioms $C(a)$, $R(a, b)$, $a \approx b$ and $a \not\approx b$ are also called *ABox axioms*; called *atomic ABox axioms* if C is a concept name and R is a role name.

A $\mathcal{SHIQ}$ ontology $\mathcal{O}$ consists of an RBox $\mathcal{O}_\mathcal{R}$, a TBox $\mathcal{O}_\mathcal{T}$, and an ABox $\mathcal{O}_\mathcal{A}$. $\mathcal{O}_\mathcal{R} \cup \mathcal{O}_\mathcal{T}$ is also called the *terminology* of $\mathcal{O}$.

The semantics of a $\mathcal{SHIQ}$ ontology $\mathcal{O}$ is given by translating it into first-order logic by the operator π from Table 1. $\mathcal{O}$ is *consistent/satisfiable* iff there exists a first-order model of $\pi(\mathcal{O})$. In this paper, a first-order model M is represented as a set of ground atoms, where ground atoms in M are interpreted as true; outside M are interpreted as false. A concept C is *satisfiable* in $\mathcal{O}$ iff there exists a first-order model of $\pi(\mathcal{O})$ that satisfies $C(a)$ for some individual a. $\mathcal{O}$ is *coherent* iff all atomic concepts in it are satisfiable.

Table 1. Semantics of $\mathcal{SHIQ}$ by mapping to first-order logic

Mapping concepts to first-order logic	
$\pi_y(\top, x) = 1$ (i.e., true)	$\pi_y(\bot, x) = 0$ (i.e., false)
$\pi_y(A, x) = A(x)$	$\pi_y(\neg C, x) = \neg\pi_y(C, x)$
$\pi_y(C_1 \sqcap C_2, x) = \pi_y(C_1, x) \wedge \pi_y(C_2, x)$	
$\pi_y(C_1 \sqcup C_2, x) = \pi_y(C_1, x) \vee \pi_y(C_2, x)$	
$\pi_y(\exists R.C, x) = \exists y : R(x, y) \wedge \pi_y(C, y)$	$\pi_y(\forall R.C, x) = \forall y : R(x, y) \rightarrow \pi_y(C, y)$
$\pi_y(\leq_n R.C, x) = \forall y_1, \ldots, y_{n+1} : \bigwedge_{i=1}^{n+1}[R(x, y_i) \wedge \pi_x(C, y_i)] \rightarrow \bigvee_{i=1}^{n}\bigvee_{j=i+1}^{n+1} y_i \approx y_j$	
$\pi_y(\geq_n R.C, x) = \exists y_1, \ldots, y_n : \bigwedge_{i=1}^{n}[R(x, y_i) \wedge \pi_x(C, y_i)] \wedge \bigwedge_{i=1}^{n-1}\bigwedge_{j=i+1}^{n} y_i \not\approx y_j$	
Mapping axioms to first-order logic	
$\pi(C \sqsubseteq D) = \forall x : \pi_y(C, x) \rightarrow \pi_y(D, x)$	$\pi(R \sqsubseteq S) = \forall x, y : R(x, y) \rightarrow S(x, y)$
$\pi(Trans(R)) = \forall x, y, z : R(x, y) \wedge R(y, z) \rightarrow R(x, z)$	
$\pi(C(a)) = \pi_y(C, a)$	$\pi(R(a, b)) = R(a, b)$
$\pi(a \approx b) = a \approx b$	$\pi(a \not\approx b) = a \not\approx b$
$\pi(\mathcal{O}) = \bigwedge_{R \in N_R} \forall x, y : R(x, y) \leftrightarrow R^-(y, x) \wedge \bigwedge_{ax \in \mathcal{O}} \pi(ax)$	

Note: x is a meta variable and is substituted with the actual variable. π_x is obtained from π_y by simultaneously substituting all $y_{(i)}$ with $x_{(i)}$ respectively, and π_y with π_x.

3.2 Lexicographic Inference

The lexicographic inference is originally defined for *stratified knowledge bases* in propositional logic [3]. A stratified knowledge base S is viewed as a set of formulas and is divided into a set of strata $\{S_1, \ldots, S_n\}$, where S_i is a subset of formulas in S such that $S = \bigcup_{i=1}^{n} S_i$ and $S_j \cap S_k = \emptyset$ for any $j \neq k$. S can be written as $(S_1, \ldots, S_n)$. The formulas in S_i have the same priority and have a higher priority than the ones in S_{i+1}. Let $|S_{(i)}|$ denote the number of formulas in $S_{(i)}$. The *lexicographic ordering* is a complete preordering between any two subbases $A = (A_1, \ldots, A_n)$ and $B = (B_1, \ldots, B_n)$ of $S = (S_1, \ldots, S_n)$, where $A_i \subseteq S_i$ and $B_i \subseteq S_i$ for any i, defined as follows: $A <^{lex} B$ iff there exists i such that $|A_i| < |B_i|$ and $|A_j| = |B_j|$ for any $j < i$; $A =^{lex} B$ iff $|A_i| = |B_i|$ for any i. By $A \leq^{lex} B$ we denote $A <^{lex} B$ or $A =^{lex} B$. S is called *consistent* if it has a model. A *lex-maximal consistent subbase* S' of S is defined as a consistent subbase of S such that for any consistent subbase S'' of S, $S'' \leq^{lex} S'$. A formula ψ is called a *lex-consistent consequence* of S if for any lex-maximal consistent

subbase S' of S, $S' \models \psi$, i.e., every model of S' is a model of ψ. Given a stratified knowledge base S and a formula ψ, the *lexicographic inference* problem checks if ψ is a lex-consistent consequence of S.

4 Lexicographic Inference in DLs

In order to apply lexicographic inference to DLs, we view a DL-based ontology as a set of axioms (i.e. as a syntactic object). This syntactic approach to treating DL-based ontologies is commonly used in handling inconsistency [11]. From this point of view, two DL-based ontologies are regarded as the same iff they have the same set of axioms.

A *stratified ontology* $\mathcal{O}$ is an ontology divided into a set of strata $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$, where $\mathcal{O}_i$ is a subset of axioms in $\mathcal{O}$ such that $\mathcal{O} = \bigcup_{i=1}^{n} \mathcal{O}_i$ and $\mathcal{O}_j \cap \mathcal{O}_k = \emptyset$ for any $j \neq k$. $\mathcal{O}$ is written as $(\mathcal{O}_1, \ldots, \mathcal{O}_n)$, where the axioms in $\mathcal{O}_i$ have the same priority and have a higher priority than the ones in $\mathcal{O}_{i+1}$. By $|\mathcal{O}_{(i)}|$ we denote *the number of axioms* in $\mathcal{O}_{(i)}$. Then the notions of *lexicographic ordering* and *lex-maximal consistent subontology* are defined analogously as in stratified knowledge bases by treating axioms as formulas [20]. Since an incoherent ontology cannot deduce nontrivial consequences on the unsatisfiable concepts, one may expect that a lex-maximal subontology is not only consistent but also coherent. For example, in the following incoherent but consistent ontology $\mathcal{O} = \{A \sqsubseteq B, A \sqsubseteq \neg B, B(a)\}$, we trivially have $\mathcal{O} \not\models A(x)$ for any individual x because A is unsatisfiable in $\mathcal{O}$. Hence, we introduce the notion of *lex-maximal coherent subontology*. A lex-maximal coherent subontology $\mathcal{O}'$ of $\mathcal{O}$ is defined as a coherent subontology of $\mathcal{O}$ such that for any coherent subontology $\mathcal{O}''$ of $\mathcal{O}$, $\mathcal{O}'' \leq^{lex} \mathcal{O}'$. Two sorts of lexicographic consequences in DLs are defined below.

Definition 1. For a stratified ontology $\mathcal{O}$, an axiom ax is called a *lex-consistent consequence* (resp. *lex-coherent consequence*) of $\mathcal{O}$, written $\mathcal{O} \vdash^{lex}_{cons} ax$ (resp. $\mathcal{O} \vdash^{lex}_{cohe} ax$), if for any lex-maximal consistent (resp. coherent) subontology $\mathcal{O}'$ of $\mathcal{O}$, $\mathcal{O}' \models ax$, i.e., every model of $\mathcal{O}'$ is a model of ax.

It should be noted that a lex-maximal coherent subontology is not necessarily a lex-maximal consistent subontology, and vice versa. There is no straightforward correspondence between lex-consistent consequences and lex-coherent consequences, as shown in the following example.

Example 1. Let $\mathcal{O} = (\{A \sqsubseteq \bot\}, \{A(a)\})$. Then $\mathcal{O}' = (\emptyset, \{A(a)\})$ is the unique lex-maximal coherent subontology of $\mathcal{O}$, but it is not a lex-maximal consistent subontology of $\mathcal{O}$, because $\mathcal{O}'' = (\{A \sqsubseteq \bot\}, \emptyset)$ is consistent and $\mathcal{O}' <^{lex} \mathcal{O}''$. On the other hand, $\mathcal{O}''$ is the unique lex-maximal consistent subontology of $\mathcal{O}$, but it is not coherent. Hence, $\mathcal{O} \vdash^{lex}_{cohe} A(a)$ but $\mathcal{O} \nvdash^{lex}_{cons} A(a)$; $\mathcal{O} \vdash^{lex}_{cons} A \sqsubseteq \bot$ but $\mathcal{O} \nvdash^{lex}_{cohe} A \sqsubseteq \bot$.

In this paper we consider checking both sorts of lexicographic consequences, where the axiom ax in Definition 1 can be a concept membership axiom $C(a)$ or

a concept inclusion axiom $C \sqsubseteq D$. The following theorem shows a reduction from the problem of checking a lex-coherent consequence to that of checking a lex-consistent consequence. So we focus on the problem of checking a lex-consistent consequence.

Theorem 1. *Let $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ be a stratified DL-based ontology, and ax a DL axiom. Then $\mathcal{O} \vdash^{lex}_{cohe} ax$ iff $\mathcal{O}' \vdash^{lex}_{cons} ax$, where $\mathcal{O}' = (\mathcal{A}, \mathcal{O}_1, \ldots, \mathcal{O}_n)$ and $\mathcal{A} = \{A(a) \mid A$ is an atomic concept in $\mathcal{O}$, and a is a new globally unique individual not occurring in $\mathcal{O}$ and $ax\}$.*

Proof. It is sufficient to show a bijection between the set of lex-maximal coherent subontologies of $\mathcal{O}$ and the set of lex-maximal consistent subontologies of $\mathcal{O}'$.

(1) Let $S = (S_1, \ldots, S_n)$ be a lex-maximal coherent subontology of $\mathcal{O}$. Then $S' = (\mathcal{A}, S_1, \ldots, S_n)$ is obviously consistent. S' must be a lex-maximal consistent subontology of $\mathcal{O}'$, otherwise there exists a consistent subontology $S'' = (S''_0, S''_1, \ldots, S''_n)$ of $\mathcal{O}'$ such that $S' <^{lex} S''$. Then $S''_0 = \mathcal{A}$ and thus $(S''_1, \ldots, S''_n)$ is coherent. But then $S <^{lex} (S''_1, \ldots, S''_n)$ contradicts that S is a lex-maximal coherent subontology of $\mathcal{O}$.

(2) Let $S = (S_0, S_1, \ldots, S_n)$ be a lex-maximal consistent subontology of $\mathcal{O}'$. Since $\{\mathcal{A}, \emptyset, \ldots, \emptyset\}$ is consistent, we have $S_0 = \mathcal{A}$, so $S' = (S_1, \ldots, S_n)$ is coherent. S' must be a lex-maximal coherent subontology of $\mathcal{O}$, otherwise there exists a coherent subontology $S'' = (S''_1, \ldots, S''_n)$ of $\mathcal{O}$ such that $S' <^{lex} S''$. Then $(\mathcal{A}, S''_1, \ldots, S''_n)$ is consistent. But then $S <^{lex} (\mathcal{A}, S''_1, \ldots, S''_n)$ contradicts that S is a lex-maximal consistent subontology of $\mathcal{O}'$. □

Example 2. Our running example is a stratified ontology $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4)$, where $\mathcal{O}_1 = \{A(a),\ B(b),\ \leq_1 T.\top(a)\}$, $\mathcal{O}_2 = \{T(a,b), T(a,c), b \not\approx c\}$, $\mathcal{O}_3 = \{A \sqsubseteq \exists R.B,\ \exists S.B \sqsubseteq \neg A\}$ and $\mathcal{O}_4 = \{R \sqsubseteq S\}$. It can be checked that there are three lex-maximal consistent subontologies of $\mathcal{O}$, i.e. $\mathcal{O}^1 = (\{A(a),\ B(b),\ \leq_1 T.\top(a)\}, \{T(a,b), T(a,c)\}, \{A \sqsubseteq \exists R.B, \exists S.B \sqsubseteq \neg A\}, \emptyset)$, $\mathcal{O}^2 = (\{A(a), B(b),\ \leq_1 T.\top(a)\}, \{T(a,b),\ b \not\approx c\}, \{A \sqsubseteq \exists R.B,\ \exists S.B \sqsubseteq \neg A\},\ \emptyset)$ and $\mathcal{O}^3 = (\{A(a),\ B(b), \leq_1 T.\top(a)\}, \{T(a,c),\ b \not\approx c\}, \{A \sqsubseteq \exists R.B,\ \exists S.B \sqsubseteq \neg A\},\ \emptyset)$. It can then be seen that $A(a)$ is a lex-consistent consequence of $\mathcal{O}$, while $B(c)$ is not. □

5 Computing Lexicographic Inference in $\mathcal{SHIQ}$

The number of lex-maximal consistent subontologies of a stratified $\mathcal{SHIQ}$ ontology $\mathcal{O}$ can be exponential in $|\mathcal{O}|$ (even in $|\mathcal{O}_\mathcal{A}|$, the number of ABox axioms in $\mathcal{O}$). Take an ontology $\mathcal{O}^\dagger = (\mathcal{O}^\dagger_1, \ldots, \mathcal{O}^\dagger_n)$ for example, where $\mathcal{O}^\dagger_1 = \{A \sqcap B \sqsubseteq \bot\}$ and $\mathcal{O}^\dagger_i = \{A(a_i),\ B(a_i)\}$ for $2 \leq i \leq n$. $\mathcal{O}^\dagger$ has 2^{n-1} lex-maximal consistent subontologies. Note also that the time complexity for lexicographic inference in propositional logic is Δ^p_2-complete [5], i.e. exactly in polynomial calls to an NP oracle. It is not desirable to compute all lex-maximal consistent subontologies before checking lex-consistent consequences, because such computation needs up to exponential calls to a $\mathcal{SHIQ}$ reasoner, where each call is worst-case NP-complete in data complexity [14].

To obtain a worst-case optimal method for computing lexicographic inference in $\mathcal{SHIQ}$, we consider transforming $\mathcal{SHIQ}$ to propositional logic. $\mathcal{SHIQ}$ is a subset of first-order logic. The hardness for transforming $\mathcal{SHIQ}$ to propositional logic lies on handling function symbols. Though the KAON2 transformation [14, 21] can get rid of function symbols and obtain an equisatisfiable DATALOG$^\vee$ [8] program from a $\mathcal{SHIQ}$ ontology, it does not maintain the correspondence between resulting DATALOG$^\vee$ rules and original axioms in the input ontology. We therefore extend the KAON2 transformation to maintain such correspondence by introducing new nullary atoms one-to-one corresponding to axioms in the original ontology. Afterwards, we ground the transformed DATALOG$^\vee$ program and apply current powerful SAT solvers to compute lexicographic inference. In this way we obtain a worst-case optimal method in data complexity. There remains a practical problem in such method, i.e. SAT solvers lack of scalability for handling large propositional programs. To tackle this problem, we partition the transformed propositional program so that we can apply SAT solvers to handle much smaller subprograms.

Our approach is outlined as follows. Let $\mathcal{O}$ be a (possibly inconsistent) stratified $\mathcal{SHIQ}$ ontology. We first consider the basic case, i.e. checking if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ for $A(a)$ an atomic concept membership axiom, then consider the checking of other consequences. The basic case is addressed in two phases. In phase 1 (subsection 5.1), we compute a DATALOG$^\vee$ program $\mathcal{R}(\mathcal{O})$, called the *repair program* of $\mathcal{O}$, by using the extended KAON2 transformation. In phase 2 (subsection 5.2), we first ground $\mathcal{R}(\mathcal{O})$ to GR($\mathcal{O}$), then treat the problem of deciding if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ as a set of satisfiability problems over GR($\mathcal{O}$) and solve them by polynomial calls to a SAT solver; in this phase we also exploit partition-based optimizations (subsection 5.3). The problem of checking other consequences is first reduced to the basic case, then solved in the same way (subsection 5.4).

5.1 Computing the Repair Program

To compute lexicographic inference in $\mathcal{SHIQ}$, we associate each axiom $ax \in \mathcal{O}$ with a new nullary *decision atom* $\hbar_{ax}$, such that the truth value of $\hbar_{ax}$ determines the existence of ax in $\mathcal{O}$. It should be noted that $S \sqsubseteq R$ and $Inv(S) \sqsubseteq Inv(R)$ (resp. $Trans(R)$ and $Trans(Inv(R))$) are treated as the same axiom and thus associated with the same decision atom, because they are assumed present or absent together (see this assumption in Preliminaries). Let X be the set of decision atoms wrt $\mathcal{O}$, i.e., $X = \{\hbar_{ax} \mid ax \in \mathcal{O}\}$. We extend the KAON2 transformation [14, 21] to compile a DATALOG$^\vee$ program $\mathcal{R}(\mathcal{O})$ (called the *repair program* of $\mathcal{O}$), such that for any truth assignment ϕ_X on X, the *reduction* of $\mathcal{O}$ wrt ϕ_X (i.e. $\mathcal{O} \downarrow \phi_X$, see Definition 2) is satisfiable iff the *reduction* of $\mathcal{R}(\mathcal{O})$ wrt ϕ_X (i.e. $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$, see Definition 3) is satisfiable. Simply speaking, $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$ is a DATALOG$^\vee$ program without atoms in X. The relationship between $\mathcal{O} \downarrow \phi_X$ and $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$ implies a correspondence between lex-maximal consistent subontologies of $\mathcal{O}$ and certain optimal models of $\mathcal{R}(\mathcal{O})$, where the optimality is defined over X.

Definition 2. Let $\mathcal{O}$ be a $\mathcal{SHIQ}$ ontology and ϕ_X a truth assignment on the set X of decision atoms. The *reduction* of $\mathcal{O}$ w.r.t. ϕ_X, written $\mathcal{O} \downarrow \phi_X$, is a subontology obtained from $\mathcal{O}$ by deleting each axiom ax such that $\phi_X(\hbar_{ax}) = 1$.

Definition 3. Let P be a logic program, i.e. a set of rules (or clauses), X be a set of ground atoms that only occur in rule heads (or only occur positively) in P, and ϕ_X be a truth assignment on X. The *reduction* of P w.r.t. ϕ_X, written $P \downarrow \phi_X$, is a subprogram obtained from P by deleting each rule (or clause) in P that has a head atom (or positive atom) $\alpha \in X$ such that $\phi_X(\alpha) = 1$, and by removing any ground atom $\alpha \in X$ from remaining rules (or clauses).

The main steps for computing the repair program include converting $\mathcal{SHIQ}$ to $\mathcal{ALCHIQ}$ (i.e. $\mathcal{SHIQ}$ without transitive roles), clausifying axioms, embedding decision atoms into the initial clause set, and appending to the clause set all nonredundant consequences.

Simplifying the Elimination of Transitivity Axioms In the original KAON2 transformation, a $\mathcal{SHIQ}$ ontology $\mathcal{O}$ is initially converted to an equisatisfiable $\mathcal{ALCHIQ}$ ontology $\Omega(\mathcal{O})$, by eliminating all transitivity axioms, and by adding the axiom $\forall R.C \sqsubseteq \forall S.(\forall S.C)$ for each concept $\forall R.C \in \text{clos}(\mathcal{O})$ and each role S such that $S \sqsubseteq^* R$ and $Trans(S) \in \mathcal{O}$, where $\text{clos}(\mathcal{O})$ is the *concept closure* of $\mathcal{O}$ (see Definition 4). To correctly embed decision atoms into the clauses transformed from axioms, we remove some redundant axioms from $\Omega(\mathcal{O})$. By $\Omega^-(\mathcal{O})$ we denote the $\mathcal{ALCHIQ}$ ontology converted from $\mathcal{O}$ by eliminating all transitivity axioms, and by adding the axiom $\forall S.C \sqsubseteq \forall S.(\forall S.C)$ for each concept $\forall R.C \in \text{clos}(\mathcal{O})$ and role S such that $S \sqsubseteq^* R$ and $Trans(S) \in \mathcal{O}$. It can be shown in Lemma 1 that $\mathcal{O}$ and $\Omega^-(\mathcal{O})$ are equisatisfiable.

Definition 4 (Definition 5.2.2 [21]). For a $\mathcal{SHIQ}$ ontology $\mathcal{O}$, the *concept closure* of $\mathcal{O}$, written $\text{clos}(\mathcal{O})$, is the smallest set of concepts satisfying the following conditions (where $\text{NNF}(C)$ denotes the negation normal form of C):

- If $C \sqsubseteq D \in \mathcal{O}_\mathcal{T}$, then $\text{NNF}(\neg C \sqcap D) \in \text{clos}(\mathcal{O})$;
- If $C(a) \in \mathcal{O}_\mathcal{A}$, then $\text{NNF}(C) \in \text{clos}(\mathcal{O})$;
- If $C \in \text{clos}(\mathcal{O})$ and D occurs in C, then $D \in \text{clos}(\mathcal{O})$;
- If $\leq_n R.C \in \text{clos}(\mathcal{O})$, then $\text{NNF}(\neg C) \in \text{clos}(\mathcal{O})$;
- If $\forall R.C \in \text{clos}(\mathcal{O})$, $S \sqsubseteq^* R$, and $Trans(S) \in \mathcal{O}_\mathcal{R}$, then $\forall S.C \in \text{clos}(\mathcal{O})$.

Lemma 1. *A $\mathcal{SHIQ}$ ontology $\mathcal{O}$ is satisfiable iff $\Omega^-(\mathcal{O})$ is satisfiable.*

Proof. Obviously, $\Omega^-(\mathcal{O}) \sqsubseteq \Omega(\mathcal{O})$. By Theorem 5.2.3 [21], $\mathcal{O}$ and $\Omega(\mathcal{O})$ are equisatisfiable, so $\Omega^-(\mathcal{O})$ is satisfiable if $\mathcal{O}$ is satisfiable. On the other hand, if $\Omega^-(\mathcal{O})$ is satisfiable, there will be a model I of $\Omega^-(\mathcal{O})$. Consider each axiom $\forall R.C \sqsubseteq \forall S.(\forall S.C)$ in $\Omega(\mathcal{O}) \setminus \Omega^-(\mathcal{O})$. For any two individual a and b such that $I \models \forall R.C(a)$ and $I \models S(a, b)$, since $I \models S \sqsubseteq^* R$, we have $I \models R(a, b)$ and then $I \models C(b)$, so $I \models \forall R.C \sqsubseteq \forall S.C$. In addition, $I \models \forall S.C \sqsubseteq \forall S.(\forall S.C)$, so $I \models \forall R.C \sqsubseteq \forall S.(\forall S.C)$. I.e., I is also a model of $\Omega(\mathcal{O})$, so $\mathcal{O}$ is satisfiable. □

Embedding Decision Atoms into the Initial Clause Set After a given $\mathcal{SHIQ}$ ontology is converted to an equisatisfiable $\mathcal{ALCHIQ}$ ontology $\Omega^-(\mathcal{O})$, each axiom in $\Omega^-(\mathcal{O})$ can be transformed to a set of clauses via the well-known *structural transformation* [26], mapping to first-order formulas, Skolemization and rewriting into conjunctive normal form. By $\mathrm{Cls}(ax)$ we denote the set of clauses obtained from the axiom ax via the translation process given in Definition 5. For each axiom $ax \in \Omega^-(\mathcal{O})$ and each clause $cl \in \mathrm{Cls}(ax)$, by $\mathrm{em}(cl, ax)$ we denote the modified clause of cl into which a decision atom about ax is embedded, defined as follows: if $ax \in \Omega^-(\mathcal{O}) \setminus \mathcal{O}$, ax must be of the form $\forall S.C \sqsubseteq \forall S.(\forall S.C)$, then $\mathrm{em}(cl, ax) \stackrel{def}{=} cl \vee \hbar_{Trans(S)}$; otherwise, $\mathrm{em}(cl, ax) \stackrel{def}{=} cl \vee \hbar_{ax}$. Recall that $\hbar_{ax}$ is the corresponding decision atom of ax. By $\Xi(\mathcal{O})$ we denote $\mathrm{Cls}_{N_R} \cup \bigcup_{ax \in \Omega^-(\mathcal{O})} \{\mathrm{em}(cl, ax) \mid cl \in \mathrm{Cls}(ax)\}$, where the clause set $\mathrm{Cls}_{N_R} = \{\neg R(x,y) \vee R^-(y,x), \neg R^-(x,y) \vee R(y,x) \mid R \in N_R\}$ with x and y variables is introduced for mapping $\Omega^-(\mathcal{O})$ to first-order logic (see Table 1).

Definition 5. The result $\Theta(A \sqsubseteq C)$ of applying the *structural transformation* to $A \sqsubseteq C$ is recursively defined as follows, where A and B are concept names or $\top$, C, C_1 and C_2 are arbitrary concepts of the negation normal form, R is a role, and Q_X is X if X is a literal concept or else a new globally unique concept name for X (note that $\neg\neg A$ is treated as A):

- $\Theta(A \sqsubseteq A) = \emptyset$; $\quad \Theta(\neg A \sqsubseteq \neg A) = \emptyset$;
- $\Theta(A \sqsubseteq B) = \{A \sqsubseteq B\}$; $\quad \Theta(A \sqsubseteq \neg B) = \{A \sqsubseteq \neg B\}$;
- $\Theta(A \sqsubseteq C_1 \sqcap C_2) = \Theta(A \sqsubseteq C_1) \cup \Theta(A \sqsubseteq C_2)$;
- $\Theta(A \sqsubseteq C_1 \sqcup C_2) = \{A \sqsubseteq Q_{C_1} \sqcup Q_{C_2}\} \cup \Theta(Q_{C_1} \sqsubseteq C_1) \cup \Theta(Q_{C_2} \sqsubseteq C_2)$;
- $\Theta(A \sqsubseteq \exists R.C) = \{A \sqsubseteq \exists R.Q_C\} \cup \Theta(Q_C \sqsubseteq C)$;
- $\Theta(A \sqsubseteq \forall R.C) = \{A \sqsubseteq \forall R.Q_C\} \cup \Theta(Q_C \sqsubseteq C)$;
- $\Theta(A \sqsubseteq \geq_n R.C) = \{A \sqsubseteq \geq_n R.Q_C\} \cup \Theta(Q_C \sqsubseteq C)$;
- $\Theta(A \sqsubseteq \leq_n R.C) = \{A \sqsubseteq \leq_n R.\neg Q_D\} \cup \Theta(Q_D \sqsubseteq D)$ for $D = \mathrm{NNF}(\neg C)$.

Let Cls_{fo} denote the operator for clausifying a first-order formula. Then, for each axiom ax in an $\mathcal{ALCHIQ}$ ontology, $\mathrm{Cls}(ax)$ is defined as follows (see Table 1 for the operator π which maps axioms to first-order formulas):

- If ax is of the form $C \sqsubseteq D$ where C and D are concepts, then $\mathrm{Cls}(ax) = \bigcup_{\varepsilon \in \Theta(\top \sqsubseteq \mathrm{NNF}(\neg C \sqcup D))} \mathrm{Cls}_{fo}(\forall x : \pi_y(\varepsilon, x))$;
- If ax is of the form $R \sqsubseteq S$ where R and S are roles, then $\mathrm{Cls}(ax) = \{\neg R(x,y) \vee S(x,y)\}$;
- If ax is of the form $C(a)$ where C is not a literal concept, then $\mathrm{Cls}(ax) = \{Q_C(a)\} \cup \mathrm{Cls}(Q_C \sqsubseteq C)$, where Q_C is a new globally unique concept name for C;
- If ax is of the other forms, i.e., ax is of the form $(\neg)A(a)$, $R(a,b)$, $a \approx b$ or $a \not\approx b$, $\mathrm{Cls}(ax) = \mathrm{Cls}_{fo}(\pi(ax))$.

Example 3 (Example 2 continued). Consider the stratified ontology $\mathcal{O}$ in Example 2. Let $ax_1 = A(a)$, $ax_2 = B(b)$, $ax_3 = \leq_1 T.\top(a)$, $ax_4 = T(a,b)$, $ax_5 = T(a,c)$, $ax_6 = b \not\approx c$, $ax_7 = A \sqsubseteq \exists R.B$, $ax_8 = \exists S.B \sqsubseteq \neg A$ and

$ax_9 = R \sqsubseteq S$. Then $\mathcal{O}$ can be rewritten as $(\{ax_1, ax_2, ax_3\}, \{ax_4, ax_5, ax_6\}, \{ax_7, ax_8\}, \{ax_9\})$. $\Xi(\mathcal{O})$ consists of the following clauses. Note that the clauses on R^- and S^- are removed from $\Xi(\mathcal{O})$, because neither R^- nor S^- occurs in $\mathcal{O}$ and the removal does not affect the satisfiability of $\Xi(\mathcal{O}) \downarrow \phi_X$ for any truth assignment ϕ_X on $X = \{\hbar_{ax_1}, \ldots, \hbar_{ax_9}\}$.

$cl_1 : \ A(a) \vee \hbar_{ax_1}. \quad cl_2 : \ B(b) \vee \hbar_{ax_2}. \quad cl_3 : \ Q_1(a) \vee \hbar_{ax_3}.$
$cl_4 : \ y_1 \approx y_2 \vee \neg Q_1(x) \vee \neg T(x, y_1) \vee \neg T(x, y_2) \vee \hbar_{ax_3}.$
$cl_5 : \ T(a, b) \vee \hbar_{ax_4}. \quad cl_6 : \ T(a, c) \vee \hbar_{ax_5}. \quad cl_7 : \ \neg(b \approx c) \vee \hbar_{ax_6}.$
$cl_8 : \ R(x, f(x)) \vee \neg A(x) \vee \hbar_{ax_7}. \quad cl_9 : \ B(f(x)) \vee \neg A(x) \vee \hbar_{ax_7}.$
$cl_{10} : \ \neg A(x) \vee \neg S(x, y) \vee \neg B(y) \vee \hbar_{ax_8}. \quad cl_{11} : \ S(x, y) \vee \neg R(x, y) \vee \hbar_{ax_9}.$ □

Lemma 2. *For a $\mathcal{SHIQ}$ ontology $\mathcal{O}$, a set of decision atoms $X = \{\hbar_{ax} \mid ax \in \mathcal{O}\}$ and a truth assignment ϕ_X on X, $\mathcal{O} \downarrow \phi_X$ is satisfiable iff $\Xi(\mathcal{O}) \downarrow \phi_X$ is satisfiable.*

Proof. For any $\mathcal{SHIQ}$ ontology $\mathcal{O}'$, we define $\Omega^-(\mathcal{O}') \downarrow \phi_X$ as the subontology obtained from $\Omega^-(\mathcal{O}')$ by deleting each axiom $ax \in \Omega^-(\mathcal{O}') \cap \mathcal{O}'$ such that $\phi_X(\hbar_{ax}) = 1$ and each $\forall S.C \sqsubseteq \forall S.(\forall S.C) \in \Omega^-(\mathcal{O}') \setminus \mathcal{O}'$ such that $\phi_X(\hbar_{Trans(S)}) = 1$. Note that for any concept C in an axiom of the form $\forall S.C \sqsubseteq \forall S.(\forall S.C)$ with $Trans(S) \in \mathcal{O} \downarrow \phi_X$, C occurs in $\mathcal{O}$ if it occurs in $O \downarrow \phi_X$, but not vice versa. So $\Omega^-(\mathcal{O} \downarrow \phi_X) \subseteq \Omega^-(\mathcal{O}) \downarrow \phi_X$ and the difference between them consists of axioms of the form $\forall S.C \sqsubseteq \forall S.(\forall S.C)$ with $Trans(S) \in \mathcal{O} \downarrow \phi_X$.

We show that $\mathcal{O} \downarrow \phi_X$ is satisfiable iff $\Omega^-(\mathcal{O}) \downarrow \phi_X$ is satisfiable. ($\Rightarrow$) If $\mathcal{O} \downarrow \phi_X$ is satisfiable, there exists a model I of $\mathcal{O} \downarrow \phi_X$. For each axiom $ax \in (\Omega^-(\mathcal{O}) \downarrow \phi_X) \setminus \mathcal{O}$, ax is of the form $\forall S.C \sqsubseteq \forall S.(\forall S.C)$, where $Trans(S) \in \mathcal{O} \downarrow \phi_X$. Since $I \models Trans(S)$, $I \models \forall S.C \sqsubseteq \forall S.(\forall S.C)$ for any concept C. So I is also a model of $\Omega^-(\mathcal{O}) \downarrow \phi_X$. ($\Leftarrow$) If $\Omega^-(\mathcal{O}) \downarrow \phi_X$ is satisfiable, $\Omega^-(\mathcal{O} \downarrow \phi_X)$ is satisfiable too. By Lemma 1, $\mathcal{O} \downarrow \phi_X$ is satisfiable.

Let Cls_{N_R} denote the set of clauses $\{\neg R(x, y) \vee R^-(y, x), \neg R^-(x, y) \vee R(y, x) \mid R \in N_R\}$. For each axiom ax in $\Omega^-(\mathcal{O}) \downarrow \phi_X$, $\{\mathrm{em}(cl, ax) \mid cl \in \mathrm{Cls}(ax)\} \downarrow \phi_X = \mathrm{Cls}(ax)$. For each axiom ax in $\Omega^-(\mathcal{O}) \setminus (\Omega^-(\mathcal{O}) \downarrow \phi_X)$, $\{\mathrm{em}(cl, ax) \mid cl \in \mathrm{Cls}(ax)\} \downarrow \phi_X = \emptyset$. So $\Xi(\mathcal{O}) \downarrow \phi_X = \mathrm{Cls}_{N_R} \cup \bigcup_{ax \in \Omega^-(\mathcal{O})} \{\mathrm{em}(cl, ax) \mid cl \in \mathrm{Cls}(ax)\} \downarrow \phi_X = \mathrm{Cls}_{N_R} \cup \bigcup_{ax \in \Omega^-(\mathcal{O}) \downarrow \phi_X} \mathrm{Cls}(ax)$. Note that the structural transformation, mapping to first-order formulas (see Table 1), Skolemization and rewriting into conjunctive normal form do not affect satisfiability, i.e., $\mathrm{Cls}_{N_R} \cup \bigcup_{ax \in \Omega^-(\mathcal{O}) \downarrow \phi_X} \mathrm{Cls}(ax)$ and $\Omega^-(\mathcal{O}) \downarrow \phi_X$ are equisatisfiable, so $\Xi(\mathcal{O}) \downarrow \phi_X$ and $\mathcal{O} \downarrow \phi_X$ are equisatisfiable. □

Extending $\mathcal{BS}^+_{DL}$ to $\mathcal{BS}^+_{DL^+}$ The KAON2 transformation extends the Basic Superposition ($\mathcal{BS}$) calculus [2, 22] with a *decomposition* rule, yielding an extended calculus $\mathcal{BS}^+$, and then parameterizes $\mathcal{BS}^+$ to $\mathcal{BS}^+_{DL}$ to handle clauses transformed from DL axioms. The decomposition rule (cf. Definition 5.4.7 [21]) is an additional inference rule that decomposes some conclusions of $\mathcal{BS}$ into simpler rules so as to guarantee termination. In order to handle clauses with decision

atoms, we further extend $\mathcal{BS}^+_{DL}$ to $\mathcal{BS}^+_{DL^+}$ by considering decision atoms in the term ordering. The extended term ordering is a *lexicographic path ordering* induced by a total precedence $\succ$ such that $f \succ c \succ P \succ Q_{R,f} \succ \hbar_{ax} \succ \top$, for each function symbol f, constant c, predicate P, predicate $Q_{R,f}$ (introduced by the decomposition rule) and decision atom $\hbar_{ax}$. The selection function and the decomposition rule in $\mathcal{BS}^+_{DL^+}$ are the same as that in $\mathcal{BS}^+_{DL}$.

We define $\mathcal{ALCHIQ}^+$-closures (see in Table 2) as $\mathcal{ALCHIQ}$-closures[7] in Table 5.2 [21] extended with disjunction of decision atoms. Then $\Xi(\mathcal{O})$ consists of $\mathcal{ALCHIQ}^+$-closures; any $\mathcal{BS}^+_{DL^+}$ inference, when applied to $\mathcal{ALCHIQ}^+$-closures, produces an $\mathcal{ALCHIQ}^+$-closure or a redundant closure (Lemma 3). By size($\mathcal{O}$) we denote the *size* of $\mathcal{O}$ with numbers coded in unary, i.e., size($\leq_n S.C$) = size($\geq_n S.C$) = size(C) + n + 1. Note that in [14, 21] $|\mathcal{O}|$ denotes the size of $\mathcal{O}$, while in this paper $|\mathcal{O}|$ denotes the number of axioms in $\mathcal{O}$. It can be shown in Lemma 4 that saturating $\Xi(\mathcal{O})$ by $\mathcal{BS}^+_{DL^+}$ with eager elimination of redundancy terminates in exponential steps.

Lemma 3. *Let $N_0, \ldots, N_i \cup \{cl\}$ be a $\mathcal{BS}^+_{DL^+}$-derivation, where $N_0 = \Xi(\mathcal{O})$ and cl is the conclusion derived from premises in N_i. Then cl is either an $\mathcal{ALCHIQ}^+$-closure or it is redundant in N_i.*

Proof Sketch. The proof is by considering all $\mathcal{BS}^+_{DL^+}$ inferences on all types of $\mathcal{ALCHIQ}^+$-closures analogously as in Lemma 5.3.6 and Theorem 5.4.8 [21]. □

Lemma 4. *For a $\mathcal{SHIQ}$ ontology $\mathcal{O}$, saturating $\Xi(\mathcal{O})$ by $\mathcal{BS}^+_{DL^+}$, with eager application of redundancy elimination rules, terminates in time exponential in* size($\mathcal{O}$)*, for unary coding of numbers in input.*

Proof Sketch. The proof is by considering the maximum number of each type of $\mathcal{ALCHIQ}^+$-closures analogously as in Lemma 5.3.10 and Theorem 5.4.8 [21]. The number of $\mathcal{ALCHIQ}^+$-closures is at most exponential in size($\mathcal{O}$) for unary coding of numbers in input. By Lemma 3, each $\mathcal{BS}^+_{DL^+}$ inference produces an $\mathcal{ALCHIQ}^+$-closure or a redundant closure. Furthermore, it can be shown that any saturation derives at most exponentially many redundant closures. Hence, saturating $\Xi(\mathcal{O})$ by $\mathcal{BS}^+_{DL^+}$ takes at most exponential steps. □

Computing the Repair Program from $\Xi(\mathcal{O})$ Let $\Xi_{var}(\mathcal{O})$ denote the set of $\mathcal{ALCHIQ}^+$-closures in $\Xi(\mathcal{O})$ that have variables, and $\Xi_{con}(\mathcal{O}) = \Xi(\mathcal{O}) \setminus \Xi_{var}(\mathcal{O})$. In a way analogous to the KAON2 transformation, we compute a DATALOG$^\vee$ [8] program from $\Xi(\mathcal{O})$ through the following steps:

1. **Saturating $\Xi_{var}(\mathcal{O})$ by $\mathcal{BS}^+_{DL^+}$.** Let $\Gamma_{var} = \Xi_{var}(\mathcal{O}) \cup \text{gen}(\mathcal{O})$, where gen($\mathcal{O}$) (cf. Definition 5.4.9 [21]) is the set of clauses of the form $\neg Q_{R,f}(x) \vee$

[7] A closure $C \cdot \sigma$ consists of a skeleton clause C and a substitution σ, and is logically equivalent to a clause $C\sigma$. Closures and clauses are called interchangeably in this paper.

Table 2. Types of $\mathcal{ALCHIQ}^+$-closures

1	$\neg R(x,y) \vee Inv(R)(y,x)$
2	$\neg R(x,y) \vee S(x,y) \vee \mathbf{\hbar_{ax}}$
3	$\mathbf{P}(x) \vee R(x, \langle f(x) \rangle) \vee \mathbf{\hbar_{ax}}$
4	$\mathbf{P}(x) \vee R([f(x)], x) \vee \mathbf{\hbar_{ax}}$
5	$\mathbf{P_1}(x) \vee \mathbf{P_2}(\langle \mathbf{f}(x) \rangle) \vee \bigvee \langle f(x_i) \rangle \circ \langle f(x_j) \rangle \vee \mathbf{\hbar_{ax}}$ for $\circ \in \{=, \neq\}$
6	$\mathbf{P_1}(x) \vee \mathbf{P_2}([g(x)]) \vee \mathbf{P_3}(\langle \mathbf{f}([g(x)]) \rangle) \vee \bigvee \langle t_i \rangle \circ \langle t_j \rangle \vee \mathbf{\hbar_{ax}}$ for $\circ \in \{=, \neq\}$, where t_i and t_j are either of the form $f([g(x)])$ or of the form x
7	$\mathbf{P_1}(x) \vee \bigvee \neg R(x, y_i) \vee \mathbf{P_2}(\mathbf{y}) \vee \bigvee y_i = y_j \vee \mathbf{\hbar_{ax}}$
8	$\mathbf{R}(\langle \mathbf{a} \rangle, \langle \mathbf{b} \rangle) \vee \mathbf{P}(\langle \mathbf{t} \rangle) \vee \bigvee \langle t_i \rangle \circ \langle t_j \rangle \vee \mathbf{\hbar_{ax}}$ for $\circ \in \{=, \neq\}$, where t, t_i and t_j are either some constant b or a function term $f_i([a])$

Conditions:
(i): In any term $f(t)$, the inner term t occurs marked.
(ii): In all positive equality literals with at least one function symbol, both sizes are marked.

Note: For a term t, $\mathbf{P}(t)$ denotes a disjunction of the form $(\neg)P_1(t) \vee \ldots \vee (\neg)P_n(t)$. $\mathbf{P}(\mathbf{f}(x))$ denotes a disjunction of the form $\mathbf{P_1}(f_1(x)) \vee \ldots \vee \mathbf{P_m}(f_m(x))$. $\mathbf{\hbar_{ax}}$ denotes a disjunction of the form $\hbar_{ax_1} \vee \ldots \vee \hbar_{ax_k}$. $\langle t \rangle$ denotes that the term t may, but need not be marked. In all closure types, some of the disjuncts may be empty.

$R(x, [f(x)])$ that can possibly be introduced by decomposition during the saturation of $\Xi_{var}(\mathcal{O})$, i.e., the decomposition rule introduces only new predicates $Q_{R,f}$ occurring in $\text{gen}(\mathcal{O})$. As in [14, 21], we append $\text{gen}(\mathcal{O})$ to $\Xi_{var}(\mathcal{O})$ in advance, before saturation. Let $\text{SAT}_\text{R}(\Gamma_{var})$ denote the set of $\mathcal{ALCHIQ}^+$-closures of types 1,2,3,5 and 7, obtained by saturating Γ_{var} by $\mathcal{BS}^+_{DL^+}$.

2. **Eliminating function symbols.** Let λ be an operator that maps constant a to a, variable x to x, function term $f(a)$ to a fresh globally unique constant a_f, and function term $f(x)$ to a fresh globally unique variable x_f. Intuitively, λ simulates function terms with new constants or variables. For a clause cl, $\lambda(cl)$ is obtained from cl as follows: each term t in cl is replaced with $\lambda(t)$; for each fresh variable x_f occurring in $\lambda(cl)$, the literal $\neg S_f(x, x_f)$ is added to $\lambda(cl)$; for each variable x only occurring in positive literals in cl, the literal $\neg HU(x)$ is added to $\lambda(cl)$.
For a substitution σ, let $\lambda(\sigma)$ denote the substitution obtained from σ by replacing each assignment $x \mapsto t$ with $x \mapsto \lambda(t)$. By λ^- we denote the inverse of λ, i.e., $\lambda^-(\lambda(\alpha)) = \alpha$ for any term, closure, or any substitution α. Note that λ is injective, but not surjective, so to make the definition of λ^- correct, we assume that λ^- is applicable only to the range of λ.
Let $\text{FF}(\mathcal{O}) = \text{FF}_\lambda(\mathcal{O}) \cup \text{FF}_{Succ}(\mathcal{O}) \cup \text{FF}_{HU}(\mathcal{O}) \cup \Xi_{con}(\mathcal{O})$ denote the function-free version of $\Xi(\mathcal{O})$, where FF_λ, FF_{Succ} and FF_{HU} are defined as follows:

$$\begin{aligned}
\text{FF}_\lambda(\mathcal{O}) &= \{\lambda(cl) | cl \in \text{SAT}_\text{R}(\Gamma_{var})\} \\
\text{FF}_{Succ}(\mathcal{O}) &= \{S_f(a, a_f) | \text{ for each constant } a \text{ and each function symbol } f \text{ in } \Xi(\mathcal{O})\} \\
\text{FF}_{HU}(\mathcal{O}) &= \{HU(a) | \text{ for each constant } a \text{ in } \Xi(\mathcal{O})\} \cup \\
&\quad \{HU(a_f) | \text{ for each constant } a \text{ and function symbol } f \text{ in } \Xi(\mathcal{O})\}
\end{aligned}$$

3. **Removing irrelevant clauses.** Let $cl_1, cl_2, \ldots, cl_n$ be a sequence of clauses from $\mathrm{FF}(\mathcal{O})$ such that $\lambda^-(cl_n), \ldots, \lambda^-(cl_2), \lambda^-(cl_1)$ is the order in which the clauses are derived in the saturation of Γ_{var}. Let $\mathrm{FF}(\mathcal{O}) = N_0, N_1, \ldots, N_n$ be a sequence of clause sets such that $N_i = N_{i-1}$ if cl_i is *relevant* w.r.t. N_{i-1}, and $N_i = N_{i-1} \setminus \{cl_i\}$ if cl_i is *irrelevant* w.r.t. N_{i-1}, where cl is irrelevant w.r.t. N if $\lambda^-(cl)$ is derived by a inference rule ξ with substitution σ (excluding the decomposition rule) such that each premise p_i of ξ occurs in N and each variable occurring in $\lambda(p_i\sigma)$ also occurs in cl. $\mathrm{FF}_{\mathrm{R}}(\mathcal{O}) = N_n$ is called the *relevant subset* of $\mathrm{FF}(\mathcal{O})$.
4. **Converting to a DATALOG$^\vee$ program.** By $\mathcal{R}(\mathcal{O})$ we denote the DATALOG$^\vee$ program that contains the rule $R: \ A_1 \vee \ldots \vee A_n \leftarrow B_1, \ldots, B_m$ for each clause $A_1 \vee \ldots \vee A_n \vee \neg B_1 \vee \ldots \vee \neg B_m$ in $\mathrm{FF}_{\mathrm{R}}(\mathcal{O})$. $\mathrm{head}(R) = \{A_1, \ldots, A_n\}$ is called the *head* of R; $\mathrm{body}(R) = \{B_1, \ldots, B_m\}$ is called the *body* of R; $\mathcal{R}(\mathcal{O})$ is called the *repair program* of $\mathcal{O}$.

Example 4 (Example 3 continued). The repair program $\mathcal{R}(\mathcal{O})$ is constructed as follows. First, $\Xi(\mathcal{O})$ is separated into $\Xi_{var}(\mathcal{O}) = \{cl_4, cl_8, cl_9, cl_{10}, cl_{11}\}$ and $\Xi_{con}(\mathcal{O}) = \{cl_1, cl_2, cl_3, cl_5, cl_6, cl_7\}$. Second, $\Xi_{var}(\mathcal{O})$ is saturated by $\mathcal{BS}^+_{DL^+}$, yielding the following new clauses (the notation $\mathrm{R}(cl_i, cl_j)$ means that a clause is derived by resolving clauses cl_i and cl_j).

$$
\begin{array}{ll}
cl_{12}: \ S(x, f(x)) \vee \neg A(x) \vee \hbar_{ax_7} \vee \hbar_{ax_9}. & \mathrm{R}(cl_8, cl_{11}) \\
cl_{13}: \ \neg A(x) \vee \neg B(f(x)) \vee \hbar_{ax_7} \vee \hbar_{ax_8} \vee \hbar_{ax_9}. & \mathrm{R}(cl_{12}, cl_{10}) \\
cl_{14}: \ \neg A(x) \vee \hbar_{ax_7} \vee \hbar_{ax_8} \vee \hbar_{ax_9}. & \mathrm{R}(cl_{13}, cl_9)
\end{array}
$$

Third, the clauses containing function symbols are mapped to function-free clauses given below, where each mapped clause is associated with the original sequence number.

$$
\begin{array}{l}
cl_8: \ \neg S_f(x, x_f) \vee R(x, x_f) \vee \neg A(x) \vee \hbar_{ax_7}. \\
cl_9: \ \neg S_f(x, x_f) \vee B(x_f) \vee \neg A(x) \vee \hbar_{ax_7}. \\
cl_{12}: \ \neg S_f(x, x_f) \vee S(x, x_f) \vee \neg A(x) \vee \hbar_{ax_7} \vee \hbar_{ax_9}. \\
cl_{13}: \ \neg S_f(x, x_f) \vee \neg A(x) \vee \neg B(x_f) \vee \hbar_{ax_7} \vee \hbar_{ax_8} \vee \hbar_{ax_9}.
\end{array}
$$

Fourth, cl_{13} and cl_{12} are in turn detected to be irrelevant and removed. Finally, remaining clauses, together with ground clauses instantiated from $S_f(x, x_f)$, are translated into rules given below.

$$
\begin{array}{l}
R_1: \ A(a) \vee \hbar_{ax_1}. \quad R_2: \ B(b) \vee \hbar_{ax_2}. \quad R_3: \ Q_1(a) \vee \hbar_{ax_3}. \\
R_4: \ y_1 \approx y_2 \vee \hbar_{ax_3} \leftarrow Q_1(x), T(x, y_1), T(x, y_2). \\
R_5: \ T(a, b) \vee \hbar_{ax_4}. \quad R_6: \ T(a, c) \vee \hbar_{ax_5}. \quad R_7: \ \hbar_{ax_6} \leftarrow b \approx c. \\
R_8: \ R(x, x_f) \vee \hbar_{ax_7} \leftarrow A(x), S_f(x, x_f). \quad R_9: \ B(x_f) \vee \hbar_{ax_7} \leftarrow A(x), S_f(x, x_f). \\
R_{10}: \ \hbar_{ax_8} \leftarrow A(x), S(x, y), B(y). \quad R_{11}: \ S(x, y) \vee \hbar_{ax_9} \leftarrow R(x, y). \\
R_{12}: \ \hbar_{ax_7} \vee \hbar_{ax_8} \vee \hbar_{ax_9} \leftarrow A(x). \\
R_{13}: \ S_f(a, a_f). \quad R_{14}: \ S_f(b, b_f). \quad R_{15}: \ S_f(c, c_f).
\end{array}
$$

Hence, $\mathcal{R}(\mathcal{O}) = \{R_1, \ldots, R_{15}\}$. □

The following theorem shows the equisatisfiability between $\mathcal{O} \downarrow \phi_X$ and $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$ for an arbitrary truth assignment ϕ_X on X. Note that when $\mathcal{O}$ is a stratified ontology, $\mathcal{R}(\mathcal{O})$ is not stratified according to the stratification of $\mathcal{O}$. Instead, we associate the stratification of $\mathcal{O}$ with the stratification of decision atoms (see Theorem 3).

Theorem 2. *Let $\mathcal{O}$ be a $\mathcal{SHIQ}$ ontology and X the set of decision atoms.*

1. *(i) If $\mathcal{O} \setminus D$ is satisfiable for some set D of axioms, then $\mathcal{R}(\mathcal{O})$ has a model M such that $M \cap X = \{\hbar_{ax} | ax \in D\}$; (ii) If $\mathcal{R}(\mathcal{O})$ has a model M, then $\mathcal{O} \setminus \{ax | \hbar_{ax} \in M \cap X\}$ is satisfiable;*
2. *The number of atoms in each rule in $\mathcal{R}(\mathcal{O})$ is at most polynomial, the number of rules in $\mathcal{R}(\mathcal{O})$ is at most exponential, and $\mathcal{R}(\mathcal{O})$ can be computed in time exponential in* size$(\mathcal{O})$*, for unary coding of numbers in input.*

Proof Sketch. (1) We can show by (a) to (e) given below and Lemma 2 that, for any truth assignment ϕ_X on X, $\mathcal{O} \downarrow \phi_X$ is unsatisfiable iff $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$ is unsatisfiable. Hence (i) and (ii) can be proved as follows: (i) If $\mathcal{O} \setminus D$ is satisfiable for some set D of axioms, $\mathcal{O} \downarrow \phi_X$ is satisfiable for the truth assignment ϕ_X on X, such that for each axiom $ax \in \mathcal{O}$, $\phi_X(\hbar_{ax}) = 1$ iff $ax \in D$. So $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$ is satisfiable and thus has a model M. Then, $M' = M \cup \{\hbar_{ax} | ax \in D\}$ is a model of $\mathcal{R}(\mathcal{O})$ such that $M' \cap X = \{\hbar_{ax} | ax \in D\}$. (ii) If $\mathcal{R}(\mathcal{O})$ has a model M, $\mathcal{R}(\mathcal{O}) \downarrow \phi_X$ is satisfiable for the truth assignment ϕ_X on X, such that for each axiom $ax \in \mathcal{O}$, $\phi_X(\hbar_{ax}) = 1$ iff $\hbar_{ax} \in M \cap X$. So $\mathcal{O} \downarrow \phi_X$ is satisfiable. It follows that $\mathcal{O} \setminus \{ax | \hbar_{ax} \in M \cap X\}$ is satisfiable.

(a) $\Xi(\mathcal{O}) \downarrow \phi_X$ is unsatisfiable iff $(\Gamma_{var} \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ is unsatisfiable, because the predicates $Q_{R,f}$ do not occur in $\Xi(\mathcal{O}) \downarrow \phi_X$.

(b) Let $\mathrm{SAT}(\Gamma_{var})$ denote a saturated set of Γ_{var} by $\mathcal{BS}^+_{DL^+}$. For any $\mathcal{BS}^+_{DL}$ inference from closures in $\mathrm{SAT}(\Gamma_{var}) \downarrow \phi_X$ with conclusion cl, there must exist a $\mathcal{BS}^+_{DL^+}$ inference from closures in $\mathrm{SAT}(\Gamma_{var})$ with conclusion cl' such that cl' is redundant and cl' is of the form $cl \vee \hbar_{ax_1} \vee \ldots \vee \hbar_{ax_m}$ with $\phi_X(\hbar_{ax_i}) = 0$ for $i = 1, \ldots, m$. Note that the set S of closures in $\mathrm{SAT}(\Gamma_{var})$ making cl' redundant contains no decision atoms not occurring in cl'. So $S' = S \downarrow \phi_X$ is actually the set of closures obtained from S by deleting all decision atoms. It follows that $S' \subseteq \mathrm{SAT}(\Gamma_{var}) \downarrow \phi_X$ makes cl redundant. Hence, $\mathrm{SAT}(\Gamma_{var}) \downarrow \phi_X$ is a saturated set of $\Gamma_{var} \downarrow \phi_X$ by $\mathcal{BS}^+_{DL}$. Since $\mathcal{BS}^+_{DL}$ is sound and complete, $(\Gamma_{var} \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ is unsatisfiable iff $(\mathrm{SAT}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ is unsatisfiable.

(c) Since $\mathrm{SAT}(\Gamma_{var}) \downarrow \phi_X$ already contains all nonground consequences of $(\Gamma_{var} \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$, according to the syntactic form of $\mathcal{ALCHIQ}$-closures, only ground closures of type 8 are derived in the saturation of $(\Gamma_{var} \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ by $\mathcal{BS}^+_{DL}$. Furthermore, closures of types 4 and 6 cannot participate in any $\mathcal{BS}^+_{DL}$ inference with a ground closure, so they can be removed from $\mathrm{SAT}(\Gamma_{var}) \downarrow \phi_X$ without affecting the satisfiability of $(\mathrm{SAT}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$. Recall that $\mathrm{SAT}_{\mathrm{R}}(\Gamma_{var}) \downarrow \phi_X$ is obtained from $\mathrm{SAT}(\Gamma_{var}) \downarrow \phi_X$ by deleting $\mathcal{ALCHIQ}$-closures of types 4 and 6. So $(\mathrm{SAT}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ is unsatisfiable iff

$(\mathrm{SAT_R}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ is unsatisfiable.

(d) By considering all inferences of $\mathcal{BS}^+_{DL}$ analogously as in Lemma 7.2.4 [21], we can see that for each closure cl derived in the saturation of $(\mathrm{SAT_R}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$, $\lambda(cl)$ can be derived from $\mathrm{FF}(\mathcal{O}) \downarrow \phi_X$. Hence, if the empty closure is derived by saturating $(\mathrm{SAT_R}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ by $\mathcal{BS}^+_{DL}$, it can also be derived by saturating $\mathrm{FF}(\mathcal{O}) \downarrow \phi_X$ by $\mathcal{BS}^+_{DL}$. In a similar way we can show that if a closure cl is derivable by saturating $\mathrm{FF}(\mathcal{O}) \downarrow \phi_X$ by $\mathcal{BS}^+_{DL}$, the closure $\lambda^-(cl)$ can be derived from $(\mathrm{SAT_R}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$. Hence, $(\mathrm{SAT_R}(\Gamma_{var}) \cup \Xi_{con}(\mathcal{O})) \downarrow \phi_X$ is unsatisfiable iff $\mathrm{FF}(\mathcal{O}) \downarrow \phi_X$ is unsatisfiable.

(e) We show that $\mathrm{FF}(\mathcal{O}) \downarrow \phi_X$ is unsatisfiable iff $\mathrm{FF_R}(\mathcal{O}) \downarrow \phi_X$ is unsatisfiable. The ($\Leftarrow$) direction is trivial, because $\mathrm{FF_R}(\mathcal{O}) \subseteq \mathrm{FF}(\mathcal{O})$. For the ($\Rightarrow$) direction, let $N_0 = \mathrm{FF}(\mathcal{O}), N_1, \ldots, N_n = \mathrm{FF_R}(\mathcal{O})$ be a sequence of clause sets obtained in course of deleting irrelevant clauses, and cl be the irrelevant clause w.r.t. $N_{i-1} \downarrow \phi_X$ such that $\lambda^-(cl)$ is derived from premises $p_1, \ldots, p_k$ in $N_{i-1} \downarrow \phi_X$ by an inference ξ with a substitution σ. We show that if $N_i \downarrow \phi_X$ has a model M, M is also a model of $N_{i-1} \downarrow \phi_X$. Then by induction on i, $N_0 \downarrow \phi_X$ is satisfiable if $N_n \downarrow \phi_X$ is satisfiable. Consider each ground substitute τ of cl. Clauses $p_1, \ldots, p_k$ can be of type 1,2,3,5 or 7, so σ can contain only mappings of the form $x \mapsto x'$, $x \mapsto f(x')$, or $y_i \mapsto f(x')$. The set of variables in $\lambda(p_j\sigma)$ and $\lambda(p_j)\lambda(\sigma)$ obviously coincide. Since cl is irrelevant, all variables occurring in each $\lambda(p_j\sigma)$ occur in cl as well, so τ instantiates all variables in each $\lambda(p_j)\lambda(\sigma)$. Furthermore, $\lambda(p_j)\lambda(\sigma)\tau \subseteq \lambda(p_j\sigma)\tau$ for any j. If the inclusion is strict, then the latter clause contains literals of the form $\neg S_f(a, a_f)$ that do not occur in the former one, because σ instantiates a variable from p_j to a term $f(x')$ which comes from another premise p_l. Then, $\lambda(p_l)$ contains the literal $\neg S_f(x', x'_f)$, so $\lambda(p_l)\lambda(\sigma)\tau$ contains $\neg S_f(a, a_f)$. Hence, $\{\lambda(p_j)\lambda(\sigma)\tau\}_{1 \leq j \leq k}$ can participate in a ground inference corresponding to ξ whose conclusion is $cl\ \tau$. Since $\lambda(p_1), \ldots, \lambda(p_k)$ occur in $N_i \downarrow \phi_X$, $\lambda(p_1)\lambda(\sigma)\tau, \ldots, \lambda(p_k)\lambda(\sigma)\tau$ are satisfied by M, so $cl\ \tau$ is satisfied by M too. It follows that M is a model of $N_{i-1} \downarrow \phi_X$.

(2) By the proof of Lemma 4, for each clause $cl \in \mathrm{SAT_R}(\Gamma_{var})$, the number of literals in cl is at most polynomial in size($\mathcal{O}$), and the number of clauses $|\mathrm{SAT_R}(\Gamma_{var})|$ is at most exponential in size($\mathcal{O}$). The number of constants a_f added to $\mathcal{R}(\mathcal{O})$ equals $c \cdot f$, where c is the number of constants and f the number of function symbols occurring in $\Xi(\mathcal{O})$. Under the assumption that numbers are coded in unary, both c and f are polynomial in size($\mathcal{O}$), so is the number of constants a_f. By Lemma 4, $\mathrm{SAT_R}(\Gamma_{var})$ can be computed in time exponential in size($\mathcal{O}$). In addition, computing $\mathrm{FF}(\mathcal{O})$ and its relevant subset $\mathrm{FF_R}(\mathcal{O})$ from $\mathrm{SAT_R}(\Gamma_{var})$ can be performed in polynomial time in $|\mathrm{SAT_R}(\Gamma_{var})| + \text{size}(\mathcal{O})$. Hence, the number of atoms in each rule in $\mathcal{R}(\mathcal{O})$ is at most polynomial, the number of rules in $\mathcal{R}(\mathcal{O})$ is at most exponential, and $\mathcal{R}(\mathcal{O})$ can be computed in time exponential in size($\mathcal{O}$). □

5.2 Checking the Basic Consequences

To decide if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ for an atomic concept membership axiom $A(a)$, we establish, by Claim 1 of Theorem 2, a correspondence between X-lex-minimal models of the repair program $\mathcal{R}(\mathcal{O})$ and lex-maximal consistent subontologies of $\mathcal{O}$ (see Theorem 3). This correspondence implies that the problem of deciding if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ can be reduced to a certain problem of cautious reasoning over $\mathcal{R}(\mathcal{O})$ (see Theorem 4), solved by a set of satisfiability tests (see Theorem 5).

Definition 6. For a logic program P and a stratification $X = (X_1, \ldots, X_n)$ of some ground atoms occurring in P, a model M of P is called an *X-lex-minimal model* of P if for any model M' of P, $(M \cap X_1, \ldots, M \cap X_n) \leq^{lex} (M' \cap X_1, \ldots, M' \cap X_n)$.

Theorem 3. *Let $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ be a stratified $\mathcal{SHIQ}$ ontology, and $X = (X_1, \ldots, X_n)$ be the corresponding stratification of decision atoms in $\mathcal{R}(\mathcal{O})$, i.e., $X_i = \{\hbar_{ax} \mid ax \in \mathcal{O}_i\}$ for $i = 1, \ldots, n$. Then: (1) If M is an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$, then $\mathcal{O} \setminus \{ax \mid \hbar_{ax} \in M \cap X\}$ is a lex-maximal consistent subontology of $\mathcal{O}$; (2) If $\mathcal{O}' = (\mathcal{O}'_1, \ldots, \mathcal{O}'_n)$ is a lex-maximal consistent subontology of $\mathcal{O}$, then $\mathcal{R}(\mathcal{O})$ has an X-lex-minimal model M such that for each i, $M \cap X_i = \{\hbar_{ax} \mid ax \in \mathcal{O}_i \setminus \mathcal{O}'_i\}$.*

Proof. (1) Let M be an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$. By Theorem 2, $\mathcal{O}' = \mathcal{O} \setminus \{ax \mid \hbar_{ax} \in M \cap X\}$ is satisfiable. Suppose $\mathcal{O}'$ is not lex-maximal consistent, i.e., there exists a consistent subontology $\mathcal{O}'' = (\mathcal{O}''_1, \ldots, \mathcal{O}''_n)$ such that $\mathcal{O}' <^{lex} \mathcal{O}''$. By Theorem 2, $\mathcal{R}(\mathcal{O})$ has a model M' such that $M' \cap X_i = \{\hbar_{ax} | ax \in \mathcal{O}_i \setminus \mathcal{O}''_i\}$ for each i. But $(M' \cap X_1, \ldots, M' \cap X_n) <^{lex} (M \cap X_1, \ldots, M \cap X_n)$, contradicting that M is an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$. So $\mathcal{O}'$ is a lex-maximal consistent subontology of $\mathcal{O}$.

(2) Let $\mathcal{O}' = (\mathcal{O}'_1, \ldots, \mathcal{O}'_n)$ be a lex-maximal consistent subontology of $\mathcal{O}$. By Theorem 2, $\mathcal{R}(\mathcal{O})$ has a model M such that $M \cap X_i = \{\hbar_{ax} | ax \in \mathcal{O}_i \setminus \mathcal{O}'_i\}$ for each i. Suppose M is not an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$, i.e., there exists a model M' of $\mathcal{R}(\mathcal{O})$ such that $(M' \cap X_1, \ldots, M' \cap X_n) <^{lex} (M \cap X_1, \ldots, M \cap X_n)$. By Theorem 2, $\mathcal{O}'' = \mathcal{O} \setminus \{ax \mid \hbar_{ax} \in M' \cap X\}$ is satisfiable. But $\mathcal{O}' <^{lex} \mathcal{O}''$, contradicting that $\mathcal{O}'$ is lex-maximal consistent. So M is an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$. □

Theorem 4. *$\mathcal{O} \vdash^{lex}_{cons} A(a)$ iff $A(a)$ is in all X-lex-minimal models of $\mathcal{R}(\mathcal{O})$.*

Proof. Suppose $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$. We assume that the constant a occurs in $\mathcal{O}$, otherwise this theorem trivially holds.

($\Rightarrow$) Suppose $\mathcal{O} \vdash^{lex}_{cons} A(a)$ and there exists an X-lex-minimal model M of $\mathcal{R}(\mathcal{O})$ such that $A(a) \notin M$. Then M is also an X-lex-minimal model of $\mathcal{R}(\mathcal{O}) \cup \{\leftarrow A(a)\}$. Let $\mathcal{O}_{ext} = (\{\neg A(a)\}, \mathcal{O}_1, \ldots, \mathcal{O}_n)$ and $X_{ext} = (\{\hbar_{\neg A(a)}\}, X_1, \ldots, X_n)$. Then M is an X_{ext}-lex-minimal model of $\mathcal{R}(\mathcal{O}_{ext}) = \mathcal{R}(\mathcal{O}) \cup \{\hbar_{\neg A(a)} \leftarrow A(a)\}$. By Theorem 3, $\mathcal{O}'_{ext} = \mathcal{O}_{ext} \setminus \{ax \mid \hbar_{ax} \in M \cap X\}$ is a lex-maximal consistent subontology of $\mathcal{O}_{ext}$. Let $\mathcal{O}' = (\mathcal{O}'_{ext} \cap \mathcal{O}_1, \ldots, \mathcal{O}'_{ext} \cap \mathcal{O}_n)$. Then $\mathcal{O}'$ is a lex-maximal consistent subontology of $\mathcal{O}$, otherwise by Theorem 2, M cannot be an

X-lex-minimal model of $\mathcal{R}(\mathcal{O})$. Since $\mathcal{O}'_{ext}$ is consistent and $\neg A(a) \in \mathcal{O}'_{ext}$, we have $\mathcal{O}' \not\models A(a)$, contradicting that $\mathcal{O} \vdash^{lex}_{cons} A(a)$.

($\Leftarrow$) Suppose $A(a)$ is in all X-lex-minimal models of $\mathcal{R}(\mathcal{O})$ and there exists a lex-maximal consistent subontology $\mathcal{O}' = (\mathcal{O}'_1, \ldots, \mathcal{O}'_n)$ of $\mathcal{O}$ such that $\mathcal{O}' \not\models A(a)$. Let $\mathcal{O}_{ext} = (\{\neg A(a)\}, \mathcal{O}_1, \ldots, \mathcal{O}_n)$ and $X_{ext} = (\{\hbar_{\neg A(a)}\}, X_1, \ldots, X_n)$. Then $\mathcal{O}'_{ext} = (\{\neg A(a)\}, \mathcal{O}'_1, \ldots, \mathcal{O}'_n)$ is a lex-maximal consistent subontology of $\mathcal{O}_{ext}$. By Theorem 3, there exists an X_{ext}-lex-minimal model M of $\mathcal{R}(\mathcal{O}_{ext})$ such that $\hbar_{\neg A(a)} \notin M$ and $M \cap X_i = \{\hbar_{ax} | ax \in \mathcal{O}_i \setminus \mathcal{O}'_i\}$ for each $i > 0$. Then M is also an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$, otherwise by Theorem 2, $\mathcal{O}'$ cannot be lex-maximal consistent. Since $\hbar_{\neg A(a)} \notin M$ and $\mathcal{R}(\mathcal{O}_{ext}) = \mathcal{R}(\mathcal{O}) \cup \{\hbar_{\neg A(a)} \leftarrow A(a)\}$, M is also an X-lex-minimal model of $\mathcal{R}(\mathcal{O}) \cup \{\leftarrow A(a)\}$. But then $A(a) \notin M$ contracts that $A(a)$ is in all X-lex-minimal models of $\mathcal{R}(\mathcal{O})$. □

Suppose $X = (X_1, \ldots, X_n)$. It is easy to see that $(|M \cap X_1|, \ldots, |M \cap X_n|)$ is the same for every X-lex-minimal model M of $\mathcal{R}(\mathcal{O})$. By $\mathrm{lmw}(\mathcal{R}(\mathcal{O}), X) = (|M \cap X_1|, \ldots, |M \cap X_n|)$ we denote the unique *lex-minimal weight vector* of every X-lex-minimal model M of $\mathcal{R}(\mathcal{O})$, and by $\mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X)$ we denote the i^{th} element $|M \cap X_i|$ in $\mathrm{lmw}(\mathcal{R}(\mathcal{O}), X)$. Note that an interpretation is an X-lex-minimal model of $\mathcal{R}(\mathcal{O})$ iff it is a model of $\mathcal{R}(\mathcal{O}) \cup \{\sum_{\hbar_{ax} \in X_i} \mathrm{assign}(\hbar_{ax}) \leq \mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X) \mid 1 \leq i \leq n\}$, where $\mathrm{assign}(\alpha)$ denotes the 0-1 truth value of α. So Theorem 4 can be reformulated into the following theorem.

Theorem 5. *$\mathcal{O} \vdash^{lex}_{cons} A(a)$ iff $\mathcal{R}(\mathcal{O}) \cup \{\sum_{\hbar_{ax} \in X_i} \mathrm{assign}(\hbar_{ax}) \leq \mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X) \mid 1 \leq i \leq n\} \cup \{\leftarrow A(a)\}$ is unsatisfiable.* □

In order to apply Theorem 5 to check if $\mathcal{O} \vdash^{lex}_{cons} A(a)$, we need to compute $\mathrm{lmw}(\mathcal{R}(\mathcal{O}), X)$. By Definition 6, it can be seen that $\mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X)$ is the minimum number v such that $P_i(v)$ is satisfiable, where

$$P_i(v) = \mathcal{R}(\mathcal{O}) \cup \{ \sum_{\hbar_{ax} \in X_j} \mathrm{assign}(\hbar_{ax}) \leq \mathrm{lmw}_j(\mathcal{R}(\mathcal{O}), X) \mid 1 \leq j < i\} \cup \quad (1)$$

$$\{ \sum_{\hbar_{ax} \in X_i} \mathrm{assign}(\hbar_{ax}) \leq v\} \cup \{\mathrm{assign}(\hbar_{ax}) = 0 \mid \hbar_{ax} \in \bigcup_{j=i+1}^{n} X_j\}.$$

I.e., $\mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X)$ is defined based on $\mathrm{lmw}_j(\mathcal{R}(\mathcal{O}), X)$ for all $j < i$, and can be computed one by one from $i = 1$ to n.

In order to check the satisfiability of a logic program of the form in Theorem 5 or Formula (1), we need to handle *Pseudo-Boolean constraints* (PB-constraints) of the form $\sum_i c_i \cdot \mathrm{assign}(x_i) \leq d$ with constants $c_i, d \in \mathbf{Z}$ and variables $x_i \in \{0, 1\}$, where $\mathbf{Z}$ denotes the integer domain. There has been well study on handling PB-constraints; especially, powerful SAT solvers that support PB-constraints competed each other in an annual competition[8]. A satisfiability problem with PB-constraints can be either solved by standard SAT solvers after translating PB-constraints to SAT clauses [7], or solved by extended SAT solvers that support PB-constraints natively, such as PUEBLO [31].

[8] `http://www.cril.univ-artois.fr/PB07/`

There are some issues for applying SAT solvers that support PB-constraints: they work on propositional programs only and do not distinguish equational atoms (of the form $a \approx b$) from other atoms. To treat the equality predicate $\approx$, which is interpreted as a *congruence relation* in $\mathcal{SHIQ}$, as an ordinary predicate, we use a well-known transformation from [10]. For a DATALOG$^\vee$ program P, let $P_\approx$ denote the logic program consisting of the rules stating that the equality predicate is *reflexive*, *symmetric* and *transitive*, as well as the *replacement rules* of the form "$T(X_1, \ldots, Y_i, \ldots, X_n) \leftarrow T(X_1, \ldots, X_i, \ldots, X_n), X_i \approx Y_i$", instantiated for each predicate T in P (excluding $\approx$) and each position i. Then, appending $P_\approx$ to P allows to treat $\approx$ as an ordinary predicate. It should be noted that the reflexive rule is not safe, so it is instead represented as a set of ground facts of the form $a \approx a$, instantiated for each constant a in P.

To ground $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_\approx$, we apply the disk-based grounding (DBG) technique proposed in [6]. The DBG technique extends the well-known *intelligent grounding* technique [9] for DATALOG$^\vee$ programs by pruning instantiated rules that are trivially satisfied by every minimal model. Basically, the DBG technique iteratively instantiates each rule in the input program by using derivable ground atoms (i.e. ground atoms occurring in the input program or head atoms in instantiated rules), until there is no more ground atom that can be derived. Note that if some head atom of an instantiated rule r is in the unique minimal model M_{def} of the *definite fragment* of the input program (i.e. the set of rules with single head atoms), r will be pruned and its head atoms that are not previously derived are not used to instantiate rules. In this work, $M_{def} = \mathrm{FF}_{Succ}(\mathcal{O}) \cup \mathrm{FF}_{HU}(\mathcal{O}) \cup \{a \approx a \mid a \text{ occurs in } \mathcal{R}(\mathcal{O})\}$. In the DBG technique, derivable ground atoms are maintained in a SQL database and new ground instances of rules are retrieved via SQL queries. By $\mathrm{GR}'(\mathcal{O})$ we denote the program grounded from $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_\approx$ by using the DBG technique. Then M is a minimal model of $\mathrm{GR}'(\mathcal{O})$ iff $M \cup M_{def}$ is a minimal model of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_\approx$.

We assume that there is a total ordering $\succ$ on all constants occurring in $\mathrm{GR}'(\mathcal{O})$. By $\mathrm{GR}(\mathcal{O})$ we denote the program obtained from $\mathrm{GR}'(\mathcal{O})$ by rewriting each equality atom $a \approx b$ such that $b \succ a$ into $b \approx a$ and then deleting all rewritten rules that are duplicate or of the form $a \approx b \leftarrow a \approx b$. Obviously, there exists a bijection between the set of minimal models of $\mathrm{GR}'(\mathcal{O})$ and the set of minimal models of $\mathrm{GR}(\mathcal{O})$; the difference between two corresponding minimal models consists of equality atoms $a \approx b$ such that $b \succ a$. The following corollary is an immediate consequence that follows from Theorem 4 and Theorem 5, where $\mathrm{lmw}_i(\mathrm{GR}(\mathcal{O}), X) = \mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X)$ for each i.

Corollary 1. *$\mathcal{O} \vdash^{lex}_{cons} A(a)$ iff $A(a)$ is in all X-lex-minimal models of* $\mathrm{GR}(\mathcal{O})$*, i.e.,* $\mathrm{GR}(\mathcal{O}) \cup \{\sum_{\hbar_{ax} \in X_i} \mathrm{assign}(\hbar_{ax}) \leq \mathrm{lmw}_i(\mathrm{GR}(\mathcal{O}), X) \mid 1 \leq i \leq n\} \cup \{\leftarrow A(a)\}$ *is unsatisfiable.*

Proof. It is sufficient to show that $A(a)$ is in all X-lex-minimal models of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_\approx$ iff $A(a)$ is in all X-lex-minimal models of $\mathrm{GR}'(\mathcal{O})$, where the equality predicate $\approx$ is treated as an ordinary predicate in both logic programs.

($\Rightarrow$) Let M is an arbitrary X-lex-minimal model of $\mathrm{GR}'(\mathcal{O})$. Then there exits a minimal model M' of $\mathrm{GR}'(\mathcal{O})$ such that $M' \subseteq M$ and $M' \cap X = M \cap$

X, i.e., M' is also an X-lex-minimal model of $\mathrm{GR}'(\mathcal{O})$. Since $M' \cup M_{def}$ is a minimal model of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$, $M' \cup M_{def}$ must be an X-lex-minimal model of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$, otherwise there exits a minimal model M'' of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$ such that $(M' \cup M_{def}) \cap X <^{lex} M'' \cap X$. $M'' \setminus M_{def}$ is also a minimal model of $\mathrm{GR}'(\mathcal{O})$, contradicting that M' is an X-lex-minimal model of $\mathrm{GR}'(\mathcal{O})$. Hence, $M' \cup M_{def}$ is an X-lex-minimal model of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$ and thus $A(a) \in M'$. Since $M' \subseteq M$, we have $A(a) \in M$ and thus $A(a)$ is in all X-lex-minimal models of $\mathrm{GR}'(\mathcal{O})$.

($\Leftarrow$) The proof is the same as above by exchanging the places of $\mathrm{GR}'(\mathcal{O})$ and $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$. □

Example 5 (Example 4 continued). The unique minimal model of the definite fragment of $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$ is $M_{def} = \{S(o, o_f) \mid o = a, b, c\} \cup \{o \approx o \mid o = a, b, c, a_f, b_f, c_f\}$. The initial set of ground atoms used to instantiate rules is $M_{def} \cup \{A(a), B(b), Q_1(a), T(a, b), T(a, c), b \approx c\}$. By iteratively instantiating rules using the DBG technique and then rewriting equality atoms, $\mathcal{R}(\mathcal{O}) \cup \mathcal{R}(\mathcal{O})_{\approx}$ is grounded to $\mathrm{GR}(\mathcal{O}) = \{r_1, \ldots, r_{16}\}$ given below, assuming a total ordering on constants $a \succ b \succ c \succ a_f \succ b_f \succ c_f$.

$r_1 :\ A(a) \vee \hbar_{ax_1}.$ $\quad r_2 :\ B(b) \vee \hbar_{ax_2}.$ $\quad r_3 :\ Q_1(a) \vee \hbar_{ax_3}.$
$r_4 :\ T(a, b) \vee \hbar_{ax_4}.$ $\quad r_5 :\ T(a, c) \vee \hbar_{ax_5}.$ $\quad r_6 :\ \hbar_{ax_6} \leftarrow b \approx c.$
$r_7 :\ b \approx c \vee \hbar_{ax_3} \leftarrow Q_1(a), T(a, b), T(a, c).$ $\quad r_8 :\ R(a, a_f) \vee \hbar_{ax_7} \leftarrow A(a).$
$r_9 :\ B(a_f) \vee \hbar_{ax_7} \leftarrow A(a).$ $\quad r_{10} :\ S(a, a_f) \vee \hbar_{ax_9} \leftarrow R(a, a_f).$
$r_{11} :\ \hbar_{ax_8} \leftarrow A(a), S(a, a_f), B(a_f).$ $\quad r_{12} :\ \hbar_{ax_7} \vee \hbar_{ax_8} \vee \hbar_{ax_9} \leftarrow A(a).$
$r_{13} :\ T(a, c) \leftarrow T(a, b), b \approx c.$ $\quad r_{14} :\ T(a, b) \leftarrow T(a, c), b \approx c.$
$r_{15} :\ B(c) \leftarrow B(b), b \approx c.$ $\quad r_{16} :\ B(b) \leftarrow B(c), b \approx c.$

It can be computed by Formula (1), where $\mathcal{R}(\mathcal{O})$ is replaced with $\mathrm{GR}(\mathcal{O})$, that $\mathrm{lmw}(\mathrm{GR}(\mathcal{O}), X) = (0, 1, 0, 1)$. Consider deciding if $\mathcal{O} \vdash^{lex}_{cons} A(a)$. The satisfiability of $\Pi = \mathrm{GR}(\mathcal{O}) \cup \{\mathrm{assign}(\hbar_{ax_1}) + \mathrm{assign}(\hbar_{ax_2}) + \mathrm{assign}(\hbar_{ax_3}) \leq 0, \mathrm{assign}(\hbar_{ax_4}) + \mathrm{assign}(\hbar_{ax_5}) + \mathrm{assign}(\hbar_{ax_6}) \leq 1, \mathrm{assign}(\hbar_{ax_7}) + \mathrm{assign}(\hbar_{ax_8}) \leq 0, \mathrm{assign}(\hbar_{ax_9}) \leq 1\} \cup \{\leftarrow A(a)\}$ is tested. It is easy to see that Π is unsatisfiable, so $\mathcal{O} \vdash^{lex}_{cons} A(a)$. □

Consider the time complexity for checking if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ in terms of *data complexity*, i.e. the complexity measured as a function of $|\mathcal{O}_{\mathcal{A}}|$. Since saturating the clause set $\Xi(\mathcal{O})$ is only performed over the subset transformed from terminological axioms, by viewing the considering ontology in Claim 2 of Theorem 2 as the terminology of $\mathcal{O}$ only, the saturation of $\Xi(\mathcal{O})$ is accomplished in constant time and the number of rules in $\mathcal{R}(\mathcal{O})$ is polynomial in $|\mathcal{O}_{\mathcal{A}}|$. For every rule $R \in \mathcal{R}(\mathcal{O})$, the number of variables in R is bounded by a constant, so there are at most polynomial ground instances of R in $\mathrm{GR}(\mathcal{O})$. It follows that the number of rules in $\mathrm{GR}(\mathcal{O})$ is polynomial in $|\mathcal{O}_{\mathcal{A}}|$. In addition, the computation of $\mathrm{lmw}(\mathrm{GR}(\mathcal{O}), X)$ needs at most $|X|$ satisfiability tests, so checking if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ is accomplished in $|X| + 1$ satisfiability tests over $\mathrm{GR}(\mathcal{O})$. Since the satisfiability problem with PB-constraints is NP-complete, the problem of

deciding if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ is thus in Δ^p_2 in data complexity. Note that the addressing problem is also Δ^p_2-hard (see the following theorem), so our proposed method is worst-case optimal in data complexity.

Theorem 6. *For a stratified $\mathcal{SHIQ}$ ontology $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ and an atomic concept membership axiom $A(a)$, the problem of deciding if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ is data complete for Δ^p_2.*

Proof. The Δ^p_2 membership has been proved by our proposed method. We now show the Δ^p_2 hardness by a polynomial time reduction from the following Δ^p_2-complete problem [17]: "given a satisfiable clause set $C = \{C_1, \ldots, C_m\}$ on $X = \{x_1, \ldots, x_n\}$, decide whether the lexicographically minimum truth assignment ϕ^m_X satisfying C fulfills $\phi^m_X(x_n) = 1$". W.l.o.g., we assume that each C_i consists of three literals. Under such assumption the hardness remains [32].

Given any instance I of the above problem, we transform it to an instance $I' = (\mathcal{O}, A(a_n))$ of the given problem: We set $\mathcal{O}_i = \{A(a_i)\}$ for each $i = 1, \ldots, n$ and $\mathcal{O}_0 = \{A \sqsubseteq \forall R.L_1, A \sqsubseteq \forall R.L_2, A \sqsubseteq \forall R.L_3, \top \sqsubseteq A \sqcup \forall S.L_1, \top \sqsubseteq A \sqcup \forall S.L_2, \top \sqsubseteq A \sqcup \forall S.L_3, L_1 \sqcap L_2 \sqcap L_3 \sqsubseteq \bot\}$. Then, for each clause $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ in C and each $j \in \{1, 2, 3\}$, if $l_{i,j} = x_k$ for some k, we append $R(a_k, c_i)$ to $\mathcal{O}_0$; otherwise if $l_{i,j} = \neg\, x_k$ for some k, we append $S(a_k, c_i)$ to $\mathcal{O}_0$. Let $\mathcal{O} = \{\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_n\}$, and $r(\mathcal{O}, \phi_X)$ denote $\mathcal{O} \setminus \{A(a_i) \mid 1 \leq i \leq n, \phi_X(x_i) = 1\}$ for an truth assignment ϕ_X on X.

We show that for any truth assignment ϕ_X on X satisfying C, $r(\mathcal{O}, \phi_X)$ is consistent. For any truth assignment ϕ_X on X, there must be an interpretation satisfying all axioms in $r(\mathcal{O}, \phi_X)$ except $L_1 \sqcap L_2 \sqcap L_3 \sqsubseteq \bot$. Let ϕ_X be a truth assignment that satisfies C and M a *minimal* interpretation that satisfies all axioms in $r(\mathcal{O}, \phi_X)$ except $L_1 \sqcap L_2 \sqcap L_3 \sqsubseteq \bot$. Suppose M does not satisfy $L_1 \sqcap L_2 \sqcap L_3 \sqsubseteq \bot$. Then there exists some $i \in \{1, \ldots, m\}$ such that $\{L_1(c_i), L_2(c_i), L_3(c_i)\} \subseteq M$. Consider each $j = 1, 2, 3$. In case $l_{i,j} = x_k$ for some k, since $\{R(a_k, c_i), L_j(c_i)\} \subseteq M$, we have $A(a_k) \in M$, otherwise $M \setminus \{L_j(c_i)\}$ satisfies all axioms in $r(\mathcal{O}, \phi_X)$ except $L_1 \sqcap L_2 \sqcap L_3 \sqsubseteq \bot$, contradicting that M is minimal. Thus $A(a_k) \in r(\mathcal{O}, \phi_X)$, i.e., $\phi_X(x_k) = 0$. In case $l_{i,j} = \neg\, x_k$ for some k, since $\{S(a_k, c_i), L_j(c_i)\} \subseteq M$, we have $A(a_k) \notin M$, otherwise $M \setminus \{A(a_k)\}$ satisfies all axioms in $r(\mathcal{O}, \phi_X)$ except $L_1 \sqcap L_2 \sqcap L_3 \sqsubseteq \bot$, contradicting that M is minimal. Thus $A(a_k) \notin r(\mathcal{O}, \phi_X)$, i.e., $\phi_X(x_k) = 1$. It follows that $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ is not satisfied by ϕ_X, contradicting that C is satisfied by ϕ_X. So $r(\mathcal{O}, \phi_X)$ is satisfiable by M and thus is consistent.

We show that for any consistent subontology $\mathcal{O}'$ of $\mathcal{O}$ such that $\mathcal{O}_0 \subseteq \mathcal{O}'$, the truth assignment ϕ_X on X such that $\phi_X(x_i) = 0$ iff $A(a_i) \in \mathcal{O}'$ satisfies C. Let M be a model of $\mathcal{O}'$. Then for each $i \in \{1, \ldots, m\}$, there exists some j such that $L_j(c_i) \notin M$. In case $l_{i,j} = x_k$ for some k, since $R(a_k, c_i) \in M$ and $M \models A \sqsubseteq \forall R.L_j$, we have $A(a_k) \notin M$ and thus $A(a_k) \notin \mathcal{O}'$, i.e., $\phi_X(x_k) = 1$. In case $l_{i,j} = \neg\, x_k$ for some k, since $S(a_k, c_i) \in M$ and $M \models \top \sqsubseteq A \sqcup \forall S.L_j$, we have $A(a_k) \in M$ and thus $A(a_k) \in \mathcal{O}'$, i.e., $\phi_X(x_k) = 0$. It follows that C_i is satisfied by ϕ_X and thus C is satisfied by ϕ_X.

As a result, ϕ^m_X is the lexicographically minimum truth assignment satisfying C iff $r(\mathcal{O}, \phi^m_X)$ is the unique lex-maximal consistent subontology of $\mathcal{O}$. It follows

that $\phi^m_X(x_n) = 0$ iff $\mathcal{O} \vdash^{lex}_{cons} A(a_n)$, i.e., I is false iff I' is true. In addition, I' is constructed in polynomial time in n and m; the number of terminological axioms in $\mathcal{O}$ is fixed; the number of assertional axioms in $\mathcal{O}$ is polynomial in n and m. So the given problem is Δ^p_2-hard in data complexity. □

5.3 Partition-Based Optimizations

To make the above method more scalable, we adapt the partitioning technique in [6] to decompose $\mathrm{GR}(\mathcal{O})$ into disjoint subprograms before checking if $\mathcal{O} \vdash^{lex}_{cons} A(a)$. By that means, the satisfiability test in Corollary 1 can be performed over a small relevant part of $\mathrm{GR}(\mathcal{O})$.

Let $\mathrm{atoms}(P)$ denote the set of ground atoms in a propositional program P. Given $\mathrm{GR}(\mathcal{O})$ and the set X of decision atoms occurring in $\mathrm{GR}(\mathcal{O})$, the partitioning algorithm in [6] (i.e. Algorithm 1) decomposes $\mathrm{GR}(\mathcal{O})$ into a set of disjoint subprograms $\{\mathrm{GR}_i(\mathcal{O})\}_{1 \le i \le p}$. The decomposition is through sequentially accessing $\mathrm{GR}(\mathcal{O})$ at most $\min(|\mathrm{GR}(\mathcal{O})|, |\mathrm{atoms}(\mathrm{GR}(\mathcal{O}))|)$ times, where $|\mathrm{GR}(\mathcal{O})|$ denotes the number of rules in $\mathrm{GR}(\mathcal{O})$ and $|\mathrm{atoms}(\mathrm{GR}(\mathcal{O}))|$ denotes the number of atoms in $\mathrm{GR}(\mathcal{O})$. The resulting subprograms have the following properties: (1) $\mathrm{GR}_j(\mathcal{O})$ and $\mathrm{GR}_k(\mathcal{O})$ have no common ground atoms for any $j \neq k$ (by lines 12,13); (2) each $\mathrm{GR}_i(\mathcal{O})$ is a connected component (by lines 3–13), i.e., for any two ground atoms α_1 and α_2 occurring in $\mathrm{GR}_i(\mathcal{O})$, there exists a sequence of ground atoms $\beta_1 = \alpha_1, \ldots, \beta_n = \alpha_2$ such that β_k and β_{k+1} occur together in some rule in $\mathrm{GR}_i(\mathcal{O})$ for $k = 1, \ldots, n-1$; (3) each $\mathrm{GR}_i(\mathcal{O})$ is a certain rule closure (by lines 3–11), i.e., for each rule $r \in \mathrm{GR}(\mathcal{O})$ such that $\emptyset \subset \mathrm{head}(r) \setminus X \subseteq \mathrm{atoms}(\mathrm{GR}_i(\mathcal{O}) \setminus \{r\})$, $r \in \mathrm{GR}_i(\mathcal{O})$; and (4) for each rule $r \in \mathrm{GR}(\mathcal{O}) \setminus \bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O})$, there exists some ground atom $\alpha \in \mathrm{head}(r)$ such that $\alpha \notin \mathrm{atoms}(\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O})) \cup X$ (by lines 3–11). Such partitioning on rules also divides X into disjoint subsets. Let $X^\dagger_i$ denote the subset of decision atoms that occur in $\mathrm{GR}_i(\mathcal{O})$. $X^\dagger_i$ can be written as $(X^\dagger_{i,1}, \ldots, X^\dagger_{i,n})$ based on the stratification of $\mathcal{O}$, i.e., $X^\dagger_{i,j} = X^\dagger_i \cap \{\hbar_{ax} \mid ax \in \mathcal{O}_j\}$ for $j = 1, \ldots, n$.

By using the partitioning results, we develop a novel algorithm for checking if $\mathcal{O} \vdash^{lex}_{cons} A(a)$, which is shown in Algorithm 2. $\mathrm{GR}_0(\mathcal{O})$ is defined as in the previous paragraph (line 1). X' is the set of decision atoms not occurring in $\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O})$ (line 2). $\mathrm{GR}'_0(\mathcal{O})$ is obtained from $\mathrm{GR}_0(\mathcal{O})$ by deleting all decision atoms in X' (line 3). $\mathrm{MM}_0(\mathcal{O})$ is the unique minimal model of the definite fragment of $\mathrm{GR}'_0(\mathcal{O})$ (line 4). $\mathrm{GR}^r_0(\mathcal{O})$ is obtained from $\mathrm{GR}'_0(\mathcal{O})$ by deleting each rule that has at least one head atom in $\mathrm{MM}_0(\mathcal{O})$, and by removing all atoms in $\mathrm{MM}_0(\mathcal{O})$ from the remaining rules (line 5). Then, the satisfiability test in Corollary 1 can be performed over a subprogram of $\mathrm{GR}(\mathcal{O})$, extracted from $\mathrm{GR}_i(\mathcal{O})_{1 \le i \le p}$, $\mathrm{GR}^r_0(\mathcal{O})$ or $\mathrm{MM}_0(\mathcal{O})$, as shown in the following cases (lines 6–18), where $\mathrm{lmw}_i(\mathrm{GR}_k(\mathcal{O}), X^\dagger_k)$ can be computed over $\mathrm{GR}_k(\mathcal{O})$ in the same way as computing $\mathrm{lmw}_i(\mathcal{R}(\mathcal{O}), X)$, cf. Formula (1).

In case $A(a) \in \mathrm{MM}_0(\mathcal{O})$ (line 6), we consider $\mathrm{MM}_0(\mathcal{O})$ only. Let $M_0 = \bigcup_{r \in \mathrm{GR}_0(\mathcal{O})} \mathrm{head}(r) \setminus (\mathrm{atoms}(\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O})) \cup X)$. Then by Property (4) of the partitioning results, $M_0 \cap \mathrm{head}(r) \neq \emptyset$ for all rules $r \in \mathrm{GR}_0(\mathcal{O})$. For every X-

Algorithm 1 ([6]). `Partition`($\mathrm{GR}(\mathcal{O})$, X)

Input: The grounded repair program $\mathrm{GR}(\mathcal{O})$ and the set X of decision atoms.

Output: A set of disjoint subprograms of $\mathrm{GR}(\mathcal{O})$.

Set $map(\alpha)$ as 0 for all ground atoms α occurring in $\mathrm{GR}(\mathcal{O})$; $k := 0$;
Move all rules $r \in \mathrm{GR}(\mathcal{O})$ s.t. $\mathrm{head}(r) \subseteq X$ in front of other rules in $\mathrm{GR}(\mathcal{O})$;
repeat
  $merged :=$ false;
  for each rule r sequentially retrieved from $\mathrm{GR}(\mathcal{O})$ s.t. $\mathrm{head}(r) = \emptyset$ or $\forall \alpha \in \mathrm{head}(r) \setminus X : map(\alpha) > 0$ **do**
    for each $\alpha \in \mathrm{head}(r) \cup \mathrm{body}(r)$ s.t. $map(\alpha) = 0$ **do**
      $k := k + 1$; $map(\alpha) := k$;
    if $|map(r)| > 1$ **then**
      $merged :=$ true; $min_{id} := \min(map(r))$;
      for each $\alpha \in \mathrm{head}(r) \cup \mathrm{body}(r)$ **do** $map(\alpha) := min_{id}$;
until not $merged$;
for $i = 1, \ldots, k$ **do**
  $\Pi_i := \{r \in \mathrm{GR}(\mathcal{O}) \mid \forall \alpha \in \mathrm{head}(r) \cup \mathrm{body}(r) : map(\alpha) = i\}$;
return $\{\Pi_i \neq \emptyset \mid 1 \leq i \leq k\}$;

Algorithm 2. `AtomicCheck`($\{\mathrm{GR}_i(\mathcal{O})\}_{1 \leq i \leq p}$, X, $A(a)$)

Input: The set of subprograms $\{\mathrm{GR}_i(\mathcal{O})\}_{1 \leq i \leq p}$ returned by `Partition`($\mathrm{GR}(\mathcal{O})$, X), the set X of decision atoms and an atomic concept membership axiom $A(a)$ to be checked.

Output: true if $\mathcal{O} \vdash^{lex}_{cons} A(a)$, false otherwise.

$\mathrm{GR}_0(\mathcal{O}) := \mathrm{GR}(\mathcal{O}) \setminus \bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O})$;
$X' := X \setminus \mathrm{atoms}(\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O}))$;
$\mathrm{GR}'_0(\mathcal{O}) = \{\bigvee \mathrm{head}(r) \setminus X' \leftarrow \bigwedge \mathrm{body}(r) \mid r \in \mathrm{GR}_0(\mathcal{O})\}$;
$\mathrm{MM}_0(\mathcal{O}) :=$ the unique minimal model of $\{r \in \mathrm{GR}'_0(\mathcal{O}) \mid |\mathrm{head}(r)| = 1\}$;
$\mathrm{GR}^r_0(\mathcal{O}) := \{\bigvee \mathrm{head}(r) \leftarrow \bigwedge \mathrm{body}(r) \setminus \mathrm{MM}_0(\mathcal{O}) \mid r \in \mathrm{GR}'_0(\mathcal{O}), \mathrm{head}(r) \cap \mathrm{MM}_0(\mathcal{O}) = \emptyset\}$;
if $A(a) \in \mathrm{MM}_0(\mathcal{O})$ **then return** true;
if $A(a) \in \mathrm{atoms}(\mathrm{GR}_k(\mathcal{O}))$ for some $k > 0$ **then**
  $\Pi := \mathrm{GR}_k(\mathcal{O}) \cup \{\sum_{\hbar_{ax} \in X^{\dagger}_{k,i}} \mathrm{assign}(\hbar_{ax}) \leq \mathrm{lmw}_i(\mathrm{GR}_k(\mathcal{O}), X^{\dagger}_k) \mid 1 \leq i \leq n\}$;
  if $\Pi \cup \{\leftarrow A(a)\}$ is unsatisfiable **then return** true **else return** false;
if $A(a) \in \mathrm{head}(r) \setminus \mathrm{atoms}(\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O}))$ for some rule $r \in \mathrm{GR}^r_0(\mathcal{O})$ **then**
  $\Pi_{pre} := \emptyset$; $\Pi := \{r \in \mathrm{GR}^r_0(\mathcal{O}) \mid \mathrm{head}(r) \subseteq X \cup \{A(a)\}\}$;
  while $\Pi \neq \Pi_{pre}$ **do**
    $\Pi_{pre} := \Pi$;
    $\Pi := \Pi_{pre} \cup \{r \in \mathrm{GR}^r_0(\mathcal{O}) \mid \mathrm{head}(r) \subseteq \mathrm{atoms}(\Pi_{pre} \cup \bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O}))\}$;
  $S_N := \{1 \leq i \leq p \mid \mathrm{atoms}(\Pi) \cap \mathrm{atoms}(\mathrm{GR}_i(\mathcal{O})) \neq \emptyset\}$;
  $\Pi' := \Pi \cup \bigcup_{k \in S_N} (\mathrm{GR}_k(\mathcal{O}) \cup \{\sum_{\hbar_{ax} \in X^{\dagger}_{k,i}} \mathrm{assign}(\hbar_{ax}) \leq \mathrm{lmw}_i(\mathrm{GR}_k(\mathcal{O}), X^{\dagger}_k) \mid 1 \leq i \leq n\})$;
  if $\Pi' \cup \{\leftarrow A(a)\}$ is unsatisfiable **then return** true **else return** false;
return false;

lex-minimal model M of $\mathrm{GR}(\mathcal{O})$, M does not contain any decision atom in X', otherwise $M' = (M \cap \mathrm{atoms}(\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O}))) \cup M_0$ will be a model of $\mathrm{GR}(\mathcal{O})$ such that $(M' \cap X_1, \ldots, M' \cap X_n) <^{lex} (M \cap X_1, \ldots, M \cap X_n)$, contracting that M is an X-lex-minimal model of $\mathrm{GR}(\mathcal{O})$. Hence, $M \cap \mathrm{atoms}(\mathrm{GR}'_0(\mathcal{O}))$ is a model of $\mathrm{GR}'_0(\mathcal{O})$. It follows that $\mathrm{MM}_0(\mathcal{O}) \subseteq M$ and thus $A(a) \in M$. By Corollary 1, we have $\mathcal{O} \vdash^{lex}_{cons} A(a)$.

In case $A(a) \in \mathrm{atoms}(\mathrm{GR}_k(\mathcal{O}))$ for some $k > 0$ (lines 7–9), we consider $\mathrm{GR}_k(\mathcal{O})$ only. In addition, we encode the criterion for $X^{\dagger}_k$-lex-minimal models as PB-constraints, obtaining Π (line 8). (a) If $\Pi \cup \{\leftarrow A(a)\}$ is satisfiable, there must be an $X^{\dagger}_k$-lex-minimal model M_k of $\mathrm{GR}_k(\mathcal{O})$ such that $M_k \subseteq \mathrm{atoms}(\mathrm{GR}_k(\mathcal{O}))$ and $A(a) \notin M_k$. Let M_j denote an arbitrary $X^{\dagger}_j$-lex-minimal model of $\mathrm{GR}_j(\mathcal{O})$ such that $M_j \subseteq \mathrm{atoms}(\mathrm{GR}_j(\mathcal{O}))$ for each $j > 0$ and $j \neq k$. Let $M_0 = \bigcup_{r \in \mathrm{GR}^r_0(\mathcal{O})} \mathrm{head}(r) \setminus \mathrm{atoms}(\bigcup_{j=1}^{p} \mathrm{GR}_j(\mathcal{O}))$. Then $M_0 \cap X = \emptyset$ and $M_0 \cap \mathrm{head}(r) \neq \emptyset$ for each rule $r \in \mathrm{GR}^r_0(\mathcal{O})$. Furthermore, for any $0 \leq i < j \leq p$, $M_i \cap M_j = \emptyset$; for any $0 \leq j \leq p$, $\mathrm{MM}_0(\mathcal{O}) \cap M_j = \emptyset$ and $A(a) \notin M_j$; $A(a) \notin M_0$ and $A(a) \notin \mathrm{MM}_0(\mathcal{O})$. So $M = \mathrm{MM}_0(\mathcal{O}) \cup M_0 \cup \bigcup_{j=1}^{p} M_j$ is a model of $\mathrm{GR}(\mathcal{O})$ such that $A(a) \notin M$. M must be an X-lex-minimal model of $\mathrm{GR}(\mathcal{O})$, otherwise by Theorem 3, there must exist some $j > 0$ such that M_j is not an $X^{\dagger}_j$-lex-minimal model of $\mathrm{GR}_j(\mathcal{O})$. So by Corollary 1, $\mathcal{O} \nvdash^{lex}_{cons} A(a)$. (b) If $\Pi \cup \{\leftarrow A(a)\}$ is unsatisfiable, $A(a)$ will be in every $X^{\dagger}_k$-lex-minimal model of $\mathrm{GR}_k(\mathcal{O})$. For every X-lex-minimal model M of $\mathrm{GR}(\mathcal{O})$, since $M \cap \mathrm{atoms}(\mathrm{GR}_k(\mathcal{O}))$ is an $X^{\dagger}_k$-lex-minimal model of $\mathrm{GR}_k(\mathcal{O})$, we have $A(a) \in M$, so $\mathcal{O} \vdash^{lex}_{cons} A(a)$.

In case $A(a) \in \mathrm{head}(r) \setminus \mathrm{atoms}(\bigcup_{i=1}^{p} \mathrm{GR}_i(\mathcal{O}))$ for some rule $r \in \mathrm{GR}^r_0(\mathcal{O})$ (lines 10–17), we consider the union of a subprogram Π iteratively extracted from $\mathrm{GR}^r_0(\mathcal{O})$ (lines 11–14) and every $\mathrm{GR}_k(\mathcal{O})$ that has some atoms occurring in Π (lines 15,16). In addition, we encode the criterion for $X^{\dagger}_k$-lex-minimal models as PB-constraints, obtaining Π' (line 16). (a) If $\Pi' \cup \{\leftarrow A(a)\}$ is satisfiable, there must be a model M of Π' such that $M \subseteq \mathrm{atoms}(\Pi')$ and $A(a) \notin M$. Obviously, $M \cap \mathrm{atoms}(\mathrm{GR}_k(\mathcal{O}))$ is an $X^{\dagger}_k$-lex-minimal model of $\mathrm{GR}_k(\mathcal{O})$ for every $k \in S_N$. Let $M_0 = \bigcup_{r \in \mathrm{GR}^r_0(\mathcal{O}) \setminus \Pi} \mathrm{head}(r) \setminus \mathrm{atoms}(\Pi \cup \bigcup_{j=1}^{p} \mathrm{GR}_j(\mathcal{O}))$. Then $M_0 \cap X = \emptyset$ and $M_0 \cap \mathrm{head}(r) \neq \emptyset$ for each rule $r \in \mathrm{GR}^r_0(\mathcal{O}) \setminus \Pi$. Moreover, since $A(a) \in \mathrm{atoms}(\Pi)$, we have $A(a) \notin M_0$. So $M \cup M_0$ is a model of $\Pi' \cup \mathrm{GR}^r_0(\mathcal{O})$ such that $A(a) \notin M \cup M_0$. Let M_j denote an arbitrary $X^{\dagger}_j$-lex-minimal model of $\mathrm{GR}_j(\mathcal{O})$ such that $M_j \subseteq \mathrm{atoms}(\mathrm{GR}_j(\mathcal{O}))$ for each $j \in \{1, \ldots, p\} \setminus S_N$. Then for any $j \in \{1, \ldots, p\} \setminus S_N$, $M_j \cap (\mathrm{MM}_0(\mathcal{O}) \cup M \cup M_0) = \emptyset$ and $A(a) \notin M_j$. Furthermore, $(M \cup M_0) \cap \mathrm{MM}_0(\mathcal{O}) = \emptyset$, $A(a) \notin M \cup M_0$ and $A(a) \notin \mathrm{MM}_0(\mathcal{O})$. So $M' = M \cup M_0 \cup \mathrm{MM}_0(\mathcal{O}) \cup \bigcup_{j \in \{1,\ldots,p\} \setminus S_N} M_j$ is a model of $\mathrm{GR}(\mathcal{O})$ such that $A(a) \notin M'$. M' must be an X-lex-minimal model of $\mathrm{GR}(\mathcal{O})$, otherwise by Theorem 3, there must exist some $k \in S_N$ such that $M \cap \mathrm{atoms}(\mathrm{GR}_k(\mathcal{O}))$ is not an $X^{\dagger}_k$-lex-minimal model of $\mathrm{GR}_k(\mathcal{O})$, or some $j \in \{1, \ldots, p\} \setminus S_N$ such that M_j is not an $X^{\dagger}_j$-lex-minimal model of $\mathrm{GR}_j(\mathcal{O})$. So by Corollary 1, $\mathcal{O} \nvdash^{lex}_{cons} A(a)$. (b) If $\Pi' \cup \{\leftarrow A(a)\}$ is unsatisfiable, $A(a)$ will be in every model of Π'. For every X-lex-minimal model M of $\mathrm{GR}(\mathcal{O})$, since $M \cap \mathrm{atoms}(\Pi')$ is a model of Π', we have $A(a) \in M$, so $\mathcal{O} \vdash^{lex}_{cons} A(a)$.

In other cases (line 18), we can directly get the answer. Let M_i denote an arbitrary $X_i^\dagger$-lex-minimal model of $\mathrm{GR}_i(\mathcal{O})$ such that $M_i \subseteq \mathrm{atoms}(\mathrm{GR}_i(\mathcal{O}))$ for each $i > 0$. Let $M_0 = \bigcup_{r \in \mathrm{GR}_0^r(\mathcal{O})} \mathrm{head}(r) \setminus \mathrm{atoms}(\bigcup_{i=1}^p \mathrm{GR}_i(\mathcal{O}))$. It can be shown analogously as in the second case that, $M = \mathrm{MM}_0(\mathcal{O}) \cup M_0 \cup \bigcup_{i=1}^p M_i$ is an X-lex-minimal model of $\mathrm{GR}(\mathcal{O})$ such that $A(a) \notin M$. So by Corollary 1, $\mathcal{O} \not\vdash_{cons}^{lex} A(a)$.

According to the above analyses, we have the following conclusion.

Theorem 7. *The above algorithm for checking if $\mathcal{O} \vdash_{cons}^{lex} A(a)$ is correct.* □

Example 6 (Example 5 continued). By applying Algorithm 1, $\mathrm{GR}(\mathcal{O})$ is decomposed into $\mathrm{GR}_1(\mathcal{O}) = \{r_1, r_8, r_9, r_{10}, r_{11}, r_{12}\}$ and $\mathrm{GR}_2(\mathcal{O}) = \{r_3, r_4, r_5, r_6, r_7, r_{13}, r_{14}\}$. Afterwards, the decomposition is continued to yield in turn $\mathrm{GR}_0(\mathcal{O}) = \{r_2, r_{15}, r_{16}\}$, $\mathrm{MM}_0(\mathcal{O}) = \{B(b)\}$ and $\mathrm{GR}_0^r(\mathcal{O}) = \{B(c) \leftarrow b \approx c\}$. The decomposition also divides X into $X_1^\dagger = (\{\hbar_{ax_1}\}, \emptyset, \{\hbar_{ax_7}, \hbar_{ax_8}\}, \{\hbar_{ax_9}\})$ and $X_2^\dagger = (\{\hbar_{ax_3}\}, \{\hbar_{ax_4}, \hbar_{ax_5}, \hbar_{ax_6}\}, \emptyset, \emptyset)$. It can be computed by Formula (1), where $\mathcal{R}(\mathcal{O})$ and X are replaced with $\mathrm{GR}_i(\mathcal{O})$ and $X_i^\dagger$ respectively, that $\mathrm{lmw}(\mathrm{GR}_1(\mathcal{O}), X_1^\dagger) = (0, 0, 0, 1)$ and $\mathrm{lmw}(\mathrm{GR}_2(\mathcal{O}), X_2^\dagger) = (0, 1, 0, 0)$.

Consider deciding if $\mathcal{O} \vdash_{cons}^{lex} A(a)$. Since $A(a) \in \mathrm{atoms}(\mathrm{GR}_1(\mathcal{O}))$, the satisfiability of Π is tested, where $\Pi = \mathrm{GR}_1(\mathcal{O}) \cup \{\mathrm{assign}(\hbar_{ax_1}) \leq 0, \mathrm{assign}(\hbar_{ax_7}) + \mathrm{assign}(\hbar_{ax_8}) \leq 0, \mathrm{assign}(\hbar_{ax_9}) \leq 1\}$. Since $\Pi \cup \{\leftarrow A(a)\}$ is unsatisfiable, we have $\mathcal{O} \vdash_{cons}^{lex} A(a)$.

Consider deciding if $\mathcal{O} \vdash_{cons}^{lex} B(c)$. Since $B(c) \in \mathrm{head}(r) \setminus \mathrm{atoms}(\mathrm{GR}_1(\mathcal{O}) \cup \mathrm{GR}_2(\mathcal{O}))$ for the unique rule $r \in \mathrm{GR}_0^r(\mathcal{O})$, the satisfiability of $\Pi' \cup \{\leftarrow B(c)\}$ is tested, where $\Pi' = \mathrm{GR}_0^r(\mathcal{O}) \cup \mathrm{GR}_1(\mathcal{O}) \cup \{\mathrm{assign}(\hbar_{ax_3}) \leq 0, \mathrm{assign}(\hbar_{ax_4}) + \mathrm{assign}(\hbar_{ax_5}) + \mathrm{assign}(\hbar_{ax_6}) \leq 1\}$. Since $\Pi' \cup \{\leftarrow B(c)\}$ is satisfiable (e.g., the set $\{Q_1(a), T(a, b), \hbar_{ax_5}\}$ is a model), we have $\mathcal{O} \not\vdash_{cons}^{lex} B(c)$.

Consider deciding if $\mathcal{O} \vdash_{cons}^{lex} B(b)$. Since $B(b) \in \mathrm{MM}_0(\mathcal{O})$, we have $\mathcal{O} \vdash_{cons}^{lex} B(b)$. Consider deciding if $\mathcal{O} \vdash_{cons}^{lex} B(a)$. Since $B(a) \notin \mathrm{MM}_0(\mathcal{O}) \cup \mathrm{atoms}(\mathrm{GR}_0^r(\mathcal{O}) \cup \mathrm{GR}_1(\mathcal{O}) \cup \mathrm{GR}_2(\mathcal{O}))$, we have $\mathcal{O} \not\vdash_{cons}^{lex} B(a)$. □

5.4 Checking Other Consequences

First of all, we consider deciding if $\mathcal{O} \vdash_{cohe}^{lex} A \sqsubseteq B$ for $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ and A and B atomic concepts. We assume that the extension of $\mathcal{O}$ given in Theorem 1, i.e. $\mathcal{O}' = (\mathcal{A}, \mathcal{O}_1, \ldots, \mathcal{O}_n)$, has been *fully compiled*: $\mathrm{GR}(\mathcal{O}')$ has been constructed and decomposed, yielding $\mathrm{GR}_i(\mathcal{O}')_{1 \leq i \leq p}$, $\mathrm{MM}_0(\mathcal{O}')$ and $\mathrm{GR}_0^r(\mathcal{O}')$; moreover, for each i, $\mathrm{lmw}(\mathrm{GR}_i(\mathcal{O}'), X_i^\dagger)$ has been computed. Let $A(a)$ be the axiom over A in $\mathcal{A}$. Then $A(a)$ must be in every lex-maximal consistent subontology of $\mathcal{O}'$. So $\mathcal{O}' \vdash_{cons}^{lex} A \sqsubseteq B$ iff $\mathcal{O}' \vdash_{cons}^{lex} B(a)$. By Theorem 1, we have $\mathcal{O} \vdash_{cohe}^{lex} A \sqsubseteq B$ iff $\mathcal{O}' \vdash_{cons}^{lex} B(a)$, where the latter can be checked by one call to a SAT solver using the method given in the previous subsection.

To decide if $\mathcal{O} \vdash_{cons}^{lex} C(a)$ for $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ and C a general concept, we consider a new ontology $\mathcal{O}' = (\{C \sqsubseteq Q\}, \mathcal{O}_1, \ldots, \mathcal{O}_n)$, where Q is a new concept name. Since the axiom $C \sqsubseteq Q$ must be in every lex-maximal consistent

subontology of $\mathcal{O}'$, we have $\mathcal{O} \vdash^{lex}_{cons} C(a)$ iff $\mathcal{O}' \vdash^{lex}_{cons} Q(a)$, where the latter can be checked using the proposed method, as $Q(a)$ is an atomic concept membership axiom.

To decide if $\mathcal{O} \vdash^{lex}_{cons} C \sqsubseteq D$ for C and D general concepts, we consider the axiom $(\neg C \sqcup D)(a)$ for a a new globally unique individual. Since $C \sqsubseteq D$ is equivalent to $\top \sqsubseteq \neg C \sqcup D$ and $\mathcal{O} \vdash^{lex}_{cons} (\top \sqsubseteq \neg C \sqcup D)$ iff $\mathcal{O} \vdash^{lex}_{cons} (\neg C \sqcup D)(a)$, we have $\mathcal{O} \vdash^{lex}_{cons} C \sqsubseteq D$ iff $\mathcal{O} \vdash^{lex}_{cons} (\neg C \sqcup D)(a)$, where the latter can be checked using the method just given above.

To decide if $\mathcal{O} \vdash^{lex}_{cohe} C(a)$ or decide if $\mathcal{O} \vdash^{lex}_{cohe} C \sqsubseteq D$ for C and D general concepts, we first reduce the problem to checking the corresponding lex-consistent consequence by Theorem 1, then solve it in the same way. Note that the above problems are hard to be solved as efficiently as checking if $\mathcal{O} \vdash^{lex}_{cons} A(a)$ for A an atomic concept, in the sense that they necessitate computations from scratch even when $\mathcal{O}$ is fully compiled.

6 Conclusion and Future Work

In this paper, we applied the lexicographic inference to reason over inconsistent DL-based ontologies and addressed the problem of checking lex-consistent (or lex-coherent) consequences of a $\mathcal{SHIQ}$ ontology. Basically, our proposed method compiles the input $\mathcal{SHIQ}$ ontology to a propositional program, so that the addressing problem is solved in polynomial calls to current powerful SAT solvers. The method is the first worst-case optimal one (in data complexity) for checking lex-consistent consequences of a $\mathcal{SHIQ}$ ontology. It performs the checking without computing any lex-maximal consistent subontology. It can also be applied to check lex-coherence consequences by first reducing the problem to that of checking lex-consistent consequences. In order to make the method more scalable, we also gave partition-based techniques to optimize the calling of SAT solvers.

For future work, we plan to conduct a thorough evaluation for the proposed method and extend the method to handle OWL DL ontologies. The extension for datatypes is trivial because datatypes have been handled in the KAON2 transformation. The extension for nominals is feasible in theory due to the existence of a resolution-based decision procedure for $\mathcal{SHOIQ}$ [16], which can be extended analogously by embedding decision atoms. Hence, the problem of checking a lex-consistent consequence of an OWL DL ontology can also be treated as a set of satisfiability problems, solved by the extended decision procedure after all PB-constraints on decision atoms are translated to SAT clauses. However, the time complexity of the decision procedure for $\mathcal{SHOIQ}$ is up to triply exponential [16], so we still need to develop practical methods to handle nominals.

Acknowledgments. This work is supported in part by NSFC grants 60673103 and 60721061.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.
3. S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 640–647, 1993.
4. S. Benferhat, D. Dubois, and H. Prade. How to infer from inconsisent beliefs without revising? In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1449–1457, 1995.
5. C. Cayrol and M. Lagasquie-Schiex. On the complexity of non-monotonic entailment in syntax-based approaches. In *Proc. of ECAI'94 Workshop on Algorithms, Complexity and Commonsense Reasoning*, 1994.
6. J. Du and Y. Shen. Computing minimum cost diagnoses to repair populated DL-based ontologies. In *Proc. of the 17th International World Wide Web Conference (WWW)*, 2008.
7. N. Eén and N. Sörensson. Translating pseudo-boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:1–26, 2006.
8. T. Eiter, G. Gottlob, and H. Mannila. Disjunctive datalog. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.
9. T. Eiter, N. Leone, C. Mateis, G. Pfeifer, and F. Scarcello. A deductive system for non-monotonic reasoning. In *Proc. of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 364–375, 1997.
10. M. Fitting. *First-order Logic and Automated Theorem Proving (2nd ed.)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
11. P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *Proc. of the 4th International Semantic Web Conference (ISWC)*, pages 353–367, 2005.
12. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3), 2000.
13. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 454–459, 2005.
14. U. Hustadt, B. Motik, and U. Sattler. Reasoning in description logics by a reduction to disjunctive datalog. *Journal of Automated Reasoning*, 39(3):351–384, 2007.
15. A. Kalyanpur, B. Parsia, E. Sirin, and B. C. Grau. Repairing unsatisfiable concepts in owl ontologies. In *Proc. of the 3rd European Semantic Web Conference (ESWC)*, pages 170–184, 2006.
16. Y. Kazakov and B. Motik. A resolution-based decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 3rd International Joint Conference on Automated Reasoning (IJCAR)*, pages 662–677, 2006.

17. M. W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

18. D. Lembo and M. Ruzzi. Consistent query answering over description logic ontologies. In *Proc. of the 1st International Conference on Web Reasoning and Rule Systems (RR)*, pages 194–208, 2007.

19. Y. Ma, P. Hitzler, and Z. Lin. Algorithms for paraconsistent reasoning with owl. In *Proc. of the 4th European Semantic Web Conference (ESWC)*, pages 399–413, 2007.

20. T. Meyer, K. Lee, and R. Booth. Knowledge integration for description logics. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 645–650, 2005.

21. B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Univesität karlsruhe, Germany, January 2006.

22. R. Nieuwenhuis and A. Rubio. Theorem proving with ordering and equality constrained clauses. *Journal of Symbolic Computation*, 19(4):321–351, 1995.

23. S. P. Odintsov and H.Wansing. *Inconsistency-tolerant Description Logic: Motivation and Basic Systems*, pages 301–335. Kluwer Academic Publishers, 2003.

24. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proc. of the 14th International Conference on World Wide Web (WWW)*, pages 633–640, 2005.

25. P. F. Patel-Schneider, P. Hayes, and I. Horrocks. *OWL Web Ontology Language Semantics and Abstract Syntax*. W3C Recommendation, 2004. `http://www.w3.org/TR/owl-semantics/`.

26. D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2(3):293–304, 1986.

27. G. Qi, W. Liu, and D. Bell. A revision-based approach to handling inconsistency in description logics. *Artificial Intelligence Review*, 26(1-2):115–128, 2006.

28. G. Qi, J. Z. Pan, and Q. Ji. Extending description logics with uncertainty reasoning in possibilistic logic. In *Proc. of European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, pages 828–839, 2007.

29. S. Schlobach. Diagnosing terminologies. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 670–675, 2005.

30. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 355–362, 2003.

31. H. M. Sheini and K. A. Sakallah. Pueblo: A hybrid pseudo-boolean SAT solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:157–181, 2006.

32. L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: Preliminary report. In *Proc. of the 5th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–9, 1973.