TYPE-ELIMINATION-BASED REASONING FOR THE DESCRIPTION LOGIC $\mathcal{SHIQb}_s$ USING DECISION DIAGRAMS AND DISJUNCTIVE DATALOG

SEBASTIAN RUDOLPH, MARKUS KRÖTZSCH, AND PASCAL HITZLER

Institute AIFB, Karlsruhe Institute of Technology, Germany
e-mail address: rudolph@kit.edu

Department of Computer Science, University of Oxford, UK
e-mail address: markus.kroetzsch@cs.ox.ac.uk

Kno.e.sis, Wright State University, Dayton, Ohio, US
e-mail address: pascal.hitzler@wright.edu

ABSTRACT. We propose a novel, type-elimination-based method for standard reasoning in the description logic $\mathcal{SHIQb}_s$ extended by DL-safe rules. To this end, we first establish a knowledge compilation method converting the terminological part of an $\mathcal{ALCIb}$ knowledge base into an ordered binary decision diagram (OBDD) that represents a canonical model. This OBDD can in turn be transformed into disjunctive Datalog and merged with the assertional part of the knowledge base in order to perform combined reasoning. In order to leverage our technique for full $\mathcal{SHIQb}_s$, we provide a stepwise reduction from $\mathcal{SHIQb}_s$ to $\mathcal{ALCIb}$ that preserves satisfiability and entailment of positive and negative ground facts. The proposed technique is shown to be worst-case optimal w.r.t. combined and data complexity.

1. INTRODUCTION

Description logics (DLs, see Baader et al., 2007) have become a major paradigm in Knowledge Representation and Reasoning. This can in part be attributed to the fact that the DLs have been found suitable to be the foundation for ontology modeling and reasoning for the Semantic Web. In particular, the Web Ontology Language OWL (W3C OWL Working Group, 2009), a recommended standard by the World Wide Web Consortium (W3C)[1] for ontology modeling, is essentially a description logic (see, e.g., Hitzler et al., 2009, for an introduction to OWL and an in-depth description of the correspondences). As such, DLs are currently gaining significant momentum in application areas, and are being picked up as knowledge representation paradigm by both industry and applied research.

[1] http://www.w3.org/

The DL known as $\mathcal{SHIQ}$ is among the most prominent DL fragments that do not feature nominals,[2] and it covers most of the OWL language. Various OWL reasoners implement efficient reasoning support for $\mathcal{SHIQ}$ by means of tableau methods, e.g., Pellet,[3] FaCT++,[4] or RacerPro,[5].

However, even the most efficient implementations of reasoning algorithms to date do not scale up to very data-intensive application scenarios. This motivates the search for alternative reasoning approaches that build upon different methods in order to address cases where tableau algorithms turn out to have certain weaknesses. Successful examples are KAON2 (Motik and Sattler, 2006) based on resolution, HermiT (Motik et al., 2009) based on hyper-tableaux, as well as the consequence-based systems CB (Kazakov, 2009), ConDOR (Simančík et al., 2011), and ELK (Kazakov et al., 2011). Moreover, especially for lightweight DLs, approaches based on rewriting queries (Calvanese et al., 2007a) or both queries and data (Kontchakov et al., 2010) have been proposed.

In this paper, we propose the use of a variant of *type elimination*, a notion first introduced by Pratt (1979), as a reasoning paradigm for DLs. To implement the necessary computations on large type sets in a compressed way, we suggest the use of ordered binary decision diagrams (OBDDs). OBDDs have been applied successfully in the domain of large-scale model checking and verification, but have hitherto seen only little investigation in DLs, e.g., by Pan et al. (2006).

Most of the description logics considered in this article exhibit restricted Boolean role expressions as a non-standard modeling feature, which is indicated by a *b* or (if further restricted) b_s in the name of the DL. In particular, we propose a novel method for reasoning in $\mathcal{SHIQ}b_s$ knowledge bases featuring terminological and assertional knowledge including (in)equality statements as well as DL-safe rules.

Our work starts by considering terminological reasoning in the DL $\mathcal{ALCI}b$, which is less expressive than $\mathcal{SHIQ}b_s$. We introduce a method that compiles an $\mathcal{ALCI}b$ terminology into an OBDD representation. Thereafter, we show that the output of this algorithm can be used for generating a disjunctive Datalog program that can in turn be combined with ABox data to obtain a correct reasoning procedure. Finally, the results for $\mathcal{ALCI}b$ are lifted to full $\mathcal{SHIQ}b_s$ by providing an appropriate translation from the latter to the former.

This article combines and consolidates our previous work about pure TBox reasoning (Rudolph et al., 2008c), its extension to ABoxes (Rudolph et al., 2008b) and some notes on reasoning in DLs with Boolean role expressions (Rudolph et al., 2008a) by

- providing a collection of techniques for eliminating $\mathcal{SHIQ}b_s$ modeling features that impede the use of our type elimination approach,
- laying out the model-theoretic foundations for type-elimination-based reasoning for very expressive description logics without nominals, using the *domino* metaphor for 2-types,
- elaborating the possibility of using OBDDs for making type elimination computationally feasible,
- providing a canonical translation of OBDDs into disjunctive Datalog to enable reasoning with assertional information, and
- making the full proofs accessible in a published version.

Moreover, we extend our work by adding some missing aspects and completing the theoretical investigations by

---

[2]Nominals, i.e., concepts that denote a set with exactly one element, usually cause a reasoning efficiency problem when added to $\mathcal{SHIQ}$. This is evident from the performance of existing systems, and finds its theoretical justification in the fact that they increase worst-case complexity from ExpTime-completeness to NExpTime-completeness.

[3]http://clarkparsia.com/pellet/

[4]http://owl.man.ac.uk/factplusplus/

[5]http://www.racer-systems.com/

- extending the procedures for reducing $\mathcal{SHIQb}_s$ to $\mathcal{ALCIb}$ to ABoxes and DL-safe rules,
- establishing worst-case optimality of our algorithms,
- extending the supported language: while our previous work only covered terminological reasoning in $\mathcal{SHIQ}$ (Rudolph et al., 2008c) and combined reasoning in $\mathcal{ALCIb}$ (Rudolph et al., 2008b), we now support reasoning in $\mathcal{SHIQb}_s$ knowledge bases featuring terminological and assertional knowledge, including (in)equality statements and DL-safe rules.

The structure of this article is as follows. Section 2 recalls relevant preliminaries. Section 3 discusses the computation of sets of *dominoes* that represent models of $\mathcal{ALCIb}$ knowledge bases. Section 4 casts this computation into a manipulation of OBDDs as underlying data structures. Section 5 discusses how the resulting OBDD presentation can be transformed to disjunctive Datalog and establishes the correctness of the approach. Section 6 provides a transformation from $\mathcal{SHIQb}_s$ to $\mathcal{ALCIb}$, thereby extending the applicability of the proposed method to $\mathcal{SHIQb}_s$ knowledge bases. Section 7 discusses related work and Section 8 concludes.

2. The Description Logics $\mathcal{SHIQb}_s$ and $\mathcal{ALCIb}$

We first recall some basic definitions of DLs and introduce our notation. A more gentle first introduction to DLs, together with pointers to further reading, is given in Rudolph (2011). Here, we define a rather expressive description logic $\mathcal{SHIQb}_s$ that extends $\mathcal{SHIQ}$ with restricted Boolean role expressions (see, e.g., Tobies, 2001).

Definition 1. *A $\mathcal{SHIQb}_s$ knowledge base is based on three disjoint sets of* concept names N_C, role names N_R, *and* individual names N_I. *The set of* atomic roles $\mathbf{R}$ *is defined by* $\mathbf{R} := N_R \cup \{R^- \mid R \in N_R\}$. *In addition, we let* $\mathrm{Inv}(R) := R^-$ *and* $\mathrm{Inv}(R^-) := R$, *and we extend this notation also to sets of atomic roles. In the following, we use the symbols R and S to denote atomic roles, if not specified otherwise. The set of* Boolean role expressions $\mathbf{B}$ *is defined as*

$$\mathbf{B} ::= \mathbf{R} \mid \neg \mathbf{B} \mid \mathbf{B} \sqcap \mathbf{B} \mid \mathbf{B} \sqcup \mathbf{B}.$$

We use $\vdash$ to denote entailment between sets of atomic roles and role expressions. Formally, given a set $\mathcal{R}$ of atomic roles, we inductively define:

- *for atomic roles R, $\mathcal{R} \vdash R$ if $R \in \mathcal{R}$, and $\mathcal{R} \nvdash R$ otherwise,*
- *$\mathcal{R} \vdash \neg U$ if $\mathcal{R} \nvdash U$, and $\mathcal{R} \nvdash \neg U$ otherwise,*
- *$\mathcal{R} \vdash U \sqcap V$ if $\mathcal{R} \vdash U$ and $\mathcal{R} \vdash V$, and $\mathcal{R} \nvdash U \sqcap V$ otherwise,*
- *$\mathcal{R} \vdash U \sqcup V$ if $\mathcal{R} \vdash U$ or $\mathcal{R} \vdash V$, and $\mathcal{R} \nvdash U \sqcup V$ otherwise.*

A Boolean role expression U is restricted *if $\emptyset \nvdash U$. The set of all restricted role expressions is denoted by $\mathbf{T}$, and the symbols U and V will be used throughout this paper to denote restricted role expressions. A $\mathcal{SHIQb}_s$ RBox is a set of axioms of the form $U \sqsubseteq V$ (role inclusion axiom) or $\mathrm{Tra}(R)$ (transitivity axiom). The set of* non-simple *roles (for a given RBox) is defined as the smallest subset of $\mathbf{R}$ satisfying:*

- *If there is an axiom $\mathrm{Tra}(R)$, then R is non-simple.*
- *If there is an axiom $R \sqsubseteq S$ with R non-simple, then S is non-simple.*
- *If R is non-simple, then $\mathrm{Inv}(R)$ is non-simple.*

An atomic role is simple *if it is not non-simple. In $\mathcal{SHIQb}_s$, every non-atomic Boolean role expression must contain only simple roles.*

Based on a $\mathcal{SHIQb}_s$ RBox, the set of concept expressions $\mathbf{C}$ *is defined as*

$$\mathbf{C} ::= N_C \mid \top \mid \bot \mid \neg \mathbf{C} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \forall \mathbf{T}.\mathbf{C} \mid \exists \mathbf{T}.\mathbf{C} \mid {\leqslant} n\, \mathbf{R}.\mathbf{C} \mid {\geqslant} (n+1)\, \mathbf{R}.\mathbf{C},$$

where $n \geq 0$ denotes a natural number, and the role S in expressions $\leq n\,S.C$ and $\geq(n + 1)\,S.C$ is required to be simple. Common names for the various forms of concept expressions are given in Table 1 (lower part). Throughout this paper, the symbols C, D will be used to denote concept expressions. A $\mathcal{SHIQ}b_s$ TBox (or terminology*) is a set of* general concept inclusion axioms *(GCIs) of the form $C \sqsubseteq D$.*

Besides the terminological components, DL knowledge bases typically include assertional knowledge as well. In order to increase expressivity and to allow for a uniform presentation of our approach we generalize this by allowing knowledge bases to contain so-called DL-safe rules as introduced by Motik et al. (2005).

Definition 2. *Let $\mathbf{V}$ be a countable set of first-order variables. A* term *is an element of $\mathbf{V} \cup \mathsf{N}_I$. Given terms t and u, a* concept atom/role atom/equality atom *is a formula of the form $C(t)/R(t, u)/t \approx u$ with $C \in \mathsf{N}_C$ and $R \in \mathsf{N}_R$. A DL-safe rule for $\mathcal{SHIQ}b_s$ is a formula $B \to H$, where B and H are possibly empty conjunctions of (role, concept, and equality) atoms. To simplify notation, we will often use finite sets S of atoms for representing the conjunction $\bigwedge S$.*
A set $\mathcal{P}$ of DL-safe rules is called a rule base. *An* extended $\mathcal{SHIQ}b_s$ knowledge base KB *is a triple $\langle \mathcal{T}, \mathcal{R}, \mathcal{P} \rangle$, where $\mathcal{T}$ is a $\mathcal{SHIQ}b_s$ TBox, $\mathcal{R}$ is a $\mathcal{SHIQ}b_s$ RBox, and $\mathcal{P}$ is a rule base.*

We only consider extended knowledge bases in this work, so we will often just speak of knowledge bases. In the literature, a DL ABox is usually allowed to contain assertions of the form $A(a)$, $R(a, b)$, or $a \approx b$, where $a, b \in \mathsf{N}_I$, $A \in \mathsf{N}_C$, and $R \in \mathsf{N}_R$. We assume that all roles and concepts occurring in the ABox are atomic.[6] These assertions can directly be expressed as DL-safe rules that have empty (vacuously true) bodies and a single head atom. Conversely, the negation of these assertions can be expressed by rules that have the assertion as body atom while having an empty (vacuously false) head. Knowing this, we will not specifically consider assertions or negated assertions in the proofs of this paper. For convenience we will, however, sometimes use the above notations instead of their rule counterparts when referring to (positive or negated) ground facts.
As mentioned above, we will mostly consider fragments of $\mathcal{SHIQ}b_s$. In particular, an (extended) $\mathcal{ALCI}b$ knowledge base is an (extended) $\mathcal{SHIQ}b_s$ knowledge base that contains no RBox axioms and no number restrictions (i.e., concept expressions $\leq n\,R.C$ or $\geq n\,R.C$). Consequently, an extended $\mathcal{ALCI}b$ knowledge base only consists of a pair $\langle \mathcal{T}, \mathcal{P} \rangle$, where $\mathcal{T}$ is a TBox and $\mathcal{P}$ is a rule base. The related DL $\mathcal{ALCQI}b$ has been studied by Tobies (2001).
The semantics of $\mathcal{SHIQ}b_s$ and its sublogics is defined in the usual, model-theoretic way. An interpretation $\mathcal{I}$ consists of a set $\Delta^{\mathcal{I}}$ called *domain* (the elements of it being called *individuals*) together with a function $\cdot^{\mathcal{I}}$ mapping individual names to elements of $\Delta^{\mathcal{I}}$, concept names to subsets of $\Delta^{\mathcal{I}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.
The function $\cdot^{\mathcal{I}}$ is extended to role and concept expressions as shown in Table 1. An interpretation $\mathcal{I}$ *satisfies* an axiom φ if we find that $\mathcal{I} \models \varphi$, where

- $\mathcal{I} \models U \sqsubseteq V$ if $U^{\mathcal{I}} \subseteq V^{\mathcal{I}}$,
- $\mathcal{I} \models \mathsf{Tra}(R)$ if $R^{\mathcal{I}}$ is a transitive relation,
- $\mathcal{I} \models C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$,

$\mathcal{I}$ *satisfies* a knowledge base KB, denoted $\mathcal{I} \models$ KB, if it satisfies all axioms of KB.

---

[6]This common assumption is made without loss of generality in terms of knowledge base expressivity. It is essential for defining the ABox-specific complexity measure of *data complexity*, although it might be questionable in cases where ABox statements with complex concept expressions belong to the part of the knowledge base that is frequently changing.

Name	Syntax	Semantics
inverse role	R^-	$\{\langle x, y\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle y, x\rangle \in R^{\mathcal{I}}\}$
role negation	$\neg U$	$\{\langle x, y\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle x, y\rangle \notin U^{\mathcal{I}}\}$
role conjunction	$U \sqcap V$	$U^{\mathcal{I}} \cap V^{\mathcal{I}}$
role disjunction	$U \sqcup V$	$U^{\mathcal{I}} \cup V^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
universal restriction	$\forall U.C$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in U^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$
existential restriction	$\exists U.C$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in U^{\mathcal{I}}, y \in C^{\mathcal{I}} \text{ for some } y \in \Delta^{\mathcal{I}}\}$
qualified	$\leqslant n\, S.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in S^{\mathcal{I}}, y \in C^{\mathcal{I}}\} \leq n\}$
number restriction	$\geqslant n\, S.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x, y\rangle \in S^{\mathcal{I}}, y \in C^{\mathcal{I}}\} \geq n\}$

Table 1: Semantics of constructors in $\mathcal{SHIQ}b_s$ for an interpretation $\mathcal{I}$ with domain $\Delta^{\mathcal{I}}$

It remains to define the semantics of DL-safe rules. A (DL-safe) *variable assignment Z* for an interpretation $\mathcal{I}$ is a mapping from the set of variables $\mathbf{V}$ to $\{a^{\mathcal{I}} \mid a \in \mathsf{N}_I\}$. Given a term $t \in \mathsf{N}_I \cup \mathbf{V}$, we set $t^{\mathcal{I},Z} := Z(t)$ if $t \in \mathbf{V}$, and $t^{\mathcal{I},Z} := t^{\mathcal{I}}$ otherwise. Given a concept atom $C(t)$ / role atom $R(t, u)$ / equality atom $t \approx u$, we write $\mathcal{I}, Z \models C(t)$ / $\mathcal{I}, Z \models R(t, u)$ / $\mathcal{I}, Z \models t \approx u$ if $t^{\mathcal{I},Z} \in C^{\mathcal{I}}$ / $\langle t^{\mathcal{I},Z}, u^{\mathcal{I},Z}\rangle \in R^{\mathcal{I}}$ / $t^{\mathcal{I},Z} = u^{\mathcal{I},Z}$, and we say that $\mathcal{I}$ and Z *satisfy* the atom in this case.

An interpretation $\mathcal{I}$ *satisfies* a rule $B \to H$ if, for all variable assignments Z for $\mathcal{I}$, either $\mathcal{I}$ and Z satisfy all atoms in H, or $\mathcal{I}$ and Z fail to satisfy some atom in B. In this case, we write $\mathcal{I} \models B \to H$ and say that $\mathcal{I}$ is a *model* for $B \to H$. An interpretation satisfies a rule base $\mathcal{P}$ (i.e., it is a *model* for it) whenever it satisfies all rules in it. An extended knowledge base KB $= \langle \mathcal{T}, \mathcal{R}, \mathcal{P}\rangle$ is *satisfiable* if it has an interpretation $\mathcal{I}$ that is a model for $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{P}$, and it is *unsatisfiable* otherwise. *Satisfiability*, *equivalence*, and *equisatisfiability* of (extended) knowledge bases are defined as usual.

For convenience of notation, we abbreviate TBox axioms of the form $\top \sqsubseteq C$ by writing just C. Statements such as $\mathcal{I} \models C$ and $C \in$ KB are interpreted accordingly. Note that $C \sqsubseteq D$ can thus be written as $\neg C \sqcup D$.

We often need to access a particular set of quantified and atomic subformulae of a DL concept expression. These specific *parts* are provided by the function $P : \mathbf{C} \to 2^{\mathbf{C}}$:

$$P(C) := \begin{cases} P(D) & \text{if } C = \neg D, \\ P(D) \cup P(E) & \text{if } C = D \sqcap E \text{ or } C = D \sqcup E, \\ \{C\} \cup P(D) & \text{if } C = \mathsf{Q}U.D \text{ with } \mathsf{Q} \in \{\exists, \forall, \geqslant n, \leqslant n\}, \\ \{C\} & \text{otherwise.} \end{cases}$$

We generalize P to DL knowledge bases KB by defining $P(\text{KB})$ to be the union of the sets $P(C)$ for all TBox axioms C in KB, where we express TBox axioms as simple concept expressions as explained above.

Given an extended knowledge base KB, we obtain its negation normal form $\mathsf{NNF}(\text{KB})$ by keeping all RBox statements and DL-safe rules untouched and converting every TBox concept C into its negation normal form $\mathsf{NNF}(C)$ in the usual, recursively defined way:

$$
\begin{array}{llll}
\mathsf{NNF}(\neg\top) & := \bot & \mathsf{NNF}(\forall U.C) & := \forall U.\mathsf{NNF}(C) \\
\mathsf{NNF}(\neg\bot) & := \top & \mathsf{NNF}(\neg\forall U.C) & := \exists U.\mathsf{NNF}(\neg C) \\
\mathsf{NNF}(C) & := C \text{ if } C \in \{A, \neg A, \top, \bot\} & \mathsf{NNF}(\exists U.C) & := \exists U.\mathsf{NNF}(C) \\
\mathsf{NNF}(\neg\neg C) & := \mathsf{NNF}(C) & \mathsf{NNF}(\neg\exists U.C) & := \forall U.\mathsf{NNF}(\neg C) \\
\mathsf{NNF}(C \sqcap D) & := \mathsf{NNF}(C) \sqcap \mathsf{NNF}(D) & \mathsf{NNF}(\leqslant n\, R.C) & := \leqslant n\, R.\mathsf{NNF}(C) \\
\mathsf{NNF}(\neg(C \sqcap D)) & := \mathsf{NNF}(\neg C) \sqcup \mathsf{NNF}(\neg D) & \mathsf{NNF}(\neg \leqslant n\, R.C) & := \geqslant (n+1)\, R.\mathsf{NNF}(C) \\
\mathsf{NNF}(C \sqcup D) & := \mathsf{NNF}(C) \sqcup \mathsf{NNF}(D) & \mathsf{NNF}(\geqslant n\, R.C) & := \geqslant n\, R.\mathsf{NNF}(C) \\
\mathsf{NNF}(\neg(C \sqcup D)) & := \mathsf{NNF}(\neg C) \sqcap \mathsf{NNF}(\neg D) & \mathsf{NNF}(\neg \geqslant n\, R.C) & := \leqslant (n-1)\, R.\mathsf{NNF}(C)
\end{array}
$$

It is well known that KB and NNF(KB) are semantically equivalent.

In places, we will additionally require another well-known normalization step that simplifies the structure of KB by *flattening* it to a knowledge base FLAT(KB). This is achieved by transforming KB into negation normal form and exhaustively applying the following transformation rules:

- Select an outermost occurrence of $\mathcal{Q} U.D$ in KB, such that $\mathcal{Q} \in \{\exists, \forall, \leqslant n, \geqslant n\}$ and D is a non-atomic concept.
- Substitute this occurrence with $\mathcal{Q} U.F$ where F is a fresh concept name (i.e., one not occurring in the knowledge base).
- If $\mathcal{Q} \in \{\exists, \forall, \geqslant n\}$, add $\neg F \sqcup D$ to the knowledge base.
- If $\mathcal{Q} = \leqslant n$ add $\mathsf{NNF}(\neg D) \sqcup F$ to the knowledge base.

Obviously, this procedure terminates, yielding a flat knowledge base FLAT(KB) all TBox axioms of which are $\sqcap, \sqcup$-expressions over formulae of the form $\top, \bot, A, \neg A$, or $\mathcal{Q} U.A$ with A an atomic concept name. Flattening is known to be a satisfiability-preserving transformation; we include the proof for the sake of self-containedness.

Proposition 1. *For every $\mathcal{SHIQ}b_s$ knowledge base* KB, *we find that* KB *and* FLAT(KB) *are equi-satisfiable.*

Proof We first prove inductively that every model of FLAT(KB) is a model of KB. Let KB$'$ be an intermediate knowledge base and let KB$''$ be the result of applying one single substitution step to KB$'$ as described in the above procedure. We now show that any model $\mathcal{I}$ of KB$''$ is a model of KB$'$. Let $\mathcal{Q} U.D$ be the concept expression substituted in KB$'$. Note that after every substitution step, the knowledge base is still in negation normal form. Thus, we see that $\mathcal{Q} U.D$ occurs outside the scope of any negation or quantifier in a KB$'$ axiom E', and the same is the case for $\mathcal{Q} U.F$ in the respective KB$''$ axiom E'' obtained after the substitution. Hence, if we show that $(\mathcal{Q} U.F)^{\mathcal{I}} \subseteq (\mathcal{Q} U.D)^{\mathcal{I}}$, we can conclude that $E''^{\mathcal{I}} \subseteq E'^{\mathcal{I}}$. From $\mathcal{I}$ being a model of KB$''$ and therefore $E''^{\mathcal{I}} = \Delta^{\mathcal{I}}$, we would then easily derive that $E'^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and hence find that $\mathcal{I} \models$ KB$'$, as all other axioms from KB$'$ are trivially satisfied due to their presence in KB$''$.

It remains to show $(\mathcal{Q} U.F)^{\mathcal{I}} \subseteq (\mathcal{Q} U.D)^{\mathcal{I}}$. To show this, consider some arbitrary $\delta \in (\mathcal{Q} U.F)^{\mathcal{I}}$. We distinguish various cases:

- $\mathcal{Q} = \geqslant n$
  Then there are distinct individuals $\delta_1, \ldots, \delta_n \in \Delta^{\mathcal{I}}$ with $\langle \delta, \delta_i \rangle \in U^{\mathcal{I}}$ and $\delta_i \in F^{\mathcal{I}}$ for $1 \leq i \leq n$. Since $\neg F \sqcup D \in$ KB$''$, we have $\mathcal{I} \models \neg F \sqcup D$, and therefore $\delta_i \in D^{\mathcal{I}}$ for all the n distinct δ_i. Thus $\delta \in (\geqslant n\, U.F)^{\mathcal{I}}$.
- $\mathcal{Q} = \leqslant n$
  Then the number of individuals $\delta' \in \Delta^{\mathcal{I}}$ with $\langle \delta, \delta' \rangle \in U^{\mathcal{I}}$ and $\delta' \in F^{\mathcal{I}}$ is not greater than n. Since $\mathsf{NNF}(\neg D) \sqcup F \in$ KB$''$, we know $D^{\mathcal{I}} \subseteq F^{\mathcal{I}}$. Thus, also the number of individuals $\delta' \in \Delta^{\mathcal{I}}$ with $\langle \delta, \delta' \rangle \in U^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$ cannot be greater than n, leading to the conclusion $\delta \in (\leqslant n\, U.D)^{\mathcal{I}}$. Hence, we have $(\leqslant n\, U.F)^{\mathcal{I}} \subseteq (\leqslant n\, U.D)^{\mathcal{I}}$.

The arguments for $Q = \exists$ and $Q = \forall$ are very similar, since these cases can be treated like $\geqslant 1\ U.F$ and $\leqslant 0\ U.\neg F$, respectively. Thus we obtain $\delta \in (QU.D)^{\mathcal{I}}$ in each case as required.

For the other direction of the claim, note that every model $\mathcal{I}$ of KB can be transformed into a model $\mathcal{J}$ of FLAT(KB) by following the flattening process described above: Let KB″ result from KB′ by substituting $QU.D$ by $QU.F$ and adding the respective axiom. Furthermore, let $\mathcal{I}'$ be a model of KB′. Now we construct the interpretation $\mathcal{I}''$ as follows: $F^{\mathcal{I}''} := (QU.D)^{\mathcal{I}'}$ and for all other concept and role names N we set $N^{\mathcal{I}''} := N^{\mathcal{I}'}$. Then $\mathcal{I}''$ is a model of KB″. □

3. Building Models from Domino Sets

In this section, we introduce the notion of a set of *dominoes* for a given $\mathcal{ALCI}b$ TBox. Rules (and thus ABox axioms) will be incorporated in Section 5 later on. Intuitively, a domino abstractly represents two individuals in an $\mathcal{ALCI}b$ interpretation, reflecting their satisfied concepts and mutual role relationships. Thereby, dominoes are conceptually very similar to the concept of 2-types, as used in investigations on two-variable fragments of first-order logic, e.g., by Grädel et al. (1997). We will see that suitable sets of such two-element pieces suffice to reconstruct models of $\mathcal{ALCI}b$, which also reveals certain model-theoretic properties of this not so common DL. In particular, every satisfiable $\mathcal{ALCI}b$ TBox admits tree-shaped models. This result is rather a by-product of our main goal of decomposing models into unstructured sets of local domino components, but it explains why our below constructions have some similarity with common approaches of showing tree-model properties by unraveling models.

After introducing the basics of our domino representation, we present an algorithm for deciding satisfiability of an $\mathcal{ALCI}b$ terminology based on sets of dominoes.

3.1. **From Interpretations to Dominoes.** We now introduce the basic notion of a domino set, and its relationship to interpretations. Given a DL with concepts **C** and roles **R**, a *domino* over $\mathcal{C} \subseteq \mathbf{C}$ is an arbitrary triple $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$, where $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ and $\mathcal{R} \subseteq \mathbf{R}$. In the following, we will always assume a fixed language and refer to dominoes over that language only. We now formalize the idea of deconstructing an interpretation into a set of dominoes.

Definition 3. *Given an interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, *and a set* $\mathcal{C} \subseteq \mathbf{C}$ *of concept expressions, the* domino projection *of* $\mathcal{I}$ *w.r.t.* $\mathcal{C}$, *denoted by* $\pi_{\mathcal{C}}(\mathcal{I})$ *is the set that contains, for all* $\delta, \delta' \in \Delta^{\mathcal{I}}$, *the triple* $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ *with*

$$\mathcal{A} = \{C \in \mathcal{C} \mid \delta \in C^{\mathcal{I}}\}, \qquad \mathcal{R} = \{R \in \mathbf{R} \mid \langle \delta, \delta' \rangle \in R^{\mathcal{I}}\}, \qquad \mathcal{B} = \{C \in \mathcal{C} \mid \delta' \in C^{\mathcal{I}}\}.$$

It is easy to see that domino projections do not faithfully represent the structure of the interpretation that they were constructed from. But, as we will see below, domino projections capture enough information to reconstruct models of a TBox $\mathcal{T}$, as long as $\mathcal{C}$ is chosen to contain at least $P(\mathcal{T})$. For this purpose, we introduce the inverse construction of interpretations from arbitrary domino sets.

Definition 4. *Given a set* $\mathbb{D}$ *of dominoes, the* induced *domino interpretation* $\mathcal{I}(\mathbb{D}) = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ *is defined as follows:*

  (1) $\Delta^{\mathcal{I}}$ *consists of all nonempty finite words over* $\mathbb{D}$ *where, for each pair of subsequent letters* $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ *and* $\langle \mathcal{A}', \mathcal{R}', \mathcal{B}' \rangle$ *in a word, we have* $\mathcal{B} = \mathcal{A}'$.
  (2) *For a word* $\sigma = \langle \mathcal{A}_1, \mathcal{R}_1, \mathcal{A}_2 \rangle \langle \mathcal{A}_2, \mathcal{R}_2, \mathcal{A}_3 \rangle \ldots \langle \mathcal{A}_{i-1}, \mathcal{R}_{i-1}, \mathcal{A}_i \rangle$ *and a concept name* $A \in \mathsf{N}_C$, *we define* $tail(\sigma) := \mathcal{A}_i$ *and set* $\sigma \in A^{\mathcal{I}}$ *iff* $A \in tail(\sigma)$.
  (3) *For a role name* $R \in \mathsf{N}_R$, *we set* $\langle \sigma_1, \sigma_2 \rangle \in R^{\mathcal{I}}$ *if*

$$\sigma_2 = \sigma_1 \langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle \text{ with } R \in \mathcal{R} \quad \text{or} \quad \sigma_1 = \sigma_2 \langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle \text{ with } \mathrm{Inv}(R) \in \mathcal{R}.$$

We can now show that certain domino projections contain enough information to reconstruct models of a TBox.

Proposition 2. *Consider a set $\mathcal{C} \subseteq \mathbf{C}$ of concept expressions, and an interpretation $\mathcal{J}$, and let $\mathcal{K} := \mathcal{I}(\pi_{\mathcal{C}}(\mathcal{J}))$ denote the induced domino interpretation of the domino projection of $\mathcal{J}$ w.r.t. $\mathcal{C}$. Then, for any $\mathcal{ALCIb}$ concept expression $C \in \mathbf{C}$ with $P(C) \subseteq \mathcal{C}$, we have that $\mathcal{J} \models C$ iff $\mathcal{K} \models C$. Especially, for any $\mathcal{ALCIb}$ TBox $\mathcal{T}$, we have $\mathcal{J} \models \mathcal{T}$ iff $\mathcal{I}(\pi_{P(\mathcal{T})}(\mathcal{J})) \models \mathcal{T}$.*

Proof Consider some $C \in \mathbf{C}$ as in the claim. We first show the following: given any $\mathcal{J}$-individual δ and $\mathcal{K}$-individual σ such that $\text{tail}(\sigma) = \{D \in \mathcal{C} \mid \delta \in D^{\mathcal{J}}\}$, we find that $\sigma \in C^{\mathcal{K}}$ iff $\delta \in C^{\mathcal{J}}$. Clearly, the overall claim follows from that statement using the observation that a suitable $\delta \in \Delta^{\mathcal{J}}$ must exist for all $\sigma \in \Delta^{\mathcal{K}}$ and vice versa. We proceed by induction over the structure of C, noting that $P(C) \subseteq \mathcal{C}$ implies $P(D) \subseteq \mathcal{C}$ for any subconcept D of C.
The base case $C \in \mathsf{N}_C$ is immediately satisfied by our assumption on the relationship of δ and σ, since $C \in P(C)$. For the induction step, we first note that the case $C \in \{\top, \bot\}$ is also trivial. For $C = \neg D$ and $C = D \sqcap D'$ as well as $C = D \sqcup D'$, the claim follows immediately from the induction hypothesis for D and D'.
Next consider the case $C = \exists U.D$, and assume that $\delta \in C^{\mathcal{J}}$. Hence there is some $\delta' \in \Delta^{\mathcal{J}}$ such that $\langle \delta, \delta' \rangle \in U^{\mathcal{J}}$ and $\delta' \in D^{\mathcal{J}}$. Then the pair $\langle \delta, \delta' \rangle$ generates a domino $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ and $\Delta^{\mathcal{K}}$ contains $\sigma' = \sigma \langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$. $\langle \delta, \delta' \rangle \in U^{\mathcal{J}}$ implies $\mathcal{R} \vdash U$ (by definition of $\vdash$ and due to the fact that $\mathcal{R}$ contains exactly those $R \in \mathbf{R}$ with $\langle \delta, \delta' \rangle \in R^{\mathcal{J}}$), and hence $\langle \sigma, \sigma' \rangle \in U^{\mathcal{K}}$. Applying the induction hypothesis to D, we conclude $\sigma' \in D^{\mathcal{K}}$. Now $\sigma \in C^{\mathcal{K}}$ follows from the construction of $\mathcal{K}$.
For the converse, assume that $\sigma \in C^{\mathcal{K}}$. Hence there is some $\sigma' \in \Delta^{\mathcal{K}}$ such that $\langle \sigma, \sigma' \rangle \in U^{\mathcal{K}}$ and $\sigma' \in D^{\mathcal{K}}$. By the definition of $\mathcal{K}$, there are two possible cases:

- $\sigma' = \sigma \langle \text{tail}(\sigma), \mathcal{R}, \text{tail}(\sigma') \rangle$ and $\mathcal{R} \vdash U$: Consider the two $\mathcal{J}$-individuals $\langle \delta', \delta'' \rangle$ generating the domino $\langle \text{tail}(\sigma), \mathcal{R}, \text{tail}(\sigma') \rangle$. From $\sigma' \in D^{\mathcal{K}}$ and the induction hypothesis, we obtain $\delta'' \in D^{\mathcal{J}}$. Together with $\langle \delta', \delta'' \rangle \in U^{\mathcal{J}}$ this implies $\delta' \in C^{\mathcal{J}}$. Since $C = \exists U.D \in \mathcal{C}$, we also have $C \in \text{tail}(\sigma)$ and thus $\delta \in C^{\mathcal{J}}$ as claimed.
- $\sigma = \sigma' \langle \text{tail}(\sigma'), \mathcal{R}, \text{tail}(\sigma) \rangle$ and $\text{Inv}(\mathcal{R}) \vdash U$: This case is similar to the first case, merely exchanging the order of $\langle \delta', \delta'' \rangle$ and using $\text{Inv}(\mathcal{R})$ instead of $\mathcal{R}$.

Finally, the case $C = \forall U.D$ is dual to the case $C = \exists U.D$, and we will omit the repeated argument. Note, however, that this case does not follow from the semantic equivalence of $\forall U.D$ and $\neg \exists U.\neg D$, since the proof hinges upon the fact that $\neg D$ is contained in $\mathcal{C}$ which is not given directly. $\qquad \square$

3.2. **Constructing Domino Sets.** As shown in the previous section, the domino projection of a model of an $\mathcal{ALCIb}$ TBox can contain enough information for reconstructing a model. This observation can be the basis for designing an algorithm that decides TBox satisfiability. Usually (especially in tableau-based algorithms), checking satisfiability amounts to the attempt to construct a (representation of a) model. As we have seen, in our case it suffices to try to construct just a model's domino projection. If this can be done, we know that there is a model, if not, there is none.
In what follows, we first describe the iterative construction of such a domino set from a given TBox, and then show that it is indeed a decision procedure for TBox satisfiability.
Algorithm 1 describes the construction of the canonical domino set $\mathbb{D}_{\mathcal{T}}$ of an $\mathcal{ALCIb}$ TBox $\mathcal{T}$. Thereby, roughly speaking, condition **kb** ensures that all the concept parts $\mathcal{A}$ and $\mathcal{B}$ of the constructed domino set abide by the axioms of the considered TBox. The condition **ex** guarantees that,

---

[7]Please note that the formulae in $\mathsf{FLAT}(\mathcal{T})$ and in $\mathcal{A} \subseteq \mathcal{C}$ are such that this can easily be checked by evaluating the Boolean operators in C as if $\mathcal{A}$ was a set of true propositional variables.

Algorithm 1 Computing the canonical domino set $\mathbb{D}_{\mathcal{T}}$ of a TBox $\mathcal{T}$

Input: $\mathcal{T}$ an $\mathcal{ALCI}b$ TBox, $\mathcal{C} = P(\mathsf{FLAT}(\mathcal{T}))$
Output: the canonical domino set $\mathbb{D}_{\mathcal{T}}$ of $\mathcal{T}$
1: initialize $\mathbb{D}_0$ as the set of all dominoes $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ over $\mathcal{C}$ satisfying:
2:      for all $C \in \mathsf{FLAT}(\mathcal{T})$, the GCI $\sqcap_{D \in \mathcal{A}} D \sqcap \sqcap_{D \in \mathcal{C} \setminus \mathcal{A}} \neg D \sqsubseteq C$ is a tautology[7]   **(kb)**
3:      for all $\exists U.A \in \mathcal{C}$ with $A \in \mathcal{B}$ and $\mathcal{R} \vdash U$, we have $\exists U.A \in \mathcal{A}$,   **(ex)**
4:      for all $\forall U.A \in \mathcal{C}$ with $\forall U.A \in \mathcal{A}$ and $\mathcal{R} \vdash U$, we have $A \in \mathcal{B}$.   **(uni)**
5: $i := 0$
6: **repeat**
7:    $i := i+1$
8:    determine $\mathbb{D}_i$ as the set of all dominoes $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle \in \mathbb{D}_{i-1}$ satisfying:
9:      for all $\exists U.A \in \mathcal{A}$, there is some $\langle \mathcal{A}, \mathcal{R}', \mathcal{B}' \rangle \in \mathbb{D}_{i-1}$ with $\mathcal{R}' \vdash U$ and $A \in \mathcal{B}'$,   **(delex)**
10:     for all $\forall U.A \in \mathcal{C} \setminus \mathcal{A}$, there is some $\langle \mathcal{A}, \mathcal{R}', \mathcal{B}' \rangle \in \mathbb{D}_{i-1}$ with $\mathcal{R}' \vdash U$ but $A \notin \mathcal{B}'$,   **(deluni)**
11:     $\langle \mathcal{B}, \mathsf{Inv}(\mathcal{R}), \mathcal{A} \rangle \in \mathbb{D}_{i-1}$.   **(sym)**
12: **until** $\mathbb{D}_i = \mathbb{D}_{i-1}$
13: $\mathbb{D}_{\mathcal{T}} := \mathbb{D}_i$
14: **return** $\mathbb{D}_{\mathcal{T}}$

---

in every domino $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$, the concept set $\mathcal{A}$ must contain all the existential concepts for which $\mathcal{R}$ and $\mathcal{B}$ serve as witnesses. Conversely, **uni** makes sure that every universally quantified concept recorded in $\mathcal{A}$ is appropriately propagated to $\mathcal{B}$, given a suitable $\mathcal{R}$. Once enforced, the conditions **kb**, **ex**, and **uni** remain valid even if the domino set is reduced further, hence they need to be taken care of only at the beginning of the algorithm. In contrast, the conditions **delex**, **deluni**, and **sym** may be invalidated again by removing dominoes from the set, thus they need to be applied in an iterated way until a fixpoint is reached. Condition **delex** removes all dominoes with the concept set $\mathcal{A}$ if $\mathcal{A}$ contains an existential concept for which no appropriate "witness" domino (in the above sense) can be found in the set. Likewise, **deluni** removes all dominoes with the concept set $\mathcal{A}$ if $\mathcal{A}$ does *not* contain a universal concept which should hold given all the remaining dominoes. Finally, **sym** ensures that the domino set contains only dominoes that do have a "symmetric partner", i.e., one that is created by swapping $\mathcal{A}$ with $\mathcal{B}$ and inverting all of $\mathcal{R}$.

Given that every domino $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ satisfies $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ and $\mathcal{R} \subseteq \mathbf{R}$, and that both $\mathcal{C}$ and $\mathbf{R}$ are linearly bounded by the size of $\mathcal{T}$, $\mathbb{D}_0$ is exponential in the size of the TBox, hence the iterative deletion of dominoes must terminate after at most exponentially many steps. Below we will show that this procedure is indeed sound and complete for checking TBox satisfiability. Before that, we will show a canonicity result for $\mathbb{D}_{\mathcal{T}}$.

Lemma 3. *Consider an $\mathcal{ALCI}b$ terminology $\mathcal{T}$ and an arbitrary model $\mathcal{I}$ of $\mathcal{T}$. Then the domino projection $\pi_{P(\mathsf{FLAT}(\mathcal{T}))}(\mathcal{I})$ is contained in $\mathbb{D}_{\mathcal{T}}$.*

Proof. The claim is shown by a simple induction over the construction of $\mathbb{D}_{\mathcal{T}}$. In the following, we use $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ to denote an arbitrary domino of $\pi_{P(\mathsf{FLAT}(\mathcal{T}))}(\mathcal{I})$. For the base case, we must show that $\pi_{P(\mathsf{FLAT}(\mathcal{T}))}(\mathcal{I}) \subseteq \mathbb{D}_0$. Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ to denote an arbitrary domino of $\pi_{P(\mathsf{FLAT}(\mathcal{T}))}(\mathcal{I})$ which was generated from elements $\langle \delta, \delta' \rangle$. Then $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ satisfies condition **kb**, since $\delta \in C^{\mathcal{I}}$ for any $C \in \mathsf{FLAT}(\mathcal{T})$. The conditions **ex** and **uni** are obviously satisfied.

For the induction step, assume that $\pi_{P(\mathsf{FLAT}(\mathcal{T}))}(\mathcal{I}) \subseteq \mathbb{D}_i$, and let $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ again denote an arbitrary domino of $\pi_{P(\mathsf{FLAT}(\mathcal{T}))}(\mathcal{I})$ which was generated from elements $\langle \delta, \delta' \rangle$.

- For **delex**, note that $\exists U.A \in \mathcal{A}$ implies $\delta \in (\exists U.A)^{\mathcal{I}}$. Thus there is an individual δ'' such that $\langle \delta, \delta'' \rangle \in U^{\mathcal{I}}$ and $\delta'' \in A^{\mathcal{I}}$. Clearly, the domino generated by $\langle \delta, \delta'' \rangle$ satisfies the conditions of **delex**.
- For **deluni**, note that $\forall U.A \notin \mathcal{A}$ implies $\delta \notin (\forall U.A)^{\mathcal{I}}$. Thus there is an individual δ'' such that $\langle \delta, \delta'' \rangle \in U^{\mathcal{I}}$ and $\delta'' \notin A^{\mathcal{I}}$. Clearly, the domino generated by $\langle \delta, \delta'' \rangle$ satisfies the conditions of **deluni**.
- The condition of **sym** for $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ is clearly satisfied by the domino generated from $\langle \delta', \delta \rangle$.

Therefore, the considered domino $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ must be contained in $\mathbb{D}_{i+1}$ as well. □

Note that, in contrast to tableau procedures, the presented algorithm starts with a large set of dominoes and successively deletes undesired dominoes. Indeed, we will soon show that the constructed domino set is the largest such set from which a domino model can be obtained. The algorithm thus may seem to be of little practical use. In Section 4, we therefore refine the above algorithm to employ Boolean functions as implicit representations of domino sets, such that the efficient computational methods of OBDDs can be exploited. In the meantime, however, domino sets will serve us well for showing the required correctness properties.

An important property of domino interpretations constructed from canonical domino sets is that the (semantic) concept membership of an individual can typically be (syntactically) read from the domino it has been constructed of.

Lemma 4. *Consider an $\mathcal{ALCIb}$ TBox $\mathcal{T}$ with nonempty canonical domino set $\mathbb{D}_\mathcal{T}$, and define $\mathcal{C} := P(\mathsf{FLAT}(\mathcal{T}))$ and $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle := \mathcal{I}(\mathbb{D}_\mathcal{T})$. Then, for all $C \in \mathcal{C}$ and $\sigma \in \Delta^{\mathcal{I}}$, we have that $\sigma \in C^{\mathcal{I}}$ iff $C \in \mathrm{tail}(\sigma)$. Moreover, $\mathcal{I} \models \mathsf{FLAT}(\mathcal{T})$.*

Proof First note that the domain of $\mathcal{I}$ is nonempty whenever $\mathbb{D}_\mathcal{T}$ is. Now if $C \in \mathsf{N}_C$ is an atomic concept, the first claim follows directly from the definition of $\mathcal{I}$. The remaining cases that may occur in $P(\mathsf{FLAT}(\mathcal{T}))$ are $C = \exists U.A$ and $C = \forall U.A$.

First consider the case $C = \exists U.A$, and assume that $\sigma \in C^{\mathcal{I}}$. Thus there is $\sigma' \in \Delta^{\mathcal{I}}$ with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \in A^{\mathcal{I}}$. The construction of the domino model admits two possible cases:

- $\sigma' = \sigma \langle \mathrm{tail}(\sigma), \mathcal{R}, \mathrm{tail}(\sigma') \rangle$ with $\mathcal{R} \vdash U$ and $A \in \mathrm{tail}(\sigma')$. Since $\mathbb{D}_\mathcal{T} \subseteq \mathbb{D}_0$, we find that $\langle \mathrm{tail}(\sigma), \mathcal{R}, \mathrm{tail}(\sigma') \rangle$ satisfies condition **ex**, and thus $C \in \mathrm{tail}(\sigma)$ as required.
- $\sigma = \sigma' \langle \mathrm{tail}(\sigma'), \mathcal{R}, \mathrm{tail}(\sigma) \rangle$ with $\mathrm{Inv}(\mathcal{R}) \vdash U$ and $A \in \mathrm{tail}(\sigma')$. By condition **sym**, $\mathbb{D}_\mathcal{T}$ also contains the domino $\langle \mathrm{tail}(\sigma), \mathrm{Inv}(\mathcal{R}), \mathrm{tail}(\sigma') \rangle$, and we can again invoke **ex** to conclude $C \in \mathrm{tail}(\sigma)$.

For the other direction, assume $\exists U.A \in \mathrm{tail}(\sigma)$. Thus $\mathbb{D}_\mathcal{T}$ must contain some domino $\langle \mathcal{A}, \mathcal{R}, \mathrm{tail}(\sigma) \rangle$, and by **sym** also the domino $\langle \mathrm{tail}(\sigma), \mathrm{Inv}(\mathcal{R}), \mathcal{A} \rangle$. By condition **delex**, the latter implies that $\mathbb{D}_\mathcal{T}$ contains a domino $\langle \mathrm{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$. According to **delex**, we find that $\sigma' = \sigma \langle \mathrm{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$ is an $\mathcal{I}$-individual such that $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \in A^{\mathcal{I}}$. Thus $\sigma \in (\exists U.A)^{\mathcal{I}}$ as claimed.

For the second case, consider $C = \forall U.A$ and assume that $\sigma \in C^{\mathcal{I}}$. Then $\mathbb{D}_\mathcal{T}$ contains some domino $\langle \mathcal{A}, \mathcal{R}, \mathrm{tail}(\sigma) \rangle$, and by **sym** also the domino $\langle \mathrm{tail}(\sigma), \mathrm{Inv}(\mathcal{R}), \mathcal{A} \rangle$. For a contradiction, suppose that $\forall U.A \notin \mathrm{tail}(\sigma)$. By condition **deluni**, the latter implies that $\mathbb{D}_\mathcal{T}$ contains a domino $\langle \mathrm{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$. According to **deluni**, we find that $\sigma' = \sigma \langle \mathrm{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$ is an $\mathcal{I}$-individual such that $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \notin D^{\mathcal{I}}$. But then $\sigma \notin (\forall U.A)^{\mathcal{I}}$, yielding the required contradiction.

For the other direction, assume that $\forall U.A \in \mathrm{tail}(\sigma)$. According to the construction of the domino model, there are two possible cases for elements σ' with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$:

- $\sigma' = \sigma \langle \mathrm{tail}(\sigma), \mathcal{R}, \mathrm{tail}(\sigma') \rangle$ with $\mathcal{R} \vdash U$. Since $\mathbb{D}_\mathcal{T} \subseteq \mathbb{D}_0$, $\langle \mathrm{tail}(\sigma), \mathcal{R}, \mathrm{tail}(\sigma') \rangle$ must satisfy condition **uni**, and thus $A \in \mathrm{tail}(\sigma')$.

- $\sigma = \sigma'\langle\text{tail}(\sigma'), \mathcal{R}, \text{tail}(\sigma)\rangle$ with $\text{Inv}(\mathcal{R}) \vdash U$. By condition **sym**, $\mathbb{D}_{\mathcal{T}}$ also contains the domino $\langle\text{tail}(\sigma), \text{Inv}(\mathcal{R}), \text{tail}(\sigma')\rangle$, and we can again invoke **uni** to conclude $A \in \text{tail}(\sigma')$.

Thus, $A \in \text{tail}(\sigma')$ for all U-successors σ' of σ, and hence $\sigma \in (\forall U.A)^{\mathcal{I}}$ as claimed.

For the rest of the claim, note that any domino $\langle\mathcal{A}, \mathcal{R}, \mathcal{B}\rangle$ must satisfy condition **kb**. Using condition **sym**, we conclude that for any $\sigma \in \Delta^{\mathcal{I}}$, the axiom $\bigsqcap_{D\in\text{tail}(\sigma)} D \sqsubseteq C$ is a tautology for all $C \in$ FLAT($\mathcal{T}$). As shown above, $\sigma \in D^{\mathcal{I}}$ for all $D \in \text{tail}(\sigma)$, and thus $\sigma \in C$. Hence every individual of $\mathcal{I}$ is an instance of each concept of FLAT($\mathcal{T}$) as required. $\qquad\square$

The previous lemma shows soundness of our decision algorithm. Conversely, completeness is shown by the following lemma.

Lemma 5. *Consider an $\mathcal{ALCI}b$ TBox $\mathcal{T}$. If $\mathcal{T}$ is satisfiable, then its canonical domino set $\mathbb{D}_{\mathcal{T}}$ is nonempty.*

Proof This is a straightforward consequence of Lemma 3: given a model $\mathcal{I}$ of $\mathcal{T}$, the domino projection $\pi_{P(\text{FLAT}(\mathcal{T}))}(\mathcal{I})$ is nonempty and (by Lemma 3) contained in $\mathbb{D}_{\mathcal{T}}$. Hence $\mathbb{D}_{\mathcal{T}}$ is nonempty. $\qquad\square$

We now are ready to establish our main result on checking TBox satisfiability and the complexity of the given algorithm:

Theorem 6. *An $\mathcal{ALCI}b$ TBox $\mathcal{T}$ is satisfiable iff its canonical domino set $\mathbb{D}_{\mathcal{T}}$ is nonempty. Algorithm 1 thus describes a decision procedure for satisfiability of $\mathcal{ALCI}b$ TBoxes. Moreover, the algorithm runs in exponential time and hence is worst-case optimal.*

Proof The first proposition of the theorem is a direct consequence of Lemma 4, Proposition 1 (page 6), and Lemma 5.

For worst-case optimality, recall that $\mathcal{SHIQ}b_s$ is ExpTime-complete (see Rudolph et al., 2008a, where ExpTime-hardness already directly follows from the results by Schild, 1991). Now, considering the presented algorithm, we find that the set $\mathcal{C} = P(\text{FLAT}(\mathcal{T}))$ is linearly bounded by the size of $\mathcal{T}$, whence the size of the set of all dominoes is exponentially bounded by $|\mathcal{T}|$. Applying the conditions **kb**, **ex**, and **uni** to obtain $\mathbb{D}_0$ can be done by subsequently checking every domino, each check taking at most $O(|\mathcal{T}|)$ time, hence the overall time for that step is exponentially bounded. Now, consider the iterated application of the **delex**, **deluni**, and **sym** conditions. By the same argumentation as for **kb**, **ex**, and **uni**, one iteration takes exponential time. On the other hand, each iteration step reduces the domino set by at least one domino (otherwise, the termination criterion would be satisfied) which gives us a bound of exponentially many steps. Finally note that exponentially many exponentially long steps still yield a procedure that is overall exponentially bounded. $\qquad\square$

4. Sets as Boolean Functions

The algorithm of the previous section may seem to be of little practical use, since it requires computations on an exponentially large set of dominoes. The required computation steps, however, can also be accomplished with an indirect representation of the possible dominoes based on Boolean functions. Indeed, every propositional logic formula represents a set of propositional interpretations for which the function evaluates to *true*. Using a suitable encoding, each propositional interpretation can be understood as a domino, and a propositional formula can represent a domino set.

As a representation of propositional formulae well-proven in other contexts, we use binary decision diagrams (BDDs). These data structures have been used to represent complex Boolean functions in model-checking (see, e.g., Burch et al., 1990). A particular optimization of these structures are

ordered BDDs (OBDDs) that use a dynamic precedence order of propositional variables to obtain compressed representations. We provide a first introduction to OBDDs below. A more detailed exposition and further literature pointers are given by Huth and Ryan (2000).

4.1. **Boolean Functions and Operations.** We first explain how sets can be represented by means of Boolean functions. This will enable us, given a fixed finite base set S, to represent every family of sets $\mathbb{S} \subseteq 2^S$ by a single Boolean function.

A *Boolean function* on a set Var of variables is a function $\varphi : 2^{\mathsf{Var}} \to \{true, false\}$. The underlying intuition is that $\varphi(V)$ computes the truth value of a Boolean formula based on the assumption that exactly the variables of V are set to *true*. A simple example are the functions $[\![true]\!]$ and $[\![false]\!]$, that map every input to *true* or *false*, respectively. Another example are so-called *characteristic functions* of the form $[\![v]\!]_\chi$ for some $v \in \mathsf{Var}$, which are defined as $[\![v]\!]_\chi(V) := true$ iff $v \in V$.

Boolean functions over the same set of variables can be combined and modified in several ways. Especially, there are the obvious Boolean operators for negation, conjunction, disjunction, and implication. By slight abuse of notation, we will use the common (syntactic) operator symbols $\neg$, $\wedge$, $\vee$, and $\to$ to also represent such (semantic) operators on Boolean functions. Given, e.g., Boolean functions φ and ψ, we find that $(\varphi \wedge \psi)(V) = true$ iff $\varphi(V) = true$ and $\psi(V) = true$. Note that the result of the application of $\wedge$ results in another Boolean function, and is not to be understood as a syntactic logical formula.

Another operation on Boolean functions is existential quantification over a set of variables $V \subseteq \mathsf{Var}$, written as $\exists V.\varphi$ for some function φ. Given an input set $W \subseteq \mathsf{Var}$ of variables, we define $(\exists V.\varphi)(W) = true$ iff *there is some* $V' \subseteq V$ such that $\varphi(V' \cup (W \setminus V)) = true$. In other words, there must be a way to set truth values of variables in V such that φ evaluates to *true*. Universal quantification is defined analogously, and we thus have $\forall V.\varphi := \neg\exists V.\neg\varphi$ as usual. Mark that our use of $\exists$ and $\forall$ overloads notation, and should not be confused with role restrictions in DL expressions.

4.2. **Ordered Binary Decision Diagrams.** Binary Decision Diagrams (BDDs), intuitively speaking, are a generalization of decision trees that allows for the reuse of nodes. Structurally, BDDs are directed acyclic graphs whose nodes are labeled by variables from some set Var. The only exception are two *terminal* nodes that are labeled by *true* and *false*, respectively. Every non-terminal node has two outgoing edges, corresponding to the two possible truth values of the variable.

Definition 5. *A BDD is a tuple* $\mathbb{O} = \langle N, n_{\mathsf{root}}, n_{\mathsf{true}}, n_{\mathsf{false}}, \mathsf{low}, \mathsf{high}, \mathsf{Var}, \lambda \rangle$ *where*

- *N is a finite set called* nodes*,*
- *$n_{\mathsf{root}} \in N$ is called the* root *node,*
- *$n_{\mathsf{true}}, n_{\mathsf{false}} \in N$ are called the* terminal *nodes,*
- $\mathsf{low}, \mathsf{high} : N \setminus \{n_{\mathsf{true}}, n_{\mathsf{false}}\} \to N$ *are two* child functions *assigning to every non-terminal node a* low *and a* high *child node. Furthermore the graph obtained by iterated application has to be acyclic, i.e., for no node n exists a sequence of applications of* low *and* high *resulting in n again.*
- Var *is a finite set of* variables*.*
- $\lambda : N \setminus \{n_{\mathsf{true}}, n_{\mathsf{false}}\} \to \mathsf{Var}$ *is the* labeling function *assigning to every non-terminal node a variable from* Var*.*

OBDDs are a particular realization of BDDs where a certain ordering is imposed on variables to achieve more efficient representations. We will not require to consider the background of this optimization in here. Every BDD based on a variable set $\mathsf{Var} = \{x_1, \ldots, x_n\}$ represents an *n*-ary Boolean function $\varphi : 2^{\mathsf{Var}} \to \{true, false\}$.

Definition 6. *Given a BDD $\mathbb{O} = \langle N, n_{\mathsf{root}}, n_{\mathsf{true}}, n_{\mathsf{false}}, \mathsf{low}, \mathsf{high}, \mathsf{Var}, \lambda \rangle$ the Boolean function $\varphi_{\mathbb{O}}$: $2^{\mathsf{Var}} \to \{true, false\}$ is defined recursively as follows:*

$$\varphi_{\mathbb{O}} := \varphi_{n_{\mathsf{root}}} \qquad \varphi_{n_{\mathsf{true}}} = [\![true]\!] \qquad \varphi_{n_{\mathsf{false}}} = [\![false]\!]$$

$$\varphi_n = \left(\neg [\![\lambda(n)]\!]_\chi \wedge \varphi_{\mathsf{low}(n)} \right) \vee \left([\![\lambda(n)]\!]_\chi \wedge \varphi_{\mathsf{high}(n)} \right) \quad for\ n \in N \setminus \{n_{\mathsf{true}}, n_{\mathsf{false}}\}$$

In other words, the value $\varphi(V)$ for some $V \subseteq \mathsf{Var}$ is determined by traversing the BDD, starting from the root node: at a node labeled with $v \in \mathsf{Var}$, the evaluation proceeds with the node connected by the high-edge if $v \in V$, and with the node connected by the low-edge otherwise. If a terminal node is reached, its label is returned as a result.

BDDs for some Boolean formulas might be exponentially large in general (compared to $|\mathsf{Var}|$), but often there is a representation which allows for BDDs of manageable size. Finding the optimal representation is NP-complete, but heuristics have shown to yield good approximate solutions (Wegener, 2004). Hence (O)BDDs are often conceived as efficiently compressed representations of Boolean functions. In addition, many operations on Boolean functions – such as the aforementioned negation, conjunction, disjunction, implication as well as propositional quantification – can be performed directly on the corresponding OBDDs by fast algorithms.

4.3. **Translating Dominos into Boolean Functions.**

To apply the above machinery to DL reasoning, consider a flattened $\mathcal{ALCIb}$ TBox $\mathcal{T} = \mathsf{FLAT}(\mathcal{T})$. A set of propositional variables Var is defined as $\mathsf{Var} := \mathbf{R} \cup (P(\mathcal{T}) \times \{1, 2\})$. We thus obtain a bijection between dominoes over the set $P(\mathcal{T})$ and sets $V \subseteq \mathsf{Var}$ given by $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle \mapsto (\mathcal{A} \times \{1\}) \cup \mathcal{R} \cup (\mathcal{B} \times \{2\})$. Hence, any Boolean function over Var represents a domino set as the collection of all variable sets for which it evaluates to *true*. We can use this observation to rephrase the construction of $\mathbb{D}_{\mathcal{T}}$ in Algorithm 1 into an equivalent construction of a function $[\![\mathcal{T}]\!]$.

We first represent DL concepts C and role expressions U by characteristic Boolean functions over Var as follows.

$$[\![C]\!] := \begin{cases} \neg [\![D]\!] & \text{if } C = \neg D \\ [\![D]\!] \wedge [\![E]\!] & \text{if } C = D \sqcap E \\ [\![D]\!] \vee [\![E]\!] & \text{if } C = D \sqcup E \\ [\![\langle C, 1 \rangle]\!]_\chi & \text{if } C \in P(\mathcal{T}) \end{cases} \qquad [\![U]\!] := \begin{cases} \neg [\![V]\!] & \text{if } U = \neg V \\ [\![V]\!] \wedge [\![W]\!] & \text{if } U = V \sqcap W \\ [\![V]\!] \vee [\![W]\!] & \text{if } U = V \sqcup W \\ [\![U]\!]_\chi & \text{if } U \in \mathbf{R} \end{cases}$$

We can now define a decision procedure based on Boolean functions, as displayed in Algorithm 2. This algorithm is an accurate translation of Algorithm 1, where the intermediate Boolean functions $\varphi^{\mathbf{kb}}, \varphi^{\mathbf{ex}}, \varphi^{\mathbf{uni}}, \varphi_i^{\mathbf{delex}}, \varphi_i^{\mathbf{deluni}}, \varphi_i^{\mathbf{sym}}$ represent domino sets containing all dominoes satisfying the respective conditions from Algorithm 1. By computing their conjunction with each other (and, for the latter three, with the Boolean function representing the domino set from the previous iteration) we intersect the respective domino sets which results in their successive pruning as described in Algorithm 1. The algorithm is a correct procedure for checking consistency of $\mathcal{ALCIb}$ TBoxes as unsatisfiability of $\mathcal{T}$ coincides with $[\![\mathcal{T}]\!] \equiv false$. Note that all necessary computation steps can indeed be implemented algorithmically: Any Boolean function can be evaluated for a fixed variable input V, and equality of two functions can (naively) be checked by comparing the results for all possible input sets (which are finitely many since Var is finite). The algorithm terminates since the sequence is decreasing w.r.t. $\{V \mid [\![\mathcal{T}]\!]_i(V) = true\}$, and since there are only finitely many Boolean functions over Var.

Algorithm 2 Computing the boolean representation $[\![\mathcal{T}]\!]$ of the canonical domino set $\mathbb{D}_{\mathcal{T}}$ of a TBox

---

Input: $\mathcal{T}$ an $\mathcal{ALCIb}$ TBox, $\mathcal{C} = P(\mathsf{FLAT}(\mathcal{T}))$
Output: the canonical domino set of $\mathcal{T}$, represented as Boolean function $[\![\mathcal{T}]\!]$

1: $\varphi^{\mathbf{kb}}$ $:= \bigwedge_{C \in \mathcal{T}} [\![C]\!]$

2: $\varphi^{\mathbf{uni}}$ $:= \bigwedge_{\forall U.C \in P(\mathcal{T})} [\![\langle \forall U.C, 1 \rangle]\!]_{\chi} \wedge [\![U]\!] \rightarrow [\![\langle C, 2 \rangle]\!]_{\chi}$

3: $\varphi^{\mathbf{ex}}$ $:= \bigwedge_{\exists U.C \in P(\mathcal{T})} [\![\langle C, 2 \rangle]\!]_{\chi} \wedge [\![U]\!] \rightarrow [\![\langle \exists U.C, 1 \rangle]\!]_{\chi}$

4: $[\![\mathcal{T}]\!]_0 := \varphi^{\mathbf{kb}} \wedge \varphi^{\mathbf{uni}} \wedge \varphi^{\mathbf{ex}}$

5: i := 0

6: **repeat**

7:      i := i+1

8:      $\varphi_i^{\mathbf{delex}}$ $:= \bigwedge_{\exists U.C \in P(\mathcal{T})} [\![\langle \exists U.C, 1 \rangle]\!]_{\chi} \rightarrow \exists (\mathbf{R} \cup \mathcal{C} \times \{2\}).([\![\mathcal{T}]\!]_{i-1} \wedge [\![U]\!] \wedge [\![\langle C, 2 \rangle]\!]_{\chi})$

9:      $\varphi_i^{\mathbf{deluni}}$ $:= \bigwedge_{\forall U.C \in P(\mathcal{T})} [\![\langle \forall U.C, 1 \rangle]\!]_{\chi} \rightarrow \neg\exists (\mathbf{R} \cup \mathcal{C} \times \{2\}).([\![\mathcal{T}]\!]_{i-1} \wedge [\![U]\!] \wedge \neg[\![\langle C, 2 \rangle]\!]_{\chi})$

10:     $\varphi_i^{\mathbf{sym}}(V) := [\![\mathcal{T}]\!]_{i-1}\big(\{\langle D, 1 \rangle \mid \langle D, 2 \rangle \in V\} \cup \{\mathrm{Inv}(R) \mid R \in V\} \cup \{\langle D, 2 \rangle \mid \langle D, 1 \rangle \in V\}\big)$

11:     $[\![\mathcal{T}]\!]_i := [\![\mathcal{T}]\!]_{i-1} \wedge \varphi_i^{\mathbf{delex}} \wedge \varphi_i^{\mathbf{deluni}} \wedge \varphi_i^{\mathbf{sym}}$

12: **until** $[\![\mathcal{T}]\!]_i \equiv [\![\mathcal{T}]\!]_{i-1}$

13: $[\![\mathcal{T}]\!] := [\![\mathcal{T}]\!]_i$

14: **return** $[\![\mathcal{T}]\!]$

Proposition 7. *For any $\mathcal{ALCIb}$ TBox $\mathcal{T}$ and variable set $V \in \mathsf{Var}$ as above, we find that $[\![\mathcal{T}]\!](V) = true$ iff V represents a domino in $\mathbb{D}_{\mathcal{T}}$ as defined in Definition 1.*

Proof It is easy to see that the Boolean operations used in constructing $[\![\mathcal{T}]\!]$ directly correspond to the set operations in Definition 1, such that $[\![\mathcal{T}]\!](V) = true$ iff V represents a domino in $\mathbb{D}_{\mathrm{KB}}$. $\square$

All required operations and checks are provided by standard OBDD implementations, and thus can be realized in practice.

In the remainder of this section, we illustrate the above algorithm by an extended example to which we will also come back to explain the later extensions of the inference algorithm. Therefore, consider the following $\mathcal{ALCIb}$ knowledge base $\mathcal{KB}$.

$$
\begin{aligned}
PhDStudent &\sqsubseteq \exists has.Diploma \\
Diploma &\sqsubseteq \forall has^-.Graduate \\
Diploma \sqcap Graduate &\sqsubseteq \bot \\
Diploma(laureus) \quad & \quad PhDStudent(laureus)
\end{aligned}
$$

For now, we are only interested in the terminological axioms, the consistency of which we would like to establish. As a first transformation step, all TBox axioms are transformed into the following universally valid concepts in negation normal form:

$\neg PhDStudent \sqcup \exists has.Diploma \quad \neg Diploma \sqcup \forall has^-.Graduate \quad \neg Diploma \sqcup \neg Graduate$
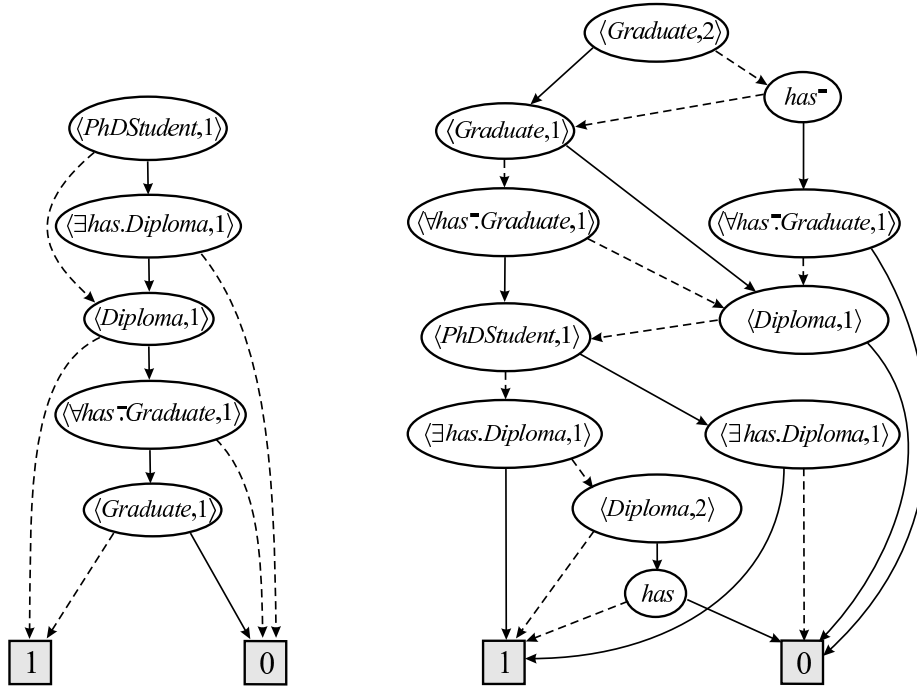
Figure 1: OBDDs arising when processing the terminology of $\mathcal{KB}$; following traditional BDD notation, solid arrows indicate high successors, dashed arrows indicate low successors, and the topmost node is the root

The flattening step can be skipped since all concepts are already flat. Now the relevant concept expressions for describing dominoes are given by the set

$$P(\mathcal{T}) = \{\exists has.Diploma, \forall has^-.Graduate, Diploma, Graduate, PhDStudent\}.$$

We thus obtain the following set Var of Boolean variables (although Var is just a set, our presentation follows the domino intuition):

$\langle\exists has.Diploma, 1\rangle$	has	$\langle\exists has.Diploma, 2\rangle$
$\langle\forall has^-.Graduate, 1\rangle$	has^-	$\langle\forall has^-.Graduate, 2\rangle$
$\langle Diploma, 1\rangle$		$\langle Diploma, 2\rangle$
$\langle Graduate, 1\rangle$		$\langle Graduate, 2\rangle$
$\langle PhDStudent, 1\rangle$		$\langle PhDStudent, 2\rangle$

We are now ready to construct the OBDDs as described. Figure 1 (left) displays an OBDD corresponding to the following Boolean function:

$$\varphi^{\mathbf{kb}} := (\neg[\![\langle PhDStudent, 1\rangle]\!]_\chi \vee [\![\langle\exists has.Diploma, 1\rangle]\!]_\chi)$$
$$\wedge(\neg[\![\langle Diploma, 1\rangle]\!]_\chi \vee [\![\langle\forall has^-.Graduate, 1\rangle]\!]_\chi)$$
$$\wedge(\neg[\![\langle Diploma, 1\rangle]\!]_\chi \vee \neg[\![\langle Graduate, 1\rangle]\!]_\chi)$$

and Fig. 1 (right) shows the OBDD representing the function $[\![\mathcal{T}]\!]_0$ obtained from $\varphi^{\mathbf{kb}}$ by conjunctively adding

$$\varphi^{\mathbf{ex}} \quad = \quad \neg[\![\langle Diploma, 2\rangle]\!]_\chi \vee \neg[\![\langle has\rangle]\!]_\chi \vee [\![\langle \exists has.Diploma, 1\rangle]\!]_\chi \quad \text{and}$$
$$\varphi^{\mathbf{uni}} \quad = \quad \neg[\![\langle \forall has^-.Graduate, 1\rangle]\!]_\chi \vee \neg[\![\langle has^-\rangle]\!]_\chi \vee [\![\langle Graduate, 2\rangle]\!]_\chi.$$

Then, after the first iteration of the algorithm, we arrive at an OBDD representing $[\![\mathcal{T}]\!]_1$ which is displayed in Fig. 2. This OBDD turns out to be the final result $[\![\mathcal{T}]\!]$. The input TBox is derived to be consistent since there is a path from the root node to 1.
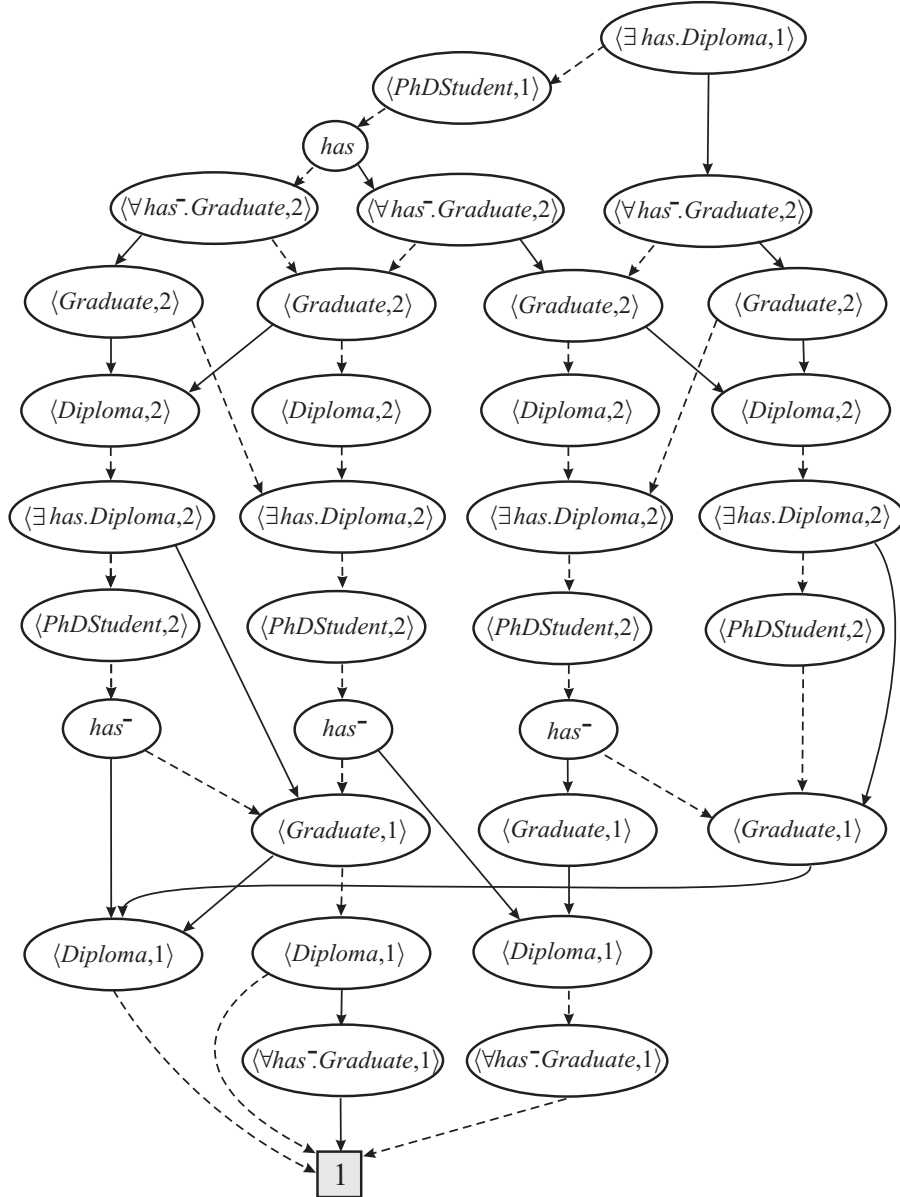

Figure 2: Final OBDD obtained when processing $\mathcal{KB}$, using notation as in Fig. 1; arrows to the 0 node have been omitted for better readability

5. Reasoning with ABox and DL-Safe Rules via Disjunctive Datalog

The above algorithm does not yet take any assertional information about individuals into account, nor does it cover DL-safe rules. The proof of Theorem 6 hinges upon the fact that the constructed domino set $\mathbb{D}_\mathcal{T}$ induces a model of the terminology $\mathcal{T}$, and Lemma 3 states that this is indeed the *greatest* model in a certain sense. This provides some first intuition of the problems arising when ABoxes are to be added to the knowledge base: $\mathcal{ALCI}b$ knowledge bases with ABoxes do generally not have a greatest model.

We thus employ *disjunctive Datalog* (see Eiter et al., 1997) as a paradigm that allows us to incorporate ABoxes into the reasoning process. The basic idea is to forge a Datalog program that – depending on two given individuals a and b – describes possible dominoes that may connect a and b in models of the knowledge base. There might be various, irreconcilable such dominoes in different models, but disjunctive Datalog supports such choice since it admits multiple minimal models. As long as the knowledge base has some model, there is at least one possible domino for every pair of individuals (possibly without connecting roles) – only if this is not the case, the Datalog program will infer a contradiction. Another reason for choosing disjunctive Datalog is that it allows for the straightforward incorporation of DL-safe rules.

We use the OBDD computed from the terminology as a kind of pre-compiled version of the relevant terminological information. ABox information is then considered as an incomplete specification of dominoes that must be accepted by the OBDD, and the Datalog program simulates the OBDD's evaluation for each of those.

Definition 7. *Consider an extended $\mathcal{ALCI}b$ knowledge base* KB $= \langle \mathcal{T}, \mathcal{P} \rangle$, *and an OBDD* $\mathbb{O} = \langle N, n_{\text{root}}, n_{\text{true}}, n_{\text{false}}, \text{low}, \text{high}, \text{Var}, \lambda \rangle$ *that represents the function* $[\![\mathcal{T}]\!]$ *as defined by Algorithm 2. A disjunctive Datalog program* DD(KB) *is defined as follows.* DD(KB) *uses the following predicates:*

- *a unary predicate S_C for every concept expression $C \in P(\text{FLAT}(\mathcal{T}))$,*
- *a binary predicate S_R for every atomic role $R \in \mathsf{N}_R$,*
- *a binary predicate A_n for every OBDD node $n \in N$,*
- *the equality predicate $\approx$.*

The constants in DD(KB) *are the individual names used in* $\mathcal{P}$. *The disjunctive Datalog rules of* DD(KB) *are defined as follows:*[8]

(1) For every DL-safe rule $B \to H$ from RB, DD(KB) *contains the rule obtained from $B \to H$ by replacing all $C(x)$ by $S_C(x)$ and all $R(x, y)$ by $S_R(x, y)$.*

(2) DD(KB) *contains rules $\to A_{n_{\text{root}}}(x, y)$ and $A_{n_{\text{false}}}(x, y) \to$.*

(3) If $n \in N$ with $\lambda(n) = \langle C, 1 \rangle$ then DD(KB) *contains rules*
$S_C(x) \land A_n(x, y) \to A_{\text{high}(n)}(x, y)$ *and* $A_n(x, y) \to A_{\text{low}(n)}(x, y) \lor S_C(x)$.

(4) If $n \in N$ with $\lambda(n) = \langle C, 2 \rangle$ then DD(KB) *contains rules*
$S_C(y) \land A_n(x, y) \to A_{\text{high}(n)}(x, y)$ *and* $A_n(x, y) \to A_{\text{low}(n)}(x, y) \lor S_C(y)$.

(5) If $n \in N$ with $\lambda(n) = R$ for some $R \in \mathsf{N}_R$ then DD(KB) *contains rules*
$S_R(x, y) \land A_n(x, y) \to A_{\text{high}(n)}(x, y)$ *and* $A_n(x, y) \to A_{\text{low}(n)}(x, y) \lor S_R(x, y)$.

(6) If $n \in N$ with $\lambda(n) = R^-$ for some $R \in \mathsf{N}_R$ then DD(KB) *contains rules*
$S_R(y, x) \land A_n(x, y) \to A_{\text{high}(n)}(x, y)$ *and* $A_n(x, y) \to A_{\text{low}(n)}(x, y) \lor S_R(y, x)$.

Note that the arity of predicates in DD(KB) is bounded by 2. Hence, the number of ground atoms is quadratic with respect to the number of constants (individual names), whence the worst-case complexity for satisfiability checking is NP w.r.t. the number of individuals (and especially w.r.t.

---

[8]Note that we use disjunctive Datalog with equality. However, every disjunctive Datalog program with equality can be reduced to one without equality in linear time, as equality can be axiomatized (see, e.g., Fitting, 1996).

the number of facts), as opposed to the NExpTime complexity of disjunctive Datalog in general (Dantsin et al., 2001). Note that, of course, DD(KB) may still be exponential in the size of KB in the worst case: DD(KB) is linear in the size of the underlying OBDD which in turn may have exponential size compared to the set of propositional variables used in the represented Boolean functions. Finally the number of these variables is linearly bounded by the size of KB. It remains to show the correctness of the Datalog translation.

Lemma 8. *Given an extended $\mathcal{ALCI}b$ knowledge base* KB *such that $\mathcal{I}$ is a model of* KB*, there is a model $\mathcal{J}$ of* DD(KB) *such that*

- *$\mathcal{I} \models C(a)$ iff $\mathcal{J} \models S_C(a)$,*
- *$\mathcal{I} \models R(a, b)$ iff $\mathcal{J} \models S_R(a, b)$, and*
- *$\mathcal{I} \models a \approx b$ iff $\mathcal{J} \models a \approx b$.*

for any $a, b \in \mathsf{N}_I$, $C \in \mathsf{N}_C$, and $R \in \mathsf{N}_R$.

Proof Let KB $= \langle \mathcal{T}, \mathcal{P} \rangle$. We define an interpretation $\mathcal{J}$ of DD(KB). The domain of $\mathcal{J}$ contains the named individuals from $\mathcal{I}$, i.e., $\Delta^{\mathcal{J}} = \{a^{\mathcal{I}} \mid a \in \mathsf{N}_I\}$. For individuals a, we set $a^{\mathcal{J}} := a^{\mathcal{I}}$. The interpretation of predicate symbols is now defined as follows (note that $A_n^{\mathcal{J}}$ is defined inductively on the path length from n_{root} to n):

- $\delta \in S_C^{\mathcal{J}}$ iff $\delta \in C^{\mathcal{I}}$
- $\langle \delta_1, \delta_2 \rangle \in S_R^{\mathcal{J}}$ iff $\langle \delta_1, \delta_2 \rangle \in R^{\mathcal{I}}$
- $\langle \delta_1, \delta_2 \rangle \in A_{n_{\mathsf{root}}}^{\mathcal{J}}$ for all $\delta_1, \delta_2 \in \Delta^{\mathcal{J}}$
- $\langle \delta_1, \delta_2 \rangle \in A_n^{\mathcal{J}}$ for $n \neq n_{\mathsf{root}}$ if there is a node n' such that $\langle \delta_1, \delta_2 \rangle \in A_{n'}^{\mathcal{J}}$, and one of the following is the case:
  - $\lambda(n') = \langle C, i \rangle$, for some $i \in \{1, 2\}$, and $n = \mathsf{low}(n')$ and $\delta_i \notin C^{\mathcal{I}}$
  - $\lambda(n') = \langle C, i \rangle$, for some $i \in \{1, 2\}$, and $n = \mathsf{high}(n')$ and $\delta_i \in C^{\mathcal{I}}$
  - $\lambda(n') = R$ and $n = \mathsf{low}(n')$ and $\langle \delta_1, \delta_2 \rangle \notin R^{\mathcal{I}}$
  - $\lambda(n') = R$ and $n = \mathsf{high}(n')$ and $\langle \delta_1, \delta_2 \rangle \in R^{\mathcal{I}}$

Mark that, in the last two items, R is any role expression from Var, i.e., a role name or its inverse. Also note that due to the acyclicity of $\mathbb{O}$, the interpretation of the A-predicates is indeed well-defined. We now show that $\mathcal{J}$ is a model of DD(KB). To this end, first note that the extensions of predicates S_C and S_R in $\mathcal{J}$ were defined to coincide with the extensions of C and R on the named individuals of $\mathcal{I}$. Since $\mathcal{I}$ satisfies $\mathcal{P}$, all rules introduced in item (1) of Definition 7 are satisfied by $\mathcal{J}$. The restriction of DL-safe rules to named individuals can be discarded here since $\Delta^{\mathcal{J}}$ contains only named individuals from $\Delta^{\mathcal{I}}$.

Similarly, we find that the rules of cases (3)–(6) are satisfied by $\mathcal{J}$. Consider the first rule of (3), $S_C(x) \wedge A_n(x, y) \to A_{\mathsf{high}(n)}(x, y)$, and assume that $\delta_1 \in S_C^{\mathcal{J}}$ and $\langle \delta_1, \delta_2 \rangle \in A_n^{\mathcal{J}}$. Thus $\delta_1 \in C^{\mathcal{I}}$. Using the preconditions of (3) and the definition of $\mathcal{J}$, we conclude that $\langle \delta_1, \delta_2 \rangle \in A_{\mathsf{high}(n)}^{\mathcal{J}}$. The second rule of case (3) covers the analogous negative case. All other cases can be treated similarly.

Finally, for case (2), we need to show that $A_{n_{\mathsf{false}}}^{\mathcal{J}} = \emptyset$. For that, we first explicate the correspondence between domain elements of $\mathcal{I}$ and sets of variables of $\mathbb{O}$. Given elements $\delta_1, \delta_2 \in \Delta^{\mathcal{I}}$ we define $V_{\delta_1, \delta_2} := \{\langle C, n \rangle \mid C \in P(\mathsf{FLAT}(\mathcal{T})), \delta_n \in C^{\mathcal{I}}\} \cup \{R \mid \langle \delta_1, \delta_2 \rangle \in R^{\mathcal{I}}\}$, the set of variables corresponding to the $\mathcal{I}$-domino between δ_1 and δ_2.

Now $A_{n_{\text{false}}}^{\mathcal{J}} = \emptyset$ clearly is a consequence of the following claim: for all $\delta_1, \delta_2 \in \Delta^{\mathcal{I}}$ and all $n \in N$, we find that $\langle \delta_1, \delta_2 \rangle \in A_n$ implies $\varphi_n(V_{\delta_1,\delta_2}) = \textit{true}$ (using the notation of Definition 6). The proof proceeds by induction. For the case $n = n_{\text{root}}$, we find that $\varphi_{n_{\text{root}}} = \llbracket \mathcal{T} \rrbracket$. Since V_{δ_1,δ_2} represents a domino of $\mathcal{I}$, the claim thus follows by combining Proposition 7 and Lemma 3.

For the induction step, let n be a node such that $\langle \delta_1, \delta_2 \rangle \in A_n$ follows from the inductive definition of $\mathcal{J}$ based on some predecessor node n' for which the claim has already been established. Note that n' may not be unique. The cases in the definition of $\mathcal{J}$ must be considered individually. Thus assume n', n, and δ_1 satisfy the first case, and that $\langle \delta_1, \delta_2 \rangle \in A_n$. By induction hypothesis, $\varphi_{n'}(V_{\delta_1,\delta_2}) = \textit{true}$, and by Definition 6 the given case yields $\varphi_n(V_{\delta_1,\delta_2}) = \textit{true}$ as well. The other cases are similar. $\qquad\square$

Lemma 9. *Given an $\mathcal{ALCIb}$ knowledge base* KB *such that $\mathcal{J}$ is a model of* DD(KB)*, there is a model $\mathcal{I}$ of* KB *such that*

- $\mathcal{I} \models C(a)$ *iff* $\mathcal{J} \models S_C(a)$,
- $\mathcal{I} \models R(a,b)$ *iff* $\mathcal{J} \models S_R(a,b)$*, and*
- $\mathcal{I} \models a \approx b$ *iff* $\mathcal{J} \models a \approx b$,

for any $a, b \in \mathsf{N}_I$, $C \in \mathsf{N}_C$, and $R \in \mathsf{N}_R$.

Proof Let KB $= \langle \mathcal{T}, \mathcal{P} \rangle$. We construct an interpretation $\mathcal{I}$ whose domain $\Delta^{\mathcal{I}}$ consists of all sequences starting with an individual name followed by a (possibly empty) sequence of dominoes from $\mathbb{D}_{\mathcal{J}}$ such that, for every $\sigma \in \Delta^{\mathcal{I}}$,

- if σ begins with $a\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$, then $\{C \mid C \in P(\mathsf{FLAT}(\mathcal{T})), a^{\mathcal{J}} \in S_C^{\mathcal{J}}\} = \mathcal{A}$, and
- if σ contains subsequent letters $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ and $\langle \mathcal{A}', \mathcal{R}', \mathcal{B}' \rangle$, then $\mathcal{B} = \mathcal{A}'$.

For a sequence $\sigma = a\langle \mathcal{A}_1, \mathcal{R}_1, \mathcal{A}_2 \rangle \langle \mathcal{A}_2, \mathcal{R}_2, \mathcal{A}_3 \rangle \ldots \langle \mathcal{A}_{i-1}, \mathcal{R}_{i-1}, \mathcal{A}_i \rangle$, we define $\mathsf{tail}(\sigma) := \mathcal{A}_i$, whereas for a $\sigma = a$ we define $\mathsf{tail}(\sigma) := \{C \mid C \in P(\mathsf{FLAT}(\mathcal{T})), a^{\mathcal{J}} \in S_C^{\mathcal{J}}\}$. Now the mappings of $\mathcal{I}$ are defined as follows:

- for $a \in \mathsf{N}_I$, we have $a^{\mathcal{I}} := a$,
- for $A \in \mathsf{N}_C$, we have $\sigma \in A^{\mathcal{I}}$ iff $A \in \mathsf{tail}(\sigma)$,
- for $R \in \mathsf{N}_R$, we have $\langle \sigma_1, \sigma_2 \rangle \in R^{\mathcal{I}}$ if one of the following holds
  - $\sigma_1 = a \in \mathsf{N}_I$ and $\sigma_2 = b \in \mathsf{N}_I$ and $\langle a, b \rangle \in S_R^{\mathcal{J}}$, or
  - $\sigma_2 = \sigma_1 \langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ with $R \in \mathcal{R}$, or
  - $\sigma_1 = \sigma_2 \langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle$ with $\mathsf{Inv}(R) \in \mathcal{R}$.

Thus, intuitively, $\mathcal{I}$ is constructed by extracting the named individuals as well their concept (and mutual role) memberships from $\mathcal{J}$, and appending an appropriate domino-constructed tree model to each of those named individuals. We proceed by showing that $\mathcal{I}$ is indeed a model of KB.

First note that the definition of $\mathcal{I}$ ensures that, for all individual names $a, b \in \mathsf{N}_I$, we indeed have $\mathcal{I} \models C(a)$ iff $\mathcal{J} \models S_C(a)$, $\mathcal{I} \models R(a,b)$ iff $\mathcal{J} \models S_R(a,b)$, and $\mathcal{I} \models a \approx b$ iff $\mathcal{J} \models a \approx b$. Therefore, the validity of the rules introduced via case (1) ensures that $\mathcal{I}$ is a model of $\mathcal{P}$.

For showing that the TBox is also satisfied, we begin with the following auxiliary observation: for every two individual names $a, b \in \mathsf{N}_I$, and $\mathcal{R}_{ab} := \{R \mid \langle a^{\mathcal{J}}, b^{\mathcal{J}} \rangle \in S_R^{\mathcal{J}}\} \cup \{\mathsf{Inv}(R) \mid \langle b^{\mathcal{J}}, a^{\mathcal{J}} \rangle \in S_R^{\mathcal{J}}\}$, the domino $\langle \mathsf{tail}(a), \mathcal{R}_{ab}, \mathsf{tail}(b) \rangle$ is contained in $\mathbb{D}_{\mathcal{J}}$ (Claim †). Using Proposition 7, it suffices to show that the Boolean function $\llbracket \mathcal{T} \rrbracket$ if applied to $V_{a,b} := \{\mathsf{tail}(a) \times \{1\} \cup \mathcal{R}_{ab} \cup \mathsf{tail}(b) \times \{2\}\}$ yields *true*. Since $\llbracket \mathcal{T} \rrbracket = \varphi_{n_{\text{root}}}$, this is obtained by showing the following: for any $a, b \in \mathsf{N}_I$, we find that $\langle a^{\mathcal{J}}, b^{\mathcal{J}} \rangle \in A_n^{\mathcal{J}}$ implies $\varphi_n(V_{a,b}) = \textit{true}$. Indeed, (†) follows since we have $\langle a^{\mathcal{J}}, b^{\mathcal{J}} \rangle \in A_{n_{\text{root}}}^{\mathcal{J}}$ due to the first rule of (2) in Definition 7. We proceed by induction, starting at the leafs of the OBDD. The case $\langle a, b \rangle \in A_{n_{\text{true}}}^{\mathcal{I}}$ is immediate, and $\langle a, b \rangle \in A_{n_{\text{false}}}^{\mathcal{I}}$ is excluded by the second rule of (2). For the induction step, consider nodes $n, n' \in N$ such that either $\lambda(n) \in V_{a,b}$ and $n' = \mathsf{high}(n)$, or $\lambda(n) \notin V_{a,b}$ and $n' = \mathsf{low}(n)$. We assume that $\langle a^{\mathcal{J}}, b^{\mathcal{J}} \rangle \in A_n^{\mathcal{J}}$, and, by induction, that the claim holds

for n'. If $\lambda_n = \langle C, 1 \rangle$, then one of the rules of case (3) applies to $a^{\mathcal{J}}$ and $b^{\mathcal{J}}$. In both cases, we can infer $\langle a^{\mathcal{J}}, b^{\mathcal{J}} \rangle \in A_{n'}^{\mathcal{J}}$, and hence $\varphi_{n'}(V_{a,b}) = \textit{true}$. Together with the assumptions for this case, Definition 6 implies $\varphi_n(V_{a,b}) = \textit{true}$, as required. The other cases are analogous. This shows (†).

Now we can proceed to show that all individuals of $\mathcal{I}$ are contained in the extension of each concept expression of $\mathsf{FLAT}(\mathcal{T})$. To this end, we first show that $\sigma \in C^{\mathcal{I}}$ iff $C \in \text{tail}(\sigma)$ for all $C \in P(\mathsf{FLAT}(\mathcal{T}))$. If $C \in \mathsf{N}_C$ is atomic, this follows directly from the definition of $\mathcal{I}$. The remaining cases that may occur in $P(\mathsf{FLAT}(\mathcal{T}))$ are $C = \exists U.A$ and $C = \forall U.A$.

First consider the case $C = \exists U.A$ and assume that $\sigma \in C^{\mathcal{I}}$. Thus there is $\sigma' \in \Delta^{\mathcal{I}}$ with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \in A^{\mathcal{I}}$. The construction of the domino model admits three possible cases:

- $\sigma, \sigma' \in \mathsf{N}_I$ and $\mathcal{R}_{\sigma\sigma'} \vdash U$ and $A \in \text{tail}(\sigma')$. Now by (†), the domino $\langle \text{tail}(\sigma), \mathcal{R}_{\sigma\sigma'}, \text{tail}(\sigma') \rangle$ satisfies condition **ex** of Algorithm 1, and thus $C \in \text{tail}(\sigma)$ as required.
- $\sigma' = \sigma\langle \text{tail}(\sigma), \mathcal{R}, \text{tail}(\sigma') \rangle$ with $\mathcal{R} \vdash U$ and $A \in \text{tail}(\sigma')$. Since $\mathbb{D}_{\mathcal{T}} \subseteq \mathbb{D}_0$, we find that $\langle \text{tail}(\sigma), \mathcal{R}, \text{tail}(\sigma') \rangle$ satisfies condition **ex**, and thus $C \in \text{tail}(\sigma)$ as required.
- $\sigma = \sigma'\langle \text{tail}(\sigma'), \mathcal{R}, \text{tail}(\sigma) \rangle$ with $\text{Inv}(\mathcal{R}) \vdash U$ and $A \in \text{tail}(\sigma')$. By condition **sym**, $\mathbb{D}_{\mathcal{T}}$ contains the domino $\langle \text{tail}(\sigma), \text{Inv}(\mathcal{R}), \text{tail}(\sigma') \rangle$, and again we use **ex** to conclude $C \in \text{tail}(\sigma)$.

For the converse, assume that $\exists U.A \in \text{tail}(\sigma)$. So $\mathbb{D}_{\mathcal{T}}$ contains a domino $\langle \mathcal{A}, \mathcal{R}, \text{tail}(\sigma) \rangle$. This is obvious if the sequence σ ends with a domino. If $\sigma = a \in \mathsf{N}_I$, then it follows by applying (†) to a with the first individual being arbitrary. By **sym** $\mathbb{D}_{\mathcal{T}}$ also contains the domino $\langle \text{tail}(\sigma), \mathcal{R}, \mathcal{A} \rangle$. By condition **delex**, the latter implies that $\mathbb{D}_{\mathcal{T}}$ contains a domino $\langle \text{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$ such that $\mathcal{R}' \vdash U$ and $A \in \mathcal{A}'$. Thus $\sigma' = \sigma\langle \text{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$ is an $\mathcal{I}$-individual such that $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \in A^{\mathcal{I}}$, and we obtain $\sigma \in (\exists U.A)^{\mathcal{I}}$ as claimed.

For the second case, consider $C = \forall U.A$ and assume that $\sigma \in C^{\mathcal{I}}$. As above, we find that $\mathbb{D}_{\mathcal{T}}$ contains some domino $\langle \mathcal{A}, \mathcal{R}, \text{tail}(\sigma) \rangle$, where (†) is needed if $\sigma \in \mathsf{N}_I$. By **sym** we find a domino $\langle \text{tail}(\sigma), \mathcal{R}, \mathcal{A} \rangle$. For a contradiction, suppose that $\forall U.A \notin \text{tail}(\sigma)$. By condition **deluni**, the latter implies that $\mathbb{D}_{\mathcal{T}}$ contains a domino $\langle \text{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$ such that $\mathcal{R}' \vdash U$ and $A \notin \mathcal{A}'$. Thus $\sigma' = \sigma\langle \text{tail}(\sigma), \mathcal{R}', \mathcal{A}' \rangle$ is an $\mathcal{I}$-individual such that $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \notin A^{\mathcal{I}}$. But then $\sigma \notin (\forall U.A)^{\mathcal{I}}$, which is the required contradiction.

For the other direction, assume that $\forall U.A \in \text{tail}(\sigma)$. According to the construction of $\mathcal{I}$, for all elements σ' with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$, there are three possible cases:

- $\sigma, \sigma' \in \mathsf{N}_I$ and $\mathcal{R}_{\sigma\sigma'} \vdash U$. Now by (†), the domino $\langle \text{tail}(\sigma), \mathcal{R}_{\sigma\sigma'}, \text{tail}(\sigma') \rangle$ satisfies condition **uni**, whence $A \in \text{tail}(\sigma')$.
- $\sigma' = \sigma\langle \text{tail}(\sigma), \mathcal{R}, \text{tail}(\sigma') \rangle$ with $\mathcal{R} \vdash U$. Since $\mathbb{D}_{\mathcal{T}} \subseteq \mathbb{D}_0$, $\langle \text{tail}(\sigma), \mathcal{R}, \text{tail}(\sigma') \rangle$ must satisfy condition **uni**, and thus $A \in \text{tail}(\sigma')$.
- $\sigma = \sigma'\langle \text{tail}(\sigma'), \mathcal{R}, \text{tail}(\sigma) \rangle$ with $\text{Inv}(\mathcal{R}) \vdash U$. By condition **sym**, $\mathbb{D}_{\mathcal{T}}$ also contains the domino $\langle \text{tail}(\sigma), \text{Inv}(\mathcal{R}), \text{tail}(\sigma') \rangle$, and we can again use **uni** to conclude $A \in \text{tail}(\sigma')$.

Thus, $A \in \text{tail}(\sigma')$ for all U-successors σ' of σ, and hence $\sigma \in (\forall U.A)^{\mathcal{I}}$ as claimed.

To finish the proof, note that any domino $\langle \mathcal{A}, \mathcal{R}, \mathcal{B} \rangle \in \mathbb{D}_{\mathcal{T}}$ satisfies condition **kb**. Using **sym**, we have that for any $\sigma \in \Delta^{\mathcal{I}}$, the axiom $\sqcap_{D \in \text{tail}(\sigma)} D \sqsubseteq C$ is a tautology for all $C \in \mathsf{FLAT}(\mathcal{T})$. As shown above, $\sigma \in D^{\mathcal{I}}$ for all $D \in \text{tail}(\sigma)$, and thus $\sigma \in C^{\mathcal{I}}$. Hence every individual of $\mathcal{I}$ is an instance of each concept of $\mathsf{FLAT}(\mathcal{T})$ as required. $\qquad\square$

Lemmas 8 and 9 give rise to the following theorem which finishes the technical development of this section by showing that $\mathsf{DD}(\mathsf{KB})$ faithfully captures both positive and negative ground conclusions of KB, and in particular that $\mathsf{DD}(\mathsf{KB})$ and KB are equisatisfiable.

Theorem 10. *For every extended $\mathcal{ALCI}b$ knowledge base* KB *hold*

- KB *and* DD(KB) *are equisatisfiable,*
- KB $\models C(a)$ *iff* DD(KB) $\models S_C(a)$,
- KB $\models R(a, b)$ *iff* DD(KB) $\models S_R(a, b)$, *and*
- KB $\models a \approx b$ *iff* DD(KB) $\models a \approx b$,

for any $a, b \in N_I$, $C \in N_C$, and $R \in N_R$.

Proof Immediate from Lemma 8 and Lemma 9. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Coming back to our example knowledge base $\mathcal{KB}$ from Section 4, the corresponding disjunctive Datalog program DD($\mathcal{KB}$) contains 70 rules: two rules for each of the 33 labeled nodes from the OBDD displayed in Fig. 2, the two rules $\to A_{n_{\text{root}}}(x, y)$ and $A_{n_{\text{false}}}(x, y) \to$ as well as the two rules $\to S_{Diploma}(laureus)$ and $\to S_{PhDStudent}(laureus)$ introduced by conceiving the two ABox statements as DL-safe rules and translating them accordingly. The program turns out to be unsatisfiable, witnessed by the unsatisfiable subprogram displayed in Fig. 3.

$$
\begin{array}{rcl}
& \to & S_{Diploma}(laureus) \\
& \to & A_0(x, y) \\
A_0(x, y) \wedge S_{\exists has.Diploma}(x) & \to & A_5(x, y) \\
A_1(x, y) \wedge S_{PhDStudent}(x) & \to & A_{\mathsf{false}}(x, y) \\
A_5(x, y) \wedge S_{\forall has^-.Graduate}(y) & \to & A_9(x, y) \\
A_8(x, y) \wedge S_{Graduate}(y) & \to & A_{13}(x, y) \\
A_9(x, y) \wedge S_{Graduate}(y) & \to & A_{13}(x, y) \\
A_{12}(x, y) \wedge S_{Diploma}(y) & \to & A_{\mathsf{false}}(x, y) \\
A_{13}(x, y) \wedge S_{Diploma}(y) & \to & A_{\mathsf{false}}(x, y) \\
A_{16}(x, y) \wedge S_{\exists has.Diploma}(y) & \to & A_{\mathsf{false}}(x, y) \\
A_{20}(x, y) \wedge S_{PhDStudent}(y) & \to & A_{\mathsf{false}}(x, y) \\
A_{\mathsf{false}}(x, y) & \to &
\end{array}
\qquad
\begin{array}{rcl}
& \to & S_{PhDStudent}(laureus) \\
& & \\
A_0(x, y) & \to & A_1(x, y) \vee S_{\exists has.Diploma}(x) \\
& & \\
A_5(x, y) & \to & A_8(x, y) \vee S_{\forall has^-.Graduate}(y) \\
A_8(x, y) & \to & A_{12}(x, y) \vee S_{Graduate}(y) \\
A_9(x, y) & \to & A_{16}(x, y) \vee S_{Graduate}(y) \\
& & \\
& & \\
A_{16}(x, y) & \to & A_{20}(x, y) \vee S_{\exists has.Diploma}(y) \\
\end{array}
$$

Figure 3: Unsatisfiable subprogram of DD($\mathcal{KB}$) witnessing unsatisfiability of $\mathcal{KB}$

6. Polynomial Transformation from $\mathcal{SHIQ}b_s$ to $\mathcal{ALCI}b$

In this section, we present a stepwise satisfiability-preserving transformation from the description logic $\mathcal{SHIQ}b_s$ to the more restricted $\mathcal{ALCI}b$. This transformation is necessary as our type-elimination method applies directly only to the latter.

6.1. **Unravelings.** For our further considerations, we will use a well-known model transformation technique which will come handy for showing equisatisfiability of knowledge base transformations introduced later on (for an introductory account on unravelings in a DL setting cf., e.g., Rudolph (2011)). Essentially, the transformation takes an arbitrary model of a $\mathcal{SHIQ}b_s$ knowledge base and converts it into a model that is "tree-like". We start with some preliminary definitions. The first one exploits that role subsumption on non-simple roles can be decided by an easy syntactic check that takes only role hierarchy axioms into account.

Definition 8. *Based on a fixed $\mathcal{SHIQ}b_s$ knowledge base* KB, *we define $\sqsubseteq^*$ as the smallest binary relation on the non-simple atomic roles $\mathbf{R}_n$ such that:*

- $R \sqsubseteq^* R$ *for every atomic role R,*
- $R \sqsubseteq^* S$ *and* $\mathrm{Inv}(R) \sqsubseteq^* \mathrm{Inv}(S)$ *for every RBox axiom $R \sqsubseteq S$, and*
- $R \sqsubseteq^* T$ *whenever $R \sqsubseteq^* S$ and $S \sqsubseteq^* T$ for some atomic role S.*

Furthermore, we write $R \sqsubset^* S$ whenever $R \sqsubseteq^* S$ and $S \not\sqsubseteq^* R$.

The next definition introduces a standard model transformation technique that is often used to show variants of the tree model property of a logic. We adopt the definition of Glimm et al. (2007).

Definition 9. *Let* KB *be a consistent extended $\mathcal{SHIQ}b_s$ knowledge base, and let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ be a model for* KB.
The unraveling *of $\mathcal{I}$ is an interpretation that is obtained from $\mathcal{I}$ as follows. We define the set $S \subseteq (\Delta^{\mathcal{I}})^*$ of* sequences *to be the smallest set such that*

- *for every $a \in \mathsf{N}_I$, $a^{\mathcal{I}}$ is a sequence;*
- $\delta_1 \cdots \delta_n \cdot \delta_{n+1}$ *is a sequence, if*
    - $\delta_1 \cdots \delta_n$ *is a sequence,*
    - $\delta_{i+1} \neq \delta_{i-1}$ *for all $i = 2, \ldots, n$,*
    - $\langle \delta_n, \delta_{n+1} \rangle \in R^{\mathcal{I}}$ *for some $R \in \mathsf{N}_R$.*

For each $\sigma = \delta_1 \cdots \delta_n \in S$, set $\mathsf{last}(\sigma) := \delta_n$. Now, we define the unraveling of $\mathcal{I}$ as the interpretation $\mathcal{J} = \langle \Delta^{\mathcal{J}}, \cdot^{\mathcal{J}} \rangle$ with $\Delta^{\mathcal{J}} = S$ and we define the interpretation of concept and role names as follows (where $\sigma, \sigma' \in \Delta^{\mathcal{J}}$ are arbitrary sequences in $\Delta^{\mathcal{J}}$):

- (a) *for each $a \in \mathsf{N}_I$, set $a^{\mathcal{J}} := a^{\mathcal{I}}$;*
- (b) *for each concept name $A \in \mathsf{N}_C$, set $\sigma \in A^{\mathcal{J}}$ iff $\mathsf{last}(\sigma) \in A^{\mathcal{I}}$;*
- (c) *for each role name $R \in \mathsf{N}_R$, set $\langle \sigma, \sigma' \rangle \in R^{\mathcal{J}}$ iff*
    - $\sigma' = \sigma\delta$ *for some $\delta \in \Delta^{\mathcal{I}}$ and $\langle \mathsf{last}(\sigma), \mathsf{last}(\sigma') \rangle \in R^{\mathcal{I}}$ or*
    - $\sigma = \sigma'\delta$ *for some $\delta \in \Delta^{\mathcal{I}}$ and $\langle \mathsf{last}(\sigma), \mathsf{last}(\sigma') \rangle \in R^{\mathcal{I}}$ or*
    - $\sigma = a^{\mathcal{I}}$, $\sigma' = b^{\mathcal{I}}$ *for some $a, b \in \mathsf{N}_I$ and $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$.*

Unraveling a model of an extended $\mathcal{SHIQ}b_s$ knowledge base results in an interpretation that still satisfies most of the knowledge base's axioms, except for transitivity axioms. The following definition provides a "repair strategy" for unravelings such that also the transitivity conditions are again satisfied. The presented definition is inspired by a similar one by Motik (2006).

Definition 10. *Given an interpretation $\mathcal{I}$ and a knowledge base* KB, *we define the* completion *of $\mathcal{I}$ with respect to* KB *as the new interpretation $\mathcal{J} = \langle \Delta^{\mathcal{J}}, \cdot^{\mathcal{J}} \rangle$ as follows:*

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}}$,
- $a^{\mathcal{J}} := a^{\mathcal{I}}$ *for every $a \in \mathsf{N}_I$,*
- $A^{\mathcal{J}} := A^{\mathcal{I}}$ *for every $A \in \mathsf{N}_C$,*
- *for all simple roles R, we set $R^{\mathcal{J}} := R^{\mathcal{I}}$,*
- *for all non-simple roles R, $R^{\mathcal{J}}$ is set to the transitive closure of $R^{\mathcal{I}}$ if $\mathsf{Tra}(R) \in$ KB, otherwise $R^{\mathcal{J}} := R^{\mathcal{I}} \cup \bigcup_{S \sqsubset^* R \text{ with } \mathsf{Tra}(S) \in \mathrm{KB} \text{ or } \mathsf{Tra}(\mathrm{Inv}(S)) \in \mathrm{KB}} (S^{\mathcal{I}})^*$, where $(S^{\mathcal{I}})^*$ denotes the transitive closure of $S^{\mathcal{I}}$.*

Having the above tools at hand, we are now ready to show that unraveling and subsequently completing a model of an extended knowledge base will result in a model. This correspondence will be helpful for showing the completeness of the knowledge base transformation steps introduced below.

Lemma 11. *Let* KB *be an extended* $\mathcal{SHIQb}_s$ *knowledge base and let* $\mathcal{I}$ *be a model of* KB. *Moreover, let* $\mathcal{J}$ *be the unraveling of* $\mathcal{I}$ *and let* $\mathcal{K}$ *be the completion of* $\mathcal{J}$. *Then the following hold:*

(1) $\mathcal{J}$ satisfies all axioms of KB *that are not transitivity axioms.*

(2) For all sequences $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}, \sigma_n$ with $n > 3$ and $\langle \sigma_i, \sigma_{i+1} \rangle \in R^{\mathcal{J}}$ for $1 \le i \le n$, and where $\sigma_1, \sigma_n \in \{a^{\mathcal{J}} \mid a \in \mathsf{N}_I\}$ and $\sigma_2, \ldots, \sigma_{n-1} \notin \{a^{\mathcal{J}} \mid a \in \mathsf{N}_I\}$, we have $\sigma_1 = \sigma_n$ and $\sigma_2 = \sigma_{n-1}$.

(3) $\mathcal{K}$ is a model of KB.

Proof For the first claim, we investigate all the possible axiom types. First, as $\mathcal{I}$ and $\mathcal{J}$ coincide w.r.t. concept and role memberships of all named individuals (i.e., individuals σ for which $\sigma = a^{\mathcal{I}}$ for some $a \in \mathsf{N}_I$), they satisfy the same DL-safe rules.

For role hierarchy axioms $U \sqsubseteq V$ with U, V restricted, suppose for a contradiction that $\mathcal{J}$ does not satisfy $U \sqsubseteq V$, i.e., that there are two elements $\sigma, \sigma' \in \Delta^{\mathcal{J}}$ such that $\langle \sigma, \sigma' \rangle \in U^{\mathcal{J}}$ but $\langle \sigma, \sigma' \rangle \notin V^{\mathcal{J}}$. As U is restricted, either both σ and σ' are named individuals or $\sigma' = \sigma\delta$ or $\sigma = \sigma'\delta$. Therefore we know that $\langle \mathsf{last}(\sigma), \mathsf{last}(\sigma') \rangle \in U^{\mathcal{I}}$ but $\langle \mathsf{last}(\sigma), \mathsf{last}(\sigma') \rangle \notin V^{\mathcal{I}}$ which would violate $U \sqsubseteq V$ and hence, gives a contradiction.

Next, we consider TBox axioms (remember that we assume them to be normalized into axioms $\top \sqsubseteq C$ with C in negation normal form). By induction on the role depth, we will show that for every concept D it holds that $\sigma \in D^{\mathcal{J}}$ iff $\mathsf{last}(\sigma) \in D^{\mathcal{I}}$. The satisfaction of $\top \sqsubseteq C$ in $\mathcal{J}$ then directly follows via $\Delta^{\mathcal{J}} = \{\sigma \in \Delta^{\mathcal{J}} \mid \mathsf{last}(\sigma) \in \Delta^{\mathcal{I}}\} = \{\sigma \in \Delta^{\mathcal{J}} \mid \mathsf{last}(\sigma) \in C^{\mathcal{I}}\} = C^{\mathcal{J}}$.

As base case, note that for $D \in \mathsf{N}_C$, the claim follows by definition, while for $D = \top$ and $D = \bot$ the claim trivially holds. For the induction steps, note that (i) the claimed correspondence transfers immediately from concepts to their Boolean combinations and (ii) that for every $\sigma \in \Delta^{\mathcal{J}}$, the function $\mathsf{last}(\cdot)$ gives rise to an isomorphism φ between the neighborhood of σ in $\mathcal{J}$ and the neighborhood of $\mathsf{last}(\sigma)$ in $\mathcal{I}$. More precisely, φ maps $\{\sigma' \in \Delta^{\mathcal{J}} \mid \langle \sigma, \sigma' \rangle \in R^{\mathcal{J}}$ for some $R \in \mathbf{R}\}$ to $\{\delta' \in \Delta^{\mathcal{I}} \mid \langle \mathsf{last}(\sigma), \delta' \rangle \in R^{\mathcal{I}}$ for some $R \in \mathbf{R}\}$ such that $\langle \sigma, \sigma' \rangle \in S^{\mathcal{J}}$ iff $\langle \mathsf{last}(\sigma), \varphi(\sigma') \rangle \in S^{\mathcal{I}}$ for all roles $S \in \mathsf{N}_R$ as well as $\sigma' \in E^{\mathcal{J}}$ iff $\varphi(\sigma') \in E^{\mathcal{I}}$ for concepts E that have a smaller role depth than D (by induction hypothesis). Thereby, the claimed correspondence transfers to existential, universal, and cardinality restrictions as well.

For the second claim, we observe that by the definition of the unraveling, no individual $\sigma = \delta_1 \ldots \delta_k$ can be directly connected by some role to an individual $\sigma' = \delta'_1 \ldots \delta'_l$ with $\delta_1 \ne \delta'_1$ unless $k = l = 1$ in which case both individuals would be named by construction. On the other hand, every role chain starting from some named individual δ and not containing any other named individual contains only individuals of the form δw with $w \in (\Delta^{\mathcal{I}})^*$. Thus, we conclude that $\sigma_1 = \sigma_n$. Now, suppose $\sigma_2 \ne \sigma_{n-1}$. By construction we have $\sigma_2 = \sigma_1 \delta$ and $\sigma_{n-1} = \sigma_n \delta' = \sigma_1 \delta'$ with $\delta \ne \delta'$. However, then by construction, every role path from σ_2 to σ_{n-1} must contain σ_1 which is named and hence contradicts the assumption. Therefore $\sigma_2 = \sigma_{n-1}$.

Considering the third claim, we easily find that all transitivity axioms as well as role hierarchy statements are satisfied by construction. For the TBox axioms, the argumentation is similar to the one used to prove the first claim but it has to be extended by the following observation: By construction, for all new role instances $\langle \sigma, \sigma' \rangle \in R^{\mathcal{K}} \setminus R^{\mathcal{J}}$ introduced by the completion, there is already a σ^* with $\langle \sigma, \sigma^* \rangle \in R^{\mathcal{J}}$ such that $\langle \sigma, \sigma^* \rangle \in S^{\mathcal{J}}$ iff $\langle \sigma, \sigma' \rangle \in S^{\mathcal{I}}$ for all roles $S \in \mathsf{N}_R$ as well as $\sigma^* \in E^{\mathcal{J}}$ iff $\sigma' \in E^{\mathcal{I}}$ for concepts E. Therefore (and since non-simple roles are forbidden in cardinality constraints) the concept extensions do not change in $\mathcal{K}$ compared to $\mathcal{J}$. Finally, the DL-safe rules are valid: Due to the first claim they hold in $\mathcal{J}$. Then, they also hold in $\mathcal{K}$ since, by construction $\mathcal{K}$ and $\mathcal{J}$ coincide when restricted to named individuals. In order to see the latter, note that $\mathcal{J}$ also coincides with $\mathcal{I}$ w.r.t. named individuals and $\mathcal{I}$ satisfies all transitivity axioms, thus the completion does not introduce new role instances, as far as named individuals are concerned. $\quad\square$

6.2. **From** $\mathcal{SHIQb}_s$ **to** $\mathcal{ALCHIQb}$. As observed by Rudolph et al. (2008a), a slight generalization of results by Motik (2006) yields that any $\mathcal{SHIQb}_s$ knowledge base KB can be transformed into an equisatisfiable $\mathcal{ALCHIQb}$ knowledge base. For the case of extended knowledge bases, this transformation has to be adapted in order to correctly treat the entailment of ground facts $R(a, b)$ for non-simple roles R via transitivity. We start by defining this modified transformation, whereby the ground fact entailment is taken care of by appropriate DL-safe rules.

Definition 11. *Let cl*(KB) *denote the smallest set of concept expressions where*

- NNF$(\neg C \sqcup D) \in cl$(KB) *for any TBox axiom $C \sqsubseteq D$,*
- $D \in cl$(KB) *for every subexpression D of some concept $C \in cl$(KB),*
- NNF$(\neg C) \in cl$(KB) *for any $\leqslant n\,R.C \in cl$(KB),*
- $\forall S.C \in cl$(KB) *whenever* Tra$(S) \in$ KB *and $S \sqsubseteq^* R$ for a role R with $\forall R.C \in cl$(KB).*

Finally, let Θ_S(KB) denote the extended knowledge base obtained from KB *by removing all transitivity axioms* Tra(R) *and*

- *adding the axiom $\forall R.C \sqsubseteq \forall R.(\forall R.C)$ to* KB *whenever $\forall R.C \in cl$(KB),*
- *adding the axiom $\exists (R \sqcap R^-).\top \sqsubseteq \text{Self}_R$ to* KB*, where Self_R is a fresh concept,*
- *adding the DL-safe rules $\text{Self}_R(x) \to R(x, x)$ and $R(x, y), R(y, z) \to R(x, z)$ to* KB*.*

Note that the knowledge base translation defined by Θ_S can be done in polynomial time. We now show that the defined transformation works as expected, making use of the model transformation techniques established in the previous section. Parts of the proof are adopted from Motik (2006).

Proposition 12. *Let* KB *be an extended $\mathcal{SHIQb}_s$ knowledge base. Then* KB *and Θ_S(KB) are equisatisfiable.*

Proof Obviously, every model $\mathcal{I}$ of KB is a model of Θ_S(KB) if we additionally stipulate $\text{Self}_R :=$ $\{\delta \mid \langle \delta, \delta \rangle \in R^{\mathcal{I}}\}$.

For the other direction, let $\mathcal{K}$ be a model of Θ_S(KB). Let now $\mathcal{I}$ be the unraveling of $\mathcal{K}$ and let $\mathcal{J}$ be the completion of $\mathcal{I}$ w.r.t. KB. As Θ_S(KB) does not contain any transitivity statements, we know by Lemma 11 (1) that $\mathcal{I}$ is a model of Θ_S(KB) as well.

As a direct consequence of the definition of the completion, note that for all simple roles V we have $V^{\mathcal{J}} = V^{\mathcal{I}}$ (fact †).

We now prove that $\mathcal{J}$ is a model of KB by considering all axioms, starting with the RBox. Every transitivity axiom of KB is obviously satisfied by the definition of $\mathcal{J}$. Moreover, every role inclusion $V \sqsubseteq W$ axiom is also satisfied:

If both V and W are Boolean role expressions (which by definition contain only simple roles) this is a trivial consequence of (†). If V is a Boolean role expression and W is a non-simple role, this follows from (†) and the fact that, by construction of $\mathcal{J}$, we have $R^{\mathcal{I}} \subseteq R^{\mathcal{J}}$ for every non-simple role R. As a remaining case, assume that both V and W are non-simple roles. If W is not transitive, this follows directly from the definition, otherwise we can conclude it from the fact that the transitive closure is a monotone operation w.r.t. set inclusion.

We proceed by examining the concept expressions $C \in cl$(KB) and show via structural induction that $C^{\mathcal{I}} \subseteq C^{\mathcal{J}}$. As base case, for every concept of the form A or $\neg A$ for $A \in \mathsf{N}_C$ this claim follows directly from the definition of $\mathcal{J}$. We proceed with the induction steps for all possible forms of a complex concept C (mark that all $C \in cl$(KB) are in negation normal form):

- Clearly, if $D_1^{\mathcal{I}} \subseteq D_1^{\mathcal{J}}$ and $D_2^{\mathcal{I}} \subseteq D_2^{\mathcal{J}}$ by induction hypothesis, we can directly conclude $(D_1 \sqcap D_2)^{\mathcal{I}} \subseteq (D_1 \sqcap D_2)^{\mathcal{J}}$ as well as $(D_1 \sqcup D_2)^{\mathcal{I}} \subseteq (D_1 \sqcup D_2)^{\mathcal{J}}$.

- Likewise, as we have $V^{\mathcal{I}} \subseteq V^{\mathcal{J}}$ for all simple role expressions and non-simple roles V and again $D^{\mathcal{I}} \subseteq D^{\mathcal{J}}$ due to the induction hypothesis, we can conclude $(\exists V.D)^{\mathcal{I}} \subseteq (\exists V.D)^{\mathcal{J}}$ as well as $(\geqslant n\, V.D)^{\mathcal{I}} \subseteq (\geqslant n\, V.D)^{\mathcal{J}}$.
- Now, consider $C = \forall V.D$. If V is a simple role expression, we know that $V^{\mathcal{J}} = V^{\mathcal{I}}$, whence we can derive $(\forall V.D)^{\mathcal{I}} \subseteq (\forall V.D)^{\mathcal{J}}$ from the induction hypothesis.

    It remains to consider the case $C = \forall R.D$ for non-simple roles R. Assume $\sigma \in (\forall R.D)^{\mathcal{I}}$. If there is no σ' with $\langle \sigma, \sigma' \rangle \in R^{\mathcal{J}}$, then $\sigma \in (\forall R.D)^{\mathcal{J}}$ is trivially true. Now assume there are such σ'. For each of them, we can distinguish two cases:
    - $\langle \sigma, \sigma' \rangle \in R^{\mathcal{I}}$, implying $\sigma' \in D^{\mathcal{I}}$ and, via the induction hypothesis, $\sigma' \in D^{\mathcal{J}}$,
    - $\langle \sigma, \sigma' \rangle \notin R^{\mathcal{I}}$. Yet, by construction of $\mathcal{J}$, this means that there is a role S with $S \sqsubseteq^* R$ and $\mathsf{Tra}(S) \in \mathrm{KB}$ and a sequence $\sigma = \sigma_0, \ldots, \sigma_n = \sigma'$ with $\langle \sigma_k, \sigma_{k+1} \rangle \in S^{\mathcal{I}}$ for all $0 \leq k < n$. Then $\sigma \in (\forall R.D)^{\mathcal{I}}$ implies $\sigma \in (\forall S.D)^{\mathcal{I}}$, and hence $\sigma_1 \in D^{\mathcal{I}}$. By Definition 11, $\Theta_S(\mathrm{KB})$ contains the axiom $\forall S.D \sqsubseteq \forall S.(\forall S.D)$, and hence $\sigma_1 \in (\forall S.D)^{\mathcal{I}}$. Continuing this simple induction, we find that $\sigma_k \in D^{\mathcal{I}}$ for all $k = 1, \ldots, n$ including $\sigma_n = \sigma'$.

    So we can conclude that for all such σ' we have $\sigma' \in D^{\mathcal{I}}$. Via the induction hypothesis follows $\sigma \in D^{\mathcal{J}}$ and hence we can conclude $\sigma \in (\forall R.D)^{\mathcal{J}}$.
- Finally, consider $C = \leqslant n\, R.D$ and assume $\sigma \in (\leqslant n\, R.D)^{\mathcal{I}}$. From the fact that R must be simple follows $R^{\mathcal{J}} = R^{\mathcal{I}}$. Moreover, since both D and $\mathsf{NNF}(\neg D)$ are contained in $cl(\mathrm{KB})$ the induction hypothesis gives $D^{\mathcal{J}} = D^{\mathcal{I}}$. Those two facts together imply $\sigma \in (\leqslant n\, R.D)^{\mathcal{I}}$.

Now considering an arbitrary KB TBox axiom $C \sqsubseteq D$, we find $\mathsf{NNF}(\neg C \sqcup D)^{\mathcal{I}} = \Delta^{\mathcal{I}}$ as $\mathcal{I}$ is a model of KB. Moreover – by the correspondence just shown – we have $\mathsf{NNF}(\neg C \sqcup D)^{\mathcal{I}} \subseteq \mathsf{NNF}(\neg C \sqcup D)^{\mathcal{J}}$ and hence also $\mathsf{NNF}(\neg C \sqcup D)^{\mathcal{J}} = \Delta^{\mathcal{J}}$ making $C \sqsubseteq D$ an axiom satisfied in $\mathcal{J}$.

For showing that all DL-safe rules from KB are satisfied, we will prove that $\mathcal{I}$ and $\mathcal{J}$ coincide on the satisfaction of all ground atoms – satisfaction of KB in $\mathcal{J}$ then follows from satisfaction of KB in $\mathcal{I}$. By construction, this is obviously the case for all atoms of the shape $a \approx b$, $C(a)$ and $R(a, b)$ for $a, b \in \mathsf{N}_I$, $C \in \mathsf{N}_C$ and $R \in \mathsf{N}_R$ simple. Moreover we have that $\mathcal{J} \models R(a, b)$ whenever $\mathcal{I} \models R(a, b)$. To settle the other direction, suppose R non-simple and $\mathcal{J} \models R(a, b)$ but $\mathcal{I} \not\models R(a, b)$. But then, there must be a role $S \sqsubseteq^* R$ that is declared transitive and satisfies $\mathcal{J} \models S(a, b)$ but $\mathcal{I} \not\models S(a, b)$. Let us assume that S is a minimal such role w.r.t. $\sqsubseteq^*$. Then, by construction, there must be a sequence $a^{\mathcal{I}} = \sigma_1, \sigma_2, \ldots, \sigma_{k-1}, \sigma_k = b^{\mathcal{I}}$ with $\langle \sigma_i, \sigma_{i+1} \rangle \in S^{\mathcal{I}}$. This sequence can be split into subsequences at elements $o_i^{\mathcal{I}}$ for which there is a $o_i \in \mathsf{N}_I$, i.e., at named individuals, leaving us with subsequences (i) of subsequent named individuals $o_i^{\mathcal{I}}, o_{i+1}^{\mathcal{I}}$ or (ii) of the shape $o_i^{\mathcal{I}} = \sigma_{i,1}, \sigma_{i,2}, \ldots, \sigma_{i,n-1}, \sigma_{i,n} = o_{i+1}^{\mathcal{I}}$ with $\sigma_{i,2}, \ldots, \sigma_{i,n-1}$ unnamed individuals. For case (ii), Lemma 11 (2) guarantees $o_i^{\mathcal{I}} = o_{i+1}^{\mathcal{I}}$ and $\sigma_{i,2} = \sigma_{i,n-1}$, which implies $o_i^{\mathcal{I}} \in (\exists(R \sqcap R^-).\top)^{\mathcal{I}}$. Then, due to the according axiom $\exists(R \sqcap R^-).\top \sqsubseteq \mathit{Self}_R$ in $\Theta_S(\mathrm{KB})$, we obtain $o_i^{\mathcal{I}} \in \mathit{Self}_R^{\mathcal{I}}$ and by the DL-safe rule $\mathit{Self}_R(x) \to R(x, x)$ we have $\langle o_i^{\mathcal{I}}, o_i^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$. Hence, we know that $R(o_i, o_{i+1})$ holds in $\mathcal{I}$ for all our subsequences $o_i^{\mathcal{I}} \ldots o_{i+1}^{\mathcal{I}}$. But then, a (possibly iterated) application of the DL-safe rule $R(x, y) \wedge R(y, z) \to R(x, z)$ also yields that $R(a, b)$ is valid in $\mathcal{I}$, contradicting our assumption. This finishes the proof. $\qquad\square$

6.3. **From $\mathcal{ALCHIQ}b$ to $\mathcal{ALCHI}b^{\leqslant}$.** We now show how any extended $\mathcal{ALCHIQ}b$ knowledge base KB can be transformed into an extended $\mathcal{ALCHI}b^{\leqslant}$ knowledge base $\Theta_{\geqslant}(\mathrm{KB})$. The difference between the two DLs is that the latter does not allow $\geqslant$ number restrictions. This transformation (as well as the one presented in Section 6.5) makes use of the Boolean role constructors and differs

conceptually and technically from another method for removing qualified number restrictions from DLs described by DeGiacomo and Lenzerini (1994).

Given an $\mathcal{ALCHIQ}b$ knowledge base KB, the $\mathcal{ALCHI}b^{\leqslant}$ knowledge base $\Theta_{\geqslant}(\text{KB})$ is obtained by first flattening KB and then iteratively applying the following procedure to FLAT(KB), terminating if no $\geqslant$ restrictions are left:

- Choose an occurrence of $\geqslant n\, U.A$ in the knowledge base.
- Substitute this occurrence by $\exists R_1.A \sqcap \ldots \sqcap \exists R_n.A$, where $R_1, \ldots, R_n$ are fresh role names.
- For every $i \in \{1, \ldots, n\}$, add $R_i \sqsubseteq U$ to the knowledge base's RBox.
- For every $1 \leq i < k \leq n$, add $\forall(R_i \sqcap R_k).\bot$ to the knowledge base.

Observe that this transformation can be done in polynomial time, assuming a unary encoding of the numbers n. It remains to show that KB and $\Theta_{\geqslant}(\text{KB})$ are indeed equisatisfiable.

Lemma 13. *Let* KB *be an extended $\mathcal{ALCHIQ}b$ knowledge base. Then we have that the extended $\mathcal{ALCHI}b^{\leqslant}$ knowledge base $\Theta_{\geqslant}(\text{KB})$ and* KB *are equisatisfiable.*

Proof First we prove that every model of $\Theta_{\geqslant}(\text{KB})$ is a model of KB. We do so by an inductive argument, showing that no additional models can be introduced in any substitution step of the above conversion procedure. Hence, assume KB'' is an intermediate knowledge base that has a model $\mathcal{I}$, and that is obtained from KB' by eliminating the occurrence of $\geqslant n\, U.A$ as described above. Considering KB'', we find due to the KB'' axioms $\forall(R_i \sqcap R_k).\bot$ that no two individuals $\delta, \delta' \in \Delta^{\mathcal{I}}$ can be connected by more than one of the roles $R_1, \ldots, R_n$. In particular, this enforces $\delta' \neq \delta''$, whenever $\langle \delta, \delta' \rangle \in R_i^{\mathcal{I}}$ and $\langle \delta, \delta'' \rangle \in R_j^{\mathcal{I}}$ for distinct R_i and R_j. Now consider an arbitrary $\delta \in (\exists R_1.A \sqcap \ldots \sqcap \exists R_n.A)^{\mathcal{I}}$. This ensures the existence of individuals $\delta_1, \ldots, \delta_n$ with $\langle \delta, \delta_i \rangle \in R_i^{\mathcal{I}}$ and $\delta_i \in A^{\mathcal{I}}$ for $1 \leq i \leq n$. By the above observation, all such δ_i are pairwise distinct. Moreover, the axioms $R_i \sqsubseteq U$ ensure $\langle \delta, \delta_i \rangle \in U^{\mathcal{I}}$ for all i, hence we find that $\delta \in (\geqslant n\, U.A)^{\mathcal{I}}$. So we know $(\exists R_1.A \sqcap \ldots \sqcap \exists R_n.A)^{\mathcal{I}} \subseteq (\geqslant n\, U.C)^{\mathcal{I}}$. From the fact that both of those concept expressions occur outside any negation or quantifier scope (as the transformation starts with a flattened knowledge base and does not itself introduce such nestings) in axioms $D'' \in \text{KB}''$ and $D' \in \text{KB}'$ which are equal up to the substituted occurrence, we can derive that $D''^{\mathcal{I}} \subseteq D'^{\mathcal{I}}$. Then, from $D''^{\mathcal{I}} = \Delta^{\mathcal{I}}$ follows $D'^{\mathcal{I}} = \Delta^{\mathcal{I}}$ making D' valid in $\mathcal{I}$. Apart from D', all other axioms from KB' coincide with those from KB'' and hence are naturally satisfied in $\mathcal{I}$. So we find that $\mathcal{I}$ is a model of KB'.

At the end of our inductive chain, we finally arrive at FLAT(KB) which is equisatisfiable to KB by Proposition 1.

Second, we show that $\Theta_{\geqslant}(\text{KB})$ has a model if KB has. By Proposition 1, satisfiability of KB entails the existence of a model of FLAT(KB). Moreover, every model of FLAT(KB) can be transformed to a model of $\Theta_{\geqslant}(\text{KB})$, as we will show using the same inductive strategy as above by doing iterated model transformations following the syntactic knowledge base conversions. Again, assume KB'' is an intermediate knowledge base obtained from KB' by eliminating the occurrence of $\geqslant n\, U.A$ as described above, and suppose $\mathcal{I}$ is a model of KB'. Based on $\mathcal{I}$, we now (nondeterministically) construct an interpretation $\mathcal{J}$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}}$,
- for all $C \in \mathsf{N}_C$, let $C^{\mathcal{J}} := C^{\mathcal{I}}$,
- for all $S \in \mathsf{N}_R \setminus \{R_i \mid 1 \leq i \leq n\}$, let $S^{\mathcal{J}} := S^{\mathcal{I}}$,
- for every $\delta \in (\geqslant n\, U.A)^{\mathcal{I}}$, choose pairwise distinct $\epsilon_1^{\delta}, \ldots, \epsilon_n^{\delta}$ with $\langle \delta, \epsilon_i^{\delta} \rangle \in U^{\mathcal{I}}$ and $\epsilon_i^{\delta} \in A^{\mathcal{I}}$ (their existence being ensured by δ's aforementioned concept membership) and let $R_i^{\mathcal{J}} := \{\langle \delta, \epsilon_i^{\delta} \rangle \mid \delta \in (\geqslant n\, U.A)^{\mathcal{I}}\}$.

Now, it is easy to see that $\mathcal{J}$ satisfies all newly introduced axioms of the shape $\forall(R_i \sqcap R_k).\bot$, as the ϵ_i^δ have been chosen to be distinct for every δ. Moreover the axioms $R_i \sqsubseteq U$ are obviously satisfied by construction. Finally, for all $\delta \in (\geqslant n\, U.A)^\mathcal{I}$ the construction ensures $\delta \in (\exists R_1.A \sqcap \ldots \sqcap \exists R_n.A)^\mathcal{J}$ witnessed by the respective ϵ_i^δ. So we have $(\geqslant n\, U.A)^\mathcal{I} \subseteq (\exists R_1.A \sqcap \ldots \sqcap \exists R_n.A)^\mathcal{J}$. Now, again exploiting the fact that both of those concept expressions occur in negation normalized universal concept axioms $D' \in \mathrm{KB}'$ and $D'' \in \mathrm{KB}''$ that are equal up to the substituted occurrence, we can derive that $D'^\mathcal{I} \subseteq D''^\mathcal{J}$. Then, from $D'^\mathcal{I} = \Delta^\mathcal{I}$ follows $D''^\mathcal{J} = \Delta^\mathcal{J}$ making D'' valid in $\mathcal{J}$. Apart from D' (and the newly introduced axioms considered above), all other axioms from KB'' coincide with those from KB' and hence are satisfied in $\mathcal{J}$, as they do not depend on the R_i whose interpretations are the only ones changed in $\mathcal{J}$ compared to $\mathcal{I}$. So we find that $\mathcal{J}$ is a model of KB''. $\qquad\square$

6.4. From $\mathcal{ALCHI}b^\leqslant$ to $\mathcal{ALCI}b^\leqslant$.

In the presence of restricted role expressions, role subsumption axioms can be easily transformed into TBox axioms, as the subsequent lemma shows. This allows to dispense with role hierarchies in $\mathcal{ALCHI}b^\leqslant$ thereby restricting it to $\mathcal{ALCI}b^\leqslant$.

Lemma 14. *For any two restricted role expressions U and V, the RBox axiom $U \sqsubseteq V$ and the TBox axiom $\forall(U \sqcap \neg V).\bot$ are equivalent.*

Proof By the semantics' definition, $U \sqsubseteq V$ holds in an interpretation $\mathcal{I}$ exactly if for every two individuals δ, δ' with $\langle \delta, \delta' \rangle \in U^\mathcal{I}$ it also holds that $\langle \delta, \delta' \rangle \in V^\mathcal{I}$. This in turn is the case if and only if there are no δ, δ' with $\langle \delta, \delta' \rangle \in U^\mathcal{I}$ but $\langle \delta, \delta' \rangle \notin V^\mathcal{I}$ (the latter being expressible as $\langle \delta, \delta' \rangle \in (\neg V)^\mathcal{I}$). This condition can be formulated as $(U \sqcap \neg V)^\mathcal{I} = \emptyset$, which is equivalent to $\forall(U \sqcap \neg V).\bot$. $\qquad\square$

Note that $U \sqcap \neg V$ is restricted (hence an admissible role expression) whenever U is – this can be seen from the fact that $\emptyset \nvdash U$ implies $\emptyset \nvdash U \sqcap \neg V$ due to the definition of $\vdash$ and the Boolean role operator $\sqcap$. Consequently, for any extended $\mathcal{ALCHI}b^\leqslant$ knowledge base KB, let $\Theta_\mathcal{H}(\mathrm{KB})$ denote the $\mathcal{ALCI}b^\leqslant$ knowledge base obtained by substituting every RBox axiom $U \sqsubseteq V$ by the TBox axiom $\forall(U \sqcap \neg V).\bot$. The above lemma assures equivalence of KB and $\Theta_\mathcal{H}(\mathrm{KB})$ (and hence also their equisatisfiability). Obviously, this reduction can be done in linear time.

6.5. From $\mathcal{ALCI}b^\leqslant$ to $\mathcal{ALCIF}b$.

The elimination of the $\leqslant$ concept descriptions from an extended $\mathcal{ALCI}b^\leqslant$ knowledge base is more intricate than the previously described transformations. Thus, to simplify our subsequent presentation, we assume that all Boolean role expressions U occurring in concept expressions of the shape $\leqslant n\, U.C$ are atomic, i.e. $U \in \mathbf{R}$. This can be easily achieved by introducing a new role name R_U and substituting $\leqslant n\, U.C$ by $\leqslant n\, R_U.C$ as well as adding the two TBox axioms $\forall(U \sqcap \neg R_U).\bot$ and $\forall(\neg U \sqcap R_U).\bot$ (this ensures that the interpretations of U and R_U always coincide).

To further make the presentation more conceivable, we subdivide it into two steps: first we eliminate concept expressions of the shape $\leqslant n\, R.C$ merely leaving axioms of the form $\leqslant 1\, R.\top$ (also known as role functionality statements) as the only occurrences of number restrictions, hence obtaining an $\mathcal{ALCIF}b$ knowledge base.[9] Then, in a second step discussed in the next section, we eliminate all occurrences of axioms of the shape $\leqslant 1\, R.\top$.

Let KB an $\mathcal{ALCI}b^\leqslant$ knowledge base. We obtain the $\mathcal{ALCIF}b$ knowledge base $\Theta_\leqslant(\mathrm{KB})$ by first flattening KB and then successively applying the following steps (stopping when no further such occurrence is left):

---

[9]Following the notational convention, we use $\mathcal{F}$ to indicate the modeling feature of role functionality.

- Choose an occurrence of the shape $\leq n\, R.A$ which is not a functionality axiom $\leq 1\, R.\top$,
- substitute this occurrence by $\forall(R \sqcap \neg R_1 \sqcap \ldots \sqcap \neg R_n).\neg A$ where $R_1, \ldots, R_n$ are fresh role names,
- for every $i \in \{1, \ldots, n\}$, add $\forall R_i.A$ as well as $\leq 1\, R_i.\top$ to the knowledge base.

This transformation can clearly be done in polynomial time, again assuming a unary encoding of the number n. We now show that this conversion yields an equisatisfiable extended knowledge base. Structurally, the proof is similar to that of Lemma 13.

Lemma 15. *Given an extended $\mathcal{ALCIb}^{\leq}$ knowledge base* KB*, the extended $\mathcal{ALCIF}b$ knowledge base* $\Theta_{\leq}(KB)$ *and* KB *are equisatisfiable.*

Proof KB and FLAT(KB) are equisatisfiable by Proposition 1, so it remains to show equisatisfiability of FLAT(KB) and $\Theta_{\leq}(KB)$.

First, we prove that every model of $\Theta_{\leq}(KB)$ is a model of FLAT(KB). We do so in an inductive way by showing that no additional models can be introduced in any substitution step of the above conversion procedure. Hence, assume KB'' is an intermediate knowledge base with model $\mathcal{I}$, and that is obtained from KB' by eliminating the occurrence of $\leq n\, R.A$ as described above. Now consider an arbitrary $\delta \in (\forall(R \sqcap \neg R_1 \sqcap \ldots \sqcap \neg R_n).\neg A)^{\mathcal{I}}$. This ensures that whenever an individual $\delta' \in \Delta^{\mathcal{I}}$ satisfies $\langle \delta, \delta' \rangle \in R^{\mathcal{I}}$ and $\delta' \in A$, it must additionally satisfy $\langle \delta, \delta' \rangle \in R_i^{\mathcal{I}}$ for one $i \in \{1, \ldots, n\}$. However, it follows from the KB''-axioms $\leq 1\, R_i.\top$ that there is at most one such δ' for each R_i. Thus, there can be at most n individuals δ' with $\langle \delta, \delta' \rangle \in R^{\mathcal{I}}$ and $\delta' \in A$. This implies $\delta \in (\leq n\, R.A)^{\mathcal{I}}$. So we have $(\forall(R \sqcap \neg R_1 \sqcap \ldots \sqcap \neg R_n).\neg A)^{\mathcal{I}} \subseteq (\leq n\, R.A)^{\mathcal{I}}$. Due to the flattened knowledge base structure, both of those concept expressions occur outside the scope of any negation or quantifier within axioms $D'' \in KB''$ and $D' \in KB'$ that are equal up to the substituted occurrence. Hence, we can derive that $D''^{\mathcal{I}} \subseteq D'^{\mathcal{I}}$. Then, from $D''^{\mathcal{I}} = \Delta^{\mathcal{I}}$ follows $D'^{\mathcal{I}} = \Delta^{\mathcal{I}}$ making D' valid in $\mathcal{I}$. Apart from D', all other axioms from KB' are contained in KB'' and hence are naturally satisfied in $\mathcal{I}$. So we find that $\mathcal{I}$ is a model of KB' as well.

Second, we show that every model of FLAT(KB) can be transformed to a model of $\Theta_{\leq}(KB)$. We use the same induction strategy as above by doing iterated model transformations following the syntactic knowledge base conversions. Again, assume KB'' is an intermediate knowledge base obtained from KB' by eliminating the occurrence of a $\leq n\, R.C$ as described above, and suppose $\mathcal{I}$ is a model of KB'. Based on $\mathcal{I}$, we now (nondeterministically) construct an interpretation $\mathcal{J}$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}}$,
- for all $C \in N_C$, let $C^{\mathcal{J}} := C^{\mathcal{I}}$,
- for all $S \in N_R \setminus \{R_i \mid 1 \leq i \leq n\}$, let $S^{\mathcal{J}} := S^{\mathcal{I}}$,
- for every $\delta \in (\leq n\, R.A)^{\mathcal{I}}$, let $\epsilon_1^{\delta}, \ldots, \epsilon_k^{\delta}$ be an exhaustive enumeration (with arbitrary but fixed order) of all those $\epsilon \in \Delta^{\mathcal{I}}$ with $\langle \delta, \epsilon \rangle \in R^{\mathcal{I}}$ and $\epsilon \in A^{\mathcal{I}}$. Thereby δ's aforementioned concept membership ensures $k \leq n$. Now, let $R_i^{\mathcal{J}} := \{\langle \delta, \epsilon_i^{\delta} \rangle \mid \delta \in (\leq n\, R.A)^{\mathcal{I}}\}$.

Now, it is easy to see that $\mathcal{J}$ satisfies all newly introduced axioms of the shape $\leq 1\, R_i.\top$ as every δ has at most one R_i-successor (namely ϵ_i^{δ}, if $\delta \in (\leq n\, R.A)^{\mathcal{I}}$, and none otherwise). Moreover, the axioms $\forall R_i.A$ are satisfied, as the ϵ_i^{δ} have been chosen accordingly.

Finally for all $\delta \in (\leq n\, R.A)^{\mathcal{I}}$ the construction ensures $\delta \in (\forall(R \sqcap \neg R_1 \sqcap \ldots \sqcap \neg R_n).\neg A)^{\mathcal{J}}$ as by construction, each R-successor of δ that lies within the extension of A is contained in $\epsilon_1^{\delta}, \ldots, \epsilon_k^{\delta}$ and therefore also R_i-successor of δ for some i. Now, again exploiting the fact that both of those concept expressions occur in negation normalized universal concept axioms $D' \in KB'$ and $D'' \in KB''$ that are equal up to the substituted occurrence, we can derive that $D'^{\mathcal{I}} \subseteq D''^{\mathcal{J}}$. Then, from

$D'^{\mathcal{I}} = \Delta^{\mathcal{I}}$ follows $D''^{\mathcal{J}} = \Delta^{\mathcal{J}}$ making D'' valid in $\mathcal{J}$. Apart from D'' (and the newly introduced axioms considered above), all other axioms from KB$''$ coincide with those from KB$'$ and hence are satisfied in $\mathcal{J}$, as they do not depend on the R_i whose interpretations are the only ones changed in $\mathcal{J}$ compared to $\mathcal{I}$. So we find that $\mathcal{J}$ is a model of KB$''$. $\square$

6.6. **From $\mathcal{ALCIF}b$ to $\mathcal{ALCI}b$.** In the sequel, we show how the role functionality axioms of the shape $\leqslant 1\,R.\top$ can be eliminated from an $\mathcal{ALCIF}b$ knowledge base while still preserving equisatisfiability. Partially, the employed rewriting is the same as the one proposed for $\mathcal{ALCIF}$ TBoxes by Calvanese et al. (1998), however, in the presence of ABoxes more needs to be done.

Essentially, the idea is to add axioms that enforce that for every functional role R, any two R-successors coincide with respect to their properties expressible in "relevant" DL role and concept expressions. To this end, we consider the parts of a knowledge base as defined in Section 2 on page 5. While it is not hard to see that the introduced axioms follow from R's functionality, the other direction (a Leibniz-style "identitas indiscernibilium" argument) needs a closer look.

Taking an extended $\mathcal{ALCIF}b$ knowledge base KB, let $\Theta_{\mathcal{F}}(KB)$ denote the extended $\mathcal{ALCI}b$ knowledge base obtained from KB by removing every role functionality axiom $\leqslant 1\,R.\top$ and instead adding

- $\forall R.\neg D \sqcup \forall R.D$ for every $D \in P(KB \setminus \{\alpha \in KB \mid \alpha = \leqslant 1\,R.\top$ for some $R \in \mathbf{R}\})$,
- $\forall (R \sqcap S).\bot \sqcup \forall (R \sqcap \neg S).\bot$ for every atomic role S from KB, as well as
- the DL-safe rule $R(x,y), R(x,z) \to y \approx z$.

Clearly, this transformation can also be done in polynomial time and space w.r.t. the size of KB.

Our goal is now to prove equisatisfiability of KB and $\Theta_{\mathcal{F}}(KB)$. The following lemma establishes the easier direction of this correspondence.

Lemma 16. *Any $\mathcal{ALCIF}b$ knowledge base* KB *entails all axioms of the $\mathcal{ALCI}b$ knowledge base* $\Theta_{\mathcal{F}}(KB)$, *i.e.* KB $\models \Theta_{\mathcal{F}}(KB)$.

Proof Let $\mathcal{J}$ be a model of KB. We need to show that $\mathcal{J}$ also satisfies the additional rules and axioms introduced in $\Theta_{\mathcal{F}}(KB)$.

First let D be an arbitrary concept. Note that $\forall R.\neg D \sqcup \forall R.D$ is equivalent to the GCI $\exists R.D \sqsubseteq \forall R.D$. This is satisfied if, for any $\delta \in \Delta^{\mathcal{J}}$, if δ has an R-successor in $D^{\mathcal{J}}$, then all R-successors of δ are in $D^{\mathcal{J}}$. This is trivially satisfied if δ has at most one R-successor, which holds since $\mathcal{J}$ satisfies the functionality axiom $\leqslant 1\,R.\top \in KB$. Since we have shown the satisfaction for arbitrary concepts D, this holds in particular for those from $P(KB \setminus \{\alpha \in KB \mid \alpha = \leqslant 1\,R.\top$ for some $R \in \mathbf{R}\})$.

Second, let S be an atomic role. Mark that $\forall (R \sqcap S).\bot \sqcup \forall (R \sqcap \neg S).\bot$ is equivalent to the GCI $\exists (R \sqcap S).\top \sqsubseteq \forall (R \sqcap \neg S).\bot$. This means that for any $\delta \in \Delta^{\mathcal{J}}$, all R-successors are also S-successors of it, whenever one of them is. Again, this is trivially satisfied as δ has at most one R-successor.

Finally all newly introduced rules of the form $R(x,y), R(x,z) \to y \approx z$ are satisfied in $\mathcal{J}$ as a consequence of the functionality statements in KB. $\square$

The other direction for showing equisatisfiability, which amounts to finding a model of KB given one for $\Theta_{\mathcal{F}}(KB)$, is somewhat more intricate and requires some intermediate considerations.

Lemma 17. *If* KB *is an $\mathcal{ALCIF}b$ knowledge base with $\leqslant 1\,R.\top \in KB$ then in every model $\mathcal{J}$ of* $\Theta_{\mathcal{F}}(KB)$ *we find that $\langle \delta, \delta_1 \rangle \in R^{\mathcal{J}}$ and $\langle \delta, \delta_2 \rangle \in R^{\mathcal{J}}$ imply*

- *for all $C \in P(KB \setminus \{\alpha \in KB \mid \alpha = \leqslant 1\,R.\top$ for some $R \in \mathbf{R}\})$, we have $\delta_1 \in C^{\mathcal{J}}$ iff $\delta_2 \in C^{\mathcal{J}}$,*
- *for all $S \in \mathsf{N}_R$, we have $\langle \delta, \delta_1 \rangle \in S^{\mathcal{J}}$ iff $\langle \delta, \delta_2 \rangle \in S^{\mathcal{J}}$.*

Proof For the first proposition, assume $\delta_1 \in C^{\mathcal{J}}$. From $\langle \delta, \delta_1 \rangle \in R^{\mathcal{J}}$ follows $\delta \in (\exists R.C)^{\mathcal{J}}$. Due to the $\Theta_{\mathcal{F}}(\text{KB})$ axiom $\forall R.\neg C \sqcup \forall R.C$ (being equivalent to the GCI $\exists R.C \sqsubseteq \forall R.C$) follows $\delta \in (\forall R.C)^{\mathcal{J}}$. Since $\langle \delta, \delta_2 \rangle \in R^{\mathcal{J}}$, this implies $\delta_2 \in C^{\mathcal{J}}$. The other direction follows by symmetry.

To show the second proposition, assume $\langle \delta, \delta_1 \rangle \in S^{\mathcal{J}}$. Since also $\langle \delta, \delta_1 \rangle \in R^{\mathcal{J}}$, we have $\langle \delta, \delta_1 \rangle \in R \sqcap S^{\mathcal{J}}$ and hence $\delta \in (\exists (R \sqcap S).\top)^{\mathcal{J}}$. From the $\Theta_{\mathcal{F}}(\text{KB})$ axiom $\forall (R \sqcap S).\bot \sqcup \forall (R \sqcap \neg S).\bot$ (which is equivalent to the GCI $\exists (R \sqcap S).\top \sqsubseteq \neg \exists (R \sqcap \neg S).\top$) we conclude $\delta \in (\neg \exists (R \sqcap \neg S).\top)^{\mathcal{J}}$, in words: δ has no R-successor that is not its S-successor. Thus, as $\langle \delta, \delta_2 \rangle \in R^{\mathcal{J}}$, it must also hold that $\langle \delta, \delta_2 \rangle \in S^{\mathcal{J}}$. Again, the other direction follows by symmetry. $\square$

In order to convert a model of $\Theta_{\mathcal{F}}(\text{KB})$ into one of KB, we will have to enforce role functionality where needed by cautiously deleting individuals from the original model. Definition 13 will provide a method for this. To this end, some auxiliary notions defined beforehand will come in handy.

Definition 12. *Let $\mathcal{J}$ be an interpretation, and let $\mathcal{I}$ be the unraveling of $\mathcal{J}$.*[10] *For a domain element $\sigma \in \Delta^{\mathcal{I}}$ and an $R \in \mathbf{R}$, we define the set of R-neighbors of σ in $\mathcal{I}$ by $nb_{\mathcal{I}}^R(\sigma) := \{\sigma' \mid \langle \sigma, \sigma' \rangle \in R^{\mathcal{I}}\}$. Among the R-neighbors, we distinguish between* subordinate R-neighbors $sub_{\mathcal{I}}^R(\sigma) := \{\sigma\delta \mid \langle \sigma, \sigma\delta \rangle \in R^{\mathcal{I}}\}$ *and the* non-subordinate R-neighbors $nonsub_{\mathcal{I}}^R(\sigma) := nb_{\mathcal{I}}^R(\sigma) \setminus sub_{\mathcal{I}}^R(\sigma)$.

Definition 13. *Let $\mathcal{J}$ be an interpretation, and let $\mathcal{I}$ be the unraveling of $\mathcal{J}$. Given an extended $\mathcal{ALCIF}b$ knowledge base KB, let $\text{KB}^* := \text{KB} \setminus \{\alpha \in \text{KB} \mid \alpha = \,\leqslant 1\, R.\top$ for some $R \in \mathbf{R}\}$, let $\mathcal{D} := P(\text{KB})$ and let $\mathcal{S} := \{R \mid \leqslant 1\, R.\top \in \text{KB}\}$.*

Then, an interpretation $\mathcal{K}$ will be called KB-pruning *of $\mathcal{I}$, if $\mathcal{K}$ can be constructed from $\mathcal{I}$ in the following way: Let first $\Delta_0 = \Delta^{\mathcal{I}}$. Next, iteratively determine Δ_{i+1} from Δ_i as follows:*

- *Select a word-length minimal σ from Δ_i where there is an $S \in \mathcal{S}$ for which $nb_{\mathcal{I}}^S(\sigma) > 1$ and $sub_{\mathcal{I}}^S(\sigma) > 0$.*
- *If $nonsub_{\mathcal{I}}^S(\sigma) > 0$, let $\Delta' = sub_{\mathcal{I}}^S(\sigma)$, otherwise let $\Delta' = sub_{\mathcal{I}}^S(\sigma) \setminus \{\sigma'\}$ for an arbitrarily chosen $\sigma' \in sub_{\mathcal{I}}^S(\sigma)$.*
  Delete Δ' from Δ_i as well as all σ^{**} having some $\sigma^* \in \Delta'$ as prefix.

Finally, let $\mathcal{K}$ be the limit of this process: $\Delta^{\mathcal{K}} := \bigcap_{i \in \mathbb{N}} \Delta_i$ and $\cdot^{\mathcal{K}}$ is the function $\cdot^{\mathcal{I}}$ restricted to $\Delta^{\mathcal{K}}$.

Roughly speaking, any KB-*pruning* of $\mathcal{I}$ is (nondeterministically) constructed by deleting surplus functional-role-successors. Mark that the tree-like structure of non-named individuals of the unraveling is crucial in order to make the process well-defined.

Lemma 18. *Let KB be an extended $\mathcal{ALCIF}b$ knowledge base, let $\mathcal{J}$ be a model of $\Theta_{\mathcal{F}}(\text{KB})$ and let $\mathcal{I}$ be an unraveling of $\mathcal{J}$. Then, any KB-pruning $\mathcal{K}$ of $\mathcal{I}$ is a model of KB.*

Proof By construction, we know that $\mathcal{I}$ is a model of $\Theta_{\mathcal{F}}(\text{KB})$. Now, let $\mathcal{K}$ be a KB-pruning of $\mathcal{I}$. For showing $\mathcal{K} \models \text{KB}$, we divide KB into two sets, namely the set of role functionality axioms $\{\alpha \in \text{KB} \mid \alpha = \,\leqslant 1\, R.\top$ for some $R \in \mathbf{R}\}$ and all the remaining axioms, denoted by KB^*, and show $\mathcal{K} \models \text{KB}^*$ and $\mathcal{K} \models \{\alpha \in \text{KB} \mid \alpha = \,\leqslant 1\, R.\top$ for some $R \in \mathbf{R}\}$ separately.

We start by showing $\mathcal{K} \models \text{KB}^*$. To this end, we prove that, for each $C \in P(\text{KB}^*)$ and for every individual σ from $\mathcal{K}$, we have $\sigma \in C^{\mathcal{K}}$ exactly if $\sigma \in C^{\mathcal{I}}$. Clearly, this statement extends to concepts that are Boolean combinations of elements from $P(\text{KB}^*)$, i.e., to all axioms in KB^*. We omit this easy structural induction.

The claim for $C \in P(\text{KB}^*)$ is shown by induction over the depth of role restrictions in C, and we assume that is has already been shown for concepts of smaller role depth. We consider three cases:

---

[10]Remember that by construction, the individuals of $\mathcal{I}$ are sequences of individuals of $\mathcal{J}$. For better readability, we will strictly use σ – with possible subscripts – for $\mathcal{I}$-individuals and δ for $\mathcal{J}$-individuals.

- $C \in \mathsf{N}_C \cup \{\top, \bot\}$
  Then the coincidence follows directly from the construction of $\mathcal{K}$.
- $C = \exists U.D$
  "$\Rightarrow$" $\sigma \in (\exists U.D)^{\mathcal{K}}$ means that there is a $\mathcal{K}$-individual σ' with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{K}}$ and $\sigma' \in D^{\mathcal{K}}$. Because of the construction of $\mathcal{K}$ by pruning $\mathcal{I}$, this means also $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and by induction hypothesis, we have $\sigma' \in D^{\mathcal{I}}$, ergo $\sigma \in (\exists U.D)^{\mathcal{I}}$.
  "$\Leftarrow$" If $\sigma \in (\exists U.D)^{\mathcal{I}}$, there is an $\mathcal{I}$-individual σ' with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \in D^{\mathcal{I}}$. In case σ' is not deleted during the construction of $\mathcal{K}$, it proves (by using the induction hypothesis on D) that $\sigma \in (\exists U.D)^{\mathcal{K}}$. Otherwise, it must have been deleted due to the existence of another $\mathcal{I}$-individual σ'' for with Lemma 17 ensures $\{R \in \mathbf{R} \mid \langle \sigma, \sigma'' \rangle \in R^{\mathcal{I}}\} = \{R \in \mathbf{R} \mid \langle \sigma, \sigma' \rangle \in R^{\mathcal{I}}\}$ and $\{E \in P(\mathrm{KB}^*) \mid \sigma'' \in E^{\mathcal{I}}\} = \{E \in P(\mathrm{KB}^*) \mid \sigma' \in E^{\mathcal{I}}\}$. W.l.o.g., σ'' does not get deleted in the whole construction procedure. Yet, then the $\mathcal{K}$-individual σ'' obviously proves $\sigma \in (\exists U.D)^{\mathcal{K}}$.
- $C = \forall R.D$
  "$\Rightarrow$" Assume the contrary, i.e., $\sigma \in (\forall U.D)^{\mathcal{K}}$ but $\sigma \notin (\forall U.D)^{\mathcal{I}}$ which means that there is an $\mathcal{I}$-individual σ' with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ but $\sigma' \notin D^{\mathcal{I}}$. In case σ' has not been deleted during the construction of $\mathcal{K}$, it disproves $\sigma \in (\forall U.D)^{\mathcal{K}}$ (by invoking the induction hypothesis on D) leading to a contradiction. Otherwise, σ' is deleted because of the existence of another $\mathcal{I}$-individual σ'' for with Lemma 17 ensures $\{R \in \mathbf{R} \mid \langle \sigma, \sigma'' \rangle \in R^{\mathcal{I}}\} = \{R \in \mathbf{R} \mid \langle \sigma, \sigma' \rangle \in R^{\mathcal{I}}\}$ and $\{E \in P(\mathrm{KB}^*) \mid \sigma'' \in E^{\mathcal{I}}\} = \{E \in P(\mathrm{KB}^*) \mid \sigma' \in E^{\mathcal{I}}\}$. W.l.o.g., σ'' does not get deleted in the whole construction procedure. Yet, then the $\mathcal{K}$-individual σ'' obviously contradicts $\sigma \in (\exists U.D)^{\mathcal{K}}$.
  "$\Leftarrow$" Assume the contrary, i.e., $\sigma \in (\forall U.D)^{\mathcal{I}}$ but $\sigma \notin (\forall U.D)^{\mathcal{K}}$. The latter means that there is a $\mathcal{K}$-individual σ' with $\langle \sigma, \sigma' \rangle \in U^{\mathcal{K}}$ and $\sigma' \notin D^{\mathcal{K}}$. Because of the construction of $\mathcal{K}$ by pruning $\mathcal{I}$, this means also $\langle \sigma, \sigma' \rangle \in U^{\mathcal{I}}$ and $\sigma' \notin D^{\mathcal{I}}$, ergo $\sigma \notin (\forall U.D)^{\mathcal{I}}$, contradicting the assumption.

We proceed by showing that every role R with $\leqslant 1\, R.\top \in \mathrm{KB}$ is functional in $\mathcal{K}$. Let $\sigma \in \Delta^{\mathcal{K}}$ and let σ_1, σ_2 be two R-successors of σ. We consider two cases: First, assume that $\sigma_1 = a_1^{\mathcal{K}}$ and $\sigma_2 = a_2^{\mathcal{K}}$ for $a_1, a_2 \in \mathsf{N}_I$. Then, by construction of the unraveling we can derive that there must be an $a_3 \in \mathsf{N}_I$ with $\sigma = a_3^{\mathcal{K}}$. However, then, the DL-safe rule $R(x, y), R(x, z) \to y \approx z$ from $\Theta_{\mathcal{F}}(\mathrm{KB})$ ensures $\sigma_1 = \sigma_2$. Next we consider the case that at least one of σ_1, σ_2 is unnamed. By Lemma 17 and the point-wise correspondence between $\mathcal{I}$ and $\mathcal{K}$ shown in the previous part of the proof, two statements hold: First, for all $C \in P(\mathrm{KB}^*)$, we have that $\sigma_1 \in C^{\mathcal{K}}$ iff $\sigma_2 \in C^{\mathcal{K}}$. Second, for all $S \in \mathsf{N}_R$ we have that $\langle \sigma, \sigma_1 \rangle \in S^{\mathcal{K}}$ iff $\langle \sigma, \sigma_2 \rangle \in S^{\mathcal{K}}$. However, in the pruning process generating $\mathcal{K}$, exactly such duplicate occurrences are erased, leaving at most one R-successor per σ. Thus we conclude $\sigma_1 = \sigma_2$. This completes the proof that all axioms from KB are satisfied in $\mathcal{K}$. □

Finally, we are ready to establish the equisatisfiability result also for this last transformation step.

Theorem 19. *For any extended $\mathcal{ALCIF}b$ knowledge base* KB*, the $\mathcal{ALCI}b$ knowledge base* $\Theta_{\mathcal{F}}(\mathrm{KB})$ *and* KB *are equisatisfiable.*

Proof Lemma 16 ensures that every model of KB is also a model of $\Theta_{\mathcal{F}}(\mathrm{KB})$. Moreover, by Lemma 18, given a model $\mathcal{J}$ for of $\Theta_{\mathcal{F}}(\mathrm{KB})$, any KB-pruning of $\mathcal{J}$'s unraveling (the existence of which is ensured by constructive definition) is a model of KB. This finishes the proof. □

Eventually, the results of this section can be composed to show how to transform an extended $\mathcal{SHIQ}b_s$ knowledge base KB into an equisatisfiable extended $\mathcal{ALCI}b$ knowledge base by computing $\Theta_{\mathcal{SHQ}}(\text{KB}) := \Theta_{\mathcal{F}}\Theta_{\leqslant}\Theta_{\mathcal{H}}\Theta_{\geqslant}\Theta_{\mathcal{S}}(\text{KB})$. Moreover, as each of the single transformation steps is time polynomial, so is the overall procedure. Therefore, we are able to check the satisfiability of any extended $\mathcal{SHIQ}$ knowledge base using the method presented in the previous sections, by first transforming it into $\mathcal{ALCI}b$ and then checking.

This result is recorded in the below theorem, where we also exploit it to show an even stronger result about the correspondence between KB and $\Theta_{\mathcal{SHQ}}(\text{KB})$.

Theorem 20. *Let* KB *be an extended* $\mathcal{SHIQ}b_s$ *knowledge base. Then the following hold:*

- KB *and* $\Theta_{\mathcal{SHQ}}(\text{KB})$ *are equisatisfiable,*
- KB $\models C(a)$ *iff* $\Theta_{\mathcal{SHQ}}(\text{KB}) \models C(a)$,
- KB $\models R(a, b)$ *iff* $\Theta_{\mathcal{SHQ}}(\text{KB}) \models R(a, b)$, *and*
- KB $\models a \approx b$ *iff* $\Theta_{\mathcal{SHQ}}(\text{KB}) \models a \approx b$,

for any $a, b \in \mathsf{N}_I$, $C \in \mathsf{N}_C$, *and* $R \in \mathsf{N}_R$.

Proof Equisatisfiability follows from the fact that each of the transformations $\Theta_{\mathcal{F}}, \Theta_{\leqslant}, \Theta_{\mathcal{H}}, \Theta_{\geqslant}, \Theta_{\mathcal{S}}$ preserves satisfiability. We then use the established equisatisfiability of KB and $\Theta_{\mathcal{SHQ}}(\text{KB})$ to prove the other claims. Assume KB $\models C(a)$. This means that the knowledge base KB$'$ obtained by extending KB with the DL-safe rule $C(a) \rightarrow$ is unsatisfiable. Now we observe that $\Theta_{\mathcal{SHQ}}(\text{KB}')$ is obtained by extending $\Theta_{\mathcal{SHQ}}(\text{KB})$ with $C(a) \rightarrow$. Since $\Theta_{\mathcal{SHQ}}(\text{KB}')$ is unsatisfiable, so is $\Theta_{\mathcal{SHQ}}(\text{KB})$ extended with $C(a) \rightarrow$, and hence $\Theta_{\mathcal{SHQ}}(\text{KB}) \models C(a)$ as required. The other direction of the claim follows via a similar argumentation. The remaining cases are shown analogously. □

Consolidating all our results, we now can formulate our main theorem for checking satisfiability as well as entailment of positive and negative ground facts for extended $\mathcal{SHIQ}b_s$ knowledge bases.

Theorem 21. *Let* KB *be an extended* $\mathcal{SHIQ}b_s$ *knowledge base and let*

$$\mathbb{P} := \mathsf{DD}(\Theta_{\mathcal{SHQ}}(\text{KB})).$$

Then the following hold:

- KB *is satisfiable iff* $\mathbb{P}$ *is,*
- KB $\models C(a)$ *iff* $\mathbb{P} \models S_C(a)$,
- KB $\models R(a, b)$ *iff* $\mathbb{P} \models S_R(a, b)$, *and*
- KB $\models a \approx b$ *iff* $\mathbb{P} \models a \approx b$,

for any $a, b \in \mathsf{N}_I$, $C \in \mathsf{N}_C$, *and* $R \in \mathsf{N}_R$.

Proof Combine Theorem 20 with Theorem 10. □

Note also that the above observation immediately allows us to add reasoning support for *DL-safe* conjunctive queries, i.e. conjunctive queries that assume all variables to range only over named individuals. It is easy to see that, as a minor extension, one could generally allow for concept expressions $\forall R.A$ and $\exists R.A$ in queries and rules, simply because $\mathsf{DD}(\text{KB})$ represents these elements of $P(\mathsf{FLAT}(\mathcal{T}))$ as atomic symbols in disjunctive Datalog.

7. RELATED WORK

Boolean constructors on roles have been investigated in the context of both description and modal logics. Borgida (1996) used them extensively for the definition of a DL that is equivalent to the two-variable fragment of FOL.

It was shown by Hustadt and Schmidt (2000) that the DL obtained by augmenting $\mathcal{ALC}$ with full Boolean role constructors ($\mathcal{ALB}$) is decidable. Lutz and Sattler (2001) established NExpTime-completeness of the standard reasoning tasks in this logic. Restricting to only role negation (Lutz and Sattler, 2001) or only role conjunction (Tobies, 2001) retains ExpTime-completeness. On the other hand, complexity does not increase beyond NExpTime even when allowing for inverses, qualified number restrictions, and nominals. This was shown by Tobies (2001) via a polynomial translation of $\mathcal{ALCOIQB}$ into C^2, the two variable fragment of first order logic with counting quantifiers, which in turn was proven to be NExpTime-complete by Pratt-Hartmann (2005). Also the description logic $\mathcal{ALBO}$ (Schmidt and Tishkovsky, 2007) falls in that range of NExpTime-complete DLs.

On the contrary, it was also shown by Tobies (2001) that restricting to *safe* Boolean role constructors keeps $\mathcal{ALC}$'s reasoning complexity in ExpTime, even when adding inverses and qualified number restrictions ($\mathcal{ALCQIb}$).

For logics including modeling constructs that deal with role composition like transitivity or – more general – complex role inclusion axioms, results on complexities in the presence of Boolean role constructors are more sparse. Lutz and Walther (2005) show that $\mathcal{ALC}$ can be extended by negation and regular expressions on roles while keeping reasoning within ExpTime. Furthermore, Calvanese et al. (2007b) provided ExpTime complexity for a similar logic that includes inverses and qualified number restriction but reverts to safe negation on roles. The present work showed that reasoning remains in ExpTime for extended $\mathcal{SHIQ}b_s$ knowledge bases. Regarding DLs that combine nominals and role composition, it was shown that *unsafe* Boolean role constructors can be added to $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ (resulting in DLs $\mathcal{SHOIQB}_s$ and $\mathcal{SROIQB}_s$) without affecting their respective worst-case complexities of NExpTime and N2ExpTime (Rudolph et al., 2008a). The restriction to simple roles, on the other hand, is essential to retain decidability. Furthermore, conjunctions of simple roles (which are trivially safe in the absence of role negation) can be added to tractable DLs of the $\mathcal{EL}$ and DLP families without increasing their worst-case complexity (Rudolph et al., 2008a).

Type-based reasoning techniques have been described sporadically in the area of DLs but never been practically adopted.

Lutz et al. (2005) use a particular kind of types, called *mosaics* for finite model reasoning. Eiter et al. (2009) use similar structures, called *knots* for query answering in the description logic $\mathcal{SHIQ}$. Both notions show a similarity to the notion of *(counting) star types* used for reasoning in fragments of first order logic (Pratt-Hartmann, 2005), in that they do not only store information about single domain individuals but also about all their direct neighbors. As opposed to this, our notion of dominoes exhibits more similarity to the notion of (non-counting) *two-types* used in first-order logic, e.g., by Grädel et al. (1997); both notions encode information related to pairs of domain individuals (rather than whole neighborhoods).

The approach of constructing a canonical model (resp. a sufficient representation of it) in a downward manner (i.e., by pruning a larger structure) shows some similarity to Pratt's type elimination technique (Pratt, 1979), originally used to decide satisfiability of modal formulae.

Canonical models themselves have been a widely used notion in modal logic (Popkorn, 1994; Blackburn et al., 2001), however, due to the additional expressive power of $\mathcal{ALCIb}$ compared to standard modal logics like K (being the modal logic counterpart of the description logic $\mathcal{ALC}$), we had to

substantially modify the notion of a canonical model used there: in order to cope with number restrictions, we use infinite tree models based on unravelings whereas the canonical models in the mentioned approaches are normally finite and obtained via filtrations.

Related in spirit (namely to use BDD-based reasoning for DL reasoning tasks and to use a type elimination-like technique for doing so) is the work presented by Pan et al. (2006). However, the established results as well as the approaches differ greatly from ours: the authors establish a procedure for deciding the satisfiability of $\mathcal{ALC}$ concepts in a setting not allowing for general TBoxes, while our approach can check satisfiability of $\mathcal{SHIQ}$ (resp. $\mathcal{ALCIb}$) knowledge bases supporting general TBoxes, thereby generalizing the results by Pan et al. (2006) significantly.

The presented method for reasoning with DL-safe rules and assertional data exhibits similarities to the algorithm underlying the KAON2 reasoner (Motik, 2006; Hustadt et al., 2007, 2008). In particular, pre-transformations are first applied to $\mathcal{SHIQ}$ knowledge bases, before a saturation procedure is applied to the TBox part that results in a disjunctive Datalog program that can be combined with the assertional part of the knowledge base. As in our case, extensions with DL-safe rules and ground conjunctive queries are possible. The processing presented here, however, is very different from KAON2. Besides using OBDDs, it also employs Boolean role constructors that admit an indirect encoding of number restrictions. Moreover, as opposed to our approach, the transformation in Motik (2006) does not preserve *all* ground consequences: $\mathcal{SHIQ}$ consequences of the form $R(a, b)$ with R being non-simple may not be entailed by the created Datalog program. This shortcoming could, however, be easily corrected along the lines of our approach. On the other hand, the KAON2 transformation avoids the use of disjunctions in Datalog for knowledge bases that are Horn (i.e., free of disjunctive information). Reasoning for Horn-$\mathcal{SHIQ}$ can thus be done in ExpTime, which is worst-case optimal (Krötzsch et al., 2012). In contrast, our OBDD encoding requires disjunctive Datalog in all cases, leading to a NExpTime procedure even for Horn-$\mathcal{SHIQ}$.

8. Discussion

We have presented a new worst-case optimal reasoning algorithm for standard reasoning tasks for extended $\mathcal{SHIQb}_s$ knowledge bases. The algorithm compiles $\mathcal{SHIQb}_s$ terminologies into disjunctive Datalog programs, which are then combined with assertional information and DL-safe rules for satisfiability checking and (ground) query answering. To this end, OBDDs are used as a convenient intermediate data structure to process terminologies and are subsequently transformed into disjunctive Datalog programs that can naturally account for ABox data and DL-safe rules. The generation of disjunctive Datalog may require exponentially many computation steps, the cost of which depends on the concrete OBDD implementation at hand – finding *optimal OBDD encodings* is NP-complete but heuristic approximations are often used in practice. Querying the disjunctive Datalog program then is co-NP-complete w.r.t. the size of the ABox, so that the data complexity of the algorithm is worst-case optimal (Motik, 2006). Concerning combined complexity of testing the satisfiability of extended knowledge bases, the ExpTime OBDD construction step dominates the subsequent disjunctive Datalog reasoning part, so the overall combined complexity of the algorithm is ExpTime resulting in worst-case optimality for this case as well, given the ExpTime-hardness of satisfiability checking in $\mathcal{SHIQb}_s$.

It is also worthwhile to briefly discuss the applicability of our method to knowledge bases featuring so-called *complex role inclusion axioms* (RIAs). By means of techniques described by Kazakov (2008), any (pure, that is, non-extended) $\mathcal{SRIQb}_s$ knowledge base can be transformed into an equisatsfiable $\mathcal{ALCHIQb}$ knowledge base, however, like Motik's original transitivity elimination,

this transformation does not preserve all ground consequences. Consequently, it is not satisfiability-preserving for extended $\mathcal{SRIQ}b_s$ knowledge bases. Still, capitalizing on these RIA-removal techniques, our method provides a way for satisfiability checking for $\mathcal{SRIQ}b_s$ knowledge bases without DL-safe rules that is worst-case optimal w.r.t both combined and data complexity. We believe, however, that it would be not to hard a task to modify the transformation to even preserve ground consequences.

For future work, the algorithm needs to be evaluated in practice. A crude prototype implementation was used to generate the examples within this paper, and has shown to outperform tableaux reasoners in certain handcrafted cases, but more extensive evaluations with an optimized implementation on real-world ontologies are needed for a conclusive statement on the practical potential of this new reasoning strategy. It is also evident that redundancy elimination techniques are required to reduce the number of generated Datalog rules, which is also an important aspect of the KAON2 implementation.

Another avenue for future research is the extension of the approach to more modeling features such as role chain axioms and nominals – significant revisions of the model-theoretic considerations are needed for these cases.

REFERENCES

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., editors (2007). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition.

Blackburn, P., de Rijke, M., and Venema, Y. (2001). *Modal Logic*. Cambridge University Press.

Borgida, A. (1996). On the relative expressiveness of description logics and predicate logics. *Artif. Intell.*, 82(1–2):353–367.

Burch, J., Clarke, E., McMillan, K., Dill, D., and Hwang, L. (1990). Symbolic model checking: 10^{20} states and beyond. In *Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science*, pages 1–33. IEEE Computer Society Press.

Calvanese, D., DeGiacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2007a). Tractable reasoning and efficient query answering in description logics: The DL-lite family. *J. Autom. Reasoning*, 39(3):385–429.

Calvanese, D., DeGiacomo, G., and Rosati, R. (1998). A note on encoding inverse roles and functional restrictions in $\mathcal{ALC}$ knowledge bases. In *Proceedings of the 11th International Workshop on Description Logic (DL'98)*, pages 69–71. CEUR.

Calvanese, D., Eiter, T., and Ortiz, M. (2007b). Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI'07)*, pages 391–396. AAAI Press.

Dantsin, E., Eiter, T., Gottlob, G., and Voronkov, A. (2001). Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425.

DeGiacomo, G. and Lenzerini, M. (1994). Boosting the correspondence between description logics and propositional dynamic logics. In *In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94)*, pages 205–212. AAAI Press.

Eiter, T., Gottlob, G., and Mannila, H. (1997). Disjunctive datalog. *ACM Trans. Database Syst.*, 22(3):364–418.

Eiter, T., Lutz, C., Ortiz, M., and Šimkus, M. (2009). Query answering in description logics: The knots approach. In Ono, H., Kanazawa, M., and de Queiroz, R. J. G. B., editors, *Proceedings of the 16th International Workshop on Logic, Language, Information and Computation (WoL-LIC'09)*, volume 5514 of *Lecture Notes in Computer Science*, pages 26–36. Springer.

Fitting, M. (1996). *First-Order Logic and Automated Theorem Proving*. Springer, 2nd edition.

Glimm, B., Horrocks, I., Lutz, C., and Sattler, U. (2007). Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In Veloso, M. M., editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 399–404.

Grädel, E., Otto, M., and Rosen, E. (1997). Two-variable logic with counting is decidable. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (LICS'97)*, pages 306–317. IEEE Computer Society.

Hitzler, P., Krötzsch, M., and Rudolph, S. (2009). *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Hustadt, U., Motik, B., and Sattler, U. (2007). Reasoning in description logics by a reduction to disjunctive datalog. *J. Autom. Reasoning*, 39(3):351–384.

Hustadt, U., Motik, B., and Sattler, U. (2008). Deciding expressive description logics in the framework of resolution. *Inf. Comput.*, 206(5):579–601.

Hustadt, U. and Schmidt, R. A. (2000). Issues of decidability for description logics in the framework of resolution. In Caferra, R. and Salzer, G., editors, *Automated Deduction in Classical and Non-Classical Logics, Selected Papers*, volume 1761 of *Lecture Notes in Computer Science*, pages 191–205. Springer.

Huth, M. R. A. and Ryan, M. D. (2000). *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge University Press.

Kazakov, Y. (2008). $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In Brewka, G. and Lang, J., editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 274–284. AAAI Press.

Kazakov, Y. (2009). Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In Boutilier, C., editor, *Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI'09)*, pages 2040–2045. IJCAI.

Kazakov, Y., Krötzsch, M., and Simančík, F. (2011). Concurrent classification of $\mathcal{EL}$ ontologies. In Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., and Blomqvist, E., editors, *Proceedings of the 10th International Semantic Web Conference (ISWC'11)*, volume 7032 of *LNCS*. Springer.

Kontchakov, R., Lutz, C., Toman, D., Wolter, F., and Zakharyaschev, M. (2010). The combined approach to query answering in DL-lite. In Lin, F., Sattler, U., and Truszczynski, M., editors, *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning (KR'10)*. AAAI Press.

Krötzsch, M., Rudolph, S., and Hitzler, P. (2012). Complexities of Horn description logics. *ACM Trans. Comp. Log.* To appear; preprint available at `http://tocl.acm.org/accepted.html`.

Lutz, C. and Sattler, U. (2001). The complexity of reasoning with boolean modal logics. In Wolter, F., Wansing, H., de Rijke, M., and Zakharyaschev, M., editors, *Advances in Modal Logics Volume 3*. CSLI Publications, Stanford.

Lutz, C., Sattler, U., and Tendera, L. (2005). The complexity of finite model reasoning in description logics. *Information and Computation*, 199:132–171.

Lutz, C. and Walther, D. (2005). PDL with negation of atomic programs. *Journal of Applied Non-Classical Logics*, 15(2):189–213.

Motik, B. (2006). *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Universität Karlsruhe (TH), Germany.

Motik, B. and Sattler, U. (2006). A comparison of reasoning techniques for querying large description logic ABoxes. In Hermann, M. and Voronkov, A., editors, *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'06)*, volume 4246 of *Lecture Notes in Computer Science*, pages 227–241. Springer.

Motik, B., Sattler, U., and Studer, R. (2005). Query answering for OWL-DL with rules. *Journal of Web Semantics*, 3(1):41–60.

Motik, B., Shearer, R., and Horrocks, I. (2009). Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228.

Pan, G., Sattler, U., and Vardi, M. Y. (2006). BDD-based decision procedures for the modal logic K. *Journal of Applied Non-Classical Logics*, 16(1-2):169–208.

Popkorn, S. (1994). *First steps in modal logic*. Cambridge University Press.

Pratt, V. R. (1979). Models of program logics. In *20th Annual Symposium on Foundations of Computer Science*, pages 115–122. IEEE.

Pratt-Hartmann, I. (2005). Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14:369–395.

Rudolph, S. (2011). Foundations of description logics. In Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., and Patel-Schneider, P. F., editors, *Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Tutorial Lectures*, volume 6848 of *Lecture Notes in Computer Science*, pages 76–136. Springer.

Rudolph, S., Krötzsch, M., and Hitzler, P. (2008a). Cheap Boolean role constructors for description logics. In Hölldobler, S., Lutz, C., and Wansing, H., editors, *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08)*, volume 5293 of *Lecture Notes in Computer Science*, pages 362–374. Springer.

Rudolph, S., Krötzsch, M., and Hitzler, P. (2008b). Description logic reasoning with decision diagrams: Compiling $\mathcal{SHIQ}$ to disjunctive datalog. In Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., and Thirunarayan, K., editors, *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*, volume 5318 of *Lecture Notes in Computer Science*, pages 435–450. Springer.

Rudolph, S., Krötzsch, M., and Hitzler, P. (2008c). Terminological reasoning in $\mathcal{SHIQ}$ with ordered binary decision diagrams. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI 2008)*, pages 529–534. AAAI Press.

Schild, K. (1991). A correspondence theory for terminological logics: Preliminary report. In Mylopoulos, J. and Reiter, R., editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 466–471. Morgan Kaufmann.

Schmidt, R. A. and Tishkovsky, D. (2007). Using tableau to decide expressive description logics with role negation. In Aberer, K., Choi, K.-S., Noy, N. F., Allemang, D., Lee, K.-I., Nixon, L. J. B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web*

Conference (ISWC'07 + ASWC'07), volume 4825 of *Lecture Notes in Computer Science*, pages 438–451. Springer.

Simančík, F., Kazakov, Y., and Horrocks, I. (2011). Consequence-based reasoning beyond Horn ontologies. In Walsh, T., editor, *Proceedings of the 22nd International Conference on Artificial Intelligence (IJCAI'11)*, pages 1093–1098. AAAI Press/IJCAI.

Tobies, S. (2001). *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany.

W3C OWL Working Group (27 October 2009). *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation. Available at `http://www.w3.org/TR/owl2-overview/`.

Wegener, I. (2004). BDDs–design, analysis, complexity, and applications. *Discrete Applied Mathematics*, 138(1-2):229–251.