The Complexity of Finite Model Reasoning in Description Logics [1]

Carsten Lutz [a] Ulrike Sattler [b] Lidia Tendera [c]

[a] *Institute for Theoretical Computer Science, TU Dresden, Germany*
lutz@tcs.inf.tu-dresden.de

[b] *Department of Computer Science, University of Manchester, UK*
sattler@cs.man.ac.uk

[c] *Institute of Mathematics and Informatics, Opole University, Poland*
tendera@math.uni.opole.pl

Abstract

We analyse the complexity of finite model reasoning in the description logic $\mathcal{ALCQI}$, i.e. $\mathcal{ALC}$ augmented with qualifying number restrictions, inverse roles, and general TBoxes. It turns out that all relevant reasoning tasks such as concept satisfiability and ABox consistency are ExpTime-complete, regardless of whether the numbers in number restrictions are coded unarily or binarily. Thus, finite model reasoning with $\mathcal{ALCQI}$ is not harder than standard reasoning with $\mathcal{ALCQI}$.

Key words:
description logic, finite satisfiability, number restrictions
PACS: 03B70, 03D15, 68Q17

1 Motivation

Description logics (DLs) are a family of logical formalisms that originated in the field of knowledge representation, and that were designed to represent and reason about conceptual knowledge. Central DL notions are *concepts* (unary predicates or classes) and *roles* (binary relations). A specific DL is mainly characterized by the constructors it provides to build complex concepts (and

[1] This is an extended version of the conference paper [1]. This work was performed while the third author was visiting the Instititute for Theoretical Computer Science at TU Dresden supported by DFG grant BA1122/3-2

roles) from atomic ones. For example, in the basic DL $\mathcal{ALC}$ [2], all roles are atomic, and concepts can be built using Boolean operators and value restrictions. The following $\mathcal{ALC}$-concept describes companies in which only managers or researchers work, and in which a parent works.

$$\texttt{Company} \sqcap (\exists \texttt{employs}.\exists \texttt{hasChild}.\texttt{Human}) \sqcap \forall \texttt{employs}.(\texttt{Researcher} \sqcup \texttt{Manager})$$

It is well-known that DLs are closely related to modal logics. For example, $\mathcal{ALC}$ is a notational variant of the basic multi-modal logic K [3], and the above $\mathcal{ALC}$ concept is the DL counterpart of the multi modal formula

$$\texttt{Company} \wedge (\langle \texttt{employs} \rangle \langle \texttt{hasChild} \rangle \texttt{Human}) \wedge [\texttt{employs}](\texttt{Researcher} \vee \texttt{Manager}).$$

A standard DL knowledge base, called TBox, consists of a set of concept equations, i.e. expressions of the form $C \doteq D$ where C and D are possibly complex concepts. Intuitively, a TBox constrains the set of models that are admitted for the interpretation of concepts. Using a TBox, we can thus describe the *terminology* of an application domain by using an (atomic) concept name on the left-hand side and its (complex) definition on the right-hand side. Moreover, we can capture general constraints that come from the application domain. The standard DL reasoning tasks are deciding concept satisfiability and concept subsumption w.r.t. a TBox: checking whether a concept C can have any instances in models of the TBox $\mathcal{T}$, and checking whether one concept D is more general than another concept C w.r.t. models of $\mathcal{T}$.

During the last decade, a lot of work has been devoted to investigating the classical trade-off between expressivity and complexity [4], i.e., to find DLs whose expressive power is appropriate for a certain kind of applications, and whose reasoning problems are still decidable, preferably of an acceptable complexity.

Applications for which such a good compromise could be found include reasoning about conceptual database models [5] and the usage of DLs as logical underpinning of ontology languages such as DAML+OIL and OWL [6; 7]. In this paper, we are concerned with the former application. Suppose that a conceptual database model is described by one of the standard formalisms: an ER diagram in the case of relational databases and a UML diagram in the case of object-oriented databases. As shown in [5], such models can be translated into a DL TBox and a description logic reasoner such as FaCT or RACER [8; 9] can be used to reason about the database model. In particular, this approach can be used to detect inconsistencies in the database model, and to infer implicit IS-A relationships between entities/classes that are not given explicitly in the model. This useful and original application has already led to the implementation of tools that provide a GUI for specifying conceptual models, automatise the translation into description logics, and display the information returned by the DL reasoner [10].

One of the most important description logics used for reasoning about conceptual database models is called $\mathcal{ALCQI}$ [11], and extends $\mathcal{ALC}$ with

- qualifying number restrictions (corresponding to graded modalities in modal logic): concepts of the form $(\geqslant n R.C)$ and $(\leqslant n R.C)$, describing objects having at least n (at most n) instances of C related to them via the role R. For example, the concept $\mathtt{Company} \sqcap (\leqslant 3 \mathtt{employs}.\mathtt{Manager})$ describes companies employing at most 3 managers.
- the inverse role constructor (corresponding to inverse modalities): $\mathcal{ALCQI}$ allows the use of the inverse R^- of a role R in number restrictions and value restrictions. For example, the concept $\mathtt{Manager} \sqcap (\geqslant 2 \mathtt{employs}^-.\mathtt{Company})$ describes managers that are employed by at least two companies.

A feature that distinguishes $\mathcal{ALCQI}$ from less expressive DLs is that $\mathcal{ALCQI}$ is capable of enforcing infinity, i.e., there are concepts and TBoxes that are satisfiable, but admit only infinite models. In other words, $\mathcal{ALCQI}$ lacks the *finite model property* (FMP).

Since reasoning about database models is one of $\mathcal{ALCQI}$'s premier applications, its lack of the FMP cannot be ignored: database models are usually encoded into $\mathcal{ALCQI}$ such that there is a tight correspondence between logical models and databases; since databases are usually considered to be finite, we should thus perform reasoning on *finite* models rather than on unrestricted ones when using $\mathcal{ALCQI}$ in this context. That the restriction to finite models indeed makes a difference is witnessed by that fact that there exist quite simple ER and UML diagrams that are satisfiable only in infinite models [12]. From a database perspective, such diagrams should thus be considered *inconsistent* rather than consistent, and thus we get an incorrect result when translating them to $\mathcal{ALCQI}$ and using unrestricted model reasoning. Interestingly, the problem of finite models is commomly ignored when using DL tools for reasoning about database models. This is due to the fact that, with FaCT and RACER [8; 9], there are two popular and highly efficient reasoners for dealing with unrestricted reasoning in $\mathcal{ALCQI}$ but, up to now, no $\mathcal{ALCQI}$ reasoner for finite models is available. We believe that one important reason for the lack of finite model reasoners is that, in contrast to reasoning w.r.t. unrestricted models, reasoning w.r.t. finite models in $\mathcal{ALCQI}$ is not yet well understood from a theoretical perspective. In particular, as we will discuss below in more detail, tight complexity bounds for finite model reasoning in $\mathcal{ALCQI}$ have never been determined. *The purpose of this paper is thus to improve the understanding of finite model reasoning in description logics by establishing tight* ExpTime *complexity bounds for finite model reasoning in the DL $\mathcal{ALCQI}$.*

As noted above, reasoning with $\mathcal{ALCQI}$ in unrestricted models is well-understood. For example, it is known that satisfiability and subsumption w.r.t. TBoxes is ExpTime-complete [11]. Note that there is a subtle issue about number restrictions here: inside $\mathcal{ALCQI}$'s constructors ($\leqslant n\ R\ C$) and ($\geqslant n\ R\ C$), we can code the number n either in unary or in binary, and the length of concepts and TBoxes will clearly be exponentially shorter in the latter case. Fortunately, the $\mathcal{ALCQI}$ ExpTime-completeness results is insensitive of this coding, i.e., it holds for both cases [13].

For finite model reasoning, no tight complexity bounds were known. It follows easily from modal correspondance theory [14] that $\mathcal{ALCQI}$ is a fragment of the two variable fragment of first order logic with counting quantifiers (C2) [15; 16]. Hence finite satisfiability of C2 being decidable [15] implies that, in $\mathcal{ALCQI}$, finite satisfiability and subsumption w.r.t. TBoxes are decidable as well. Moreover, Calvanese proves in [17] that $\mathcal{ALCQI}$ satisfiability and subsumption w.r.t. TBoxes are decidable in 2-ExpTime. Very recently, finite satisfiability of C2 was proven to be complete for non-deterministice exponential time [18; 19], which improves Calvanese's upper bound. A lower bound follows easily from the fact that reasoning in $\mathcal{ALC}$ is already ExpTime-hard [20; 3]—both w.r.t. unrestricted and finite models since $\mathcal{ALC}$ enjoys the finite model property. This leaves us with a gap between ExpTime and NExptime for finite model reasoning in $\mathcal{ALCQI}$ and the question whether it is as insensitive to the coding of numbers as unrestricted model reasoning: all upper bounds mentioned were proved for unary coding of numbers. In this paper, we will close this gap by providing a tight ExpTime upper bound and show that, similar to the unrestricted case, the complexity is insensitive to the coding of numbers. More precisely, we present the following results:

In Section 3, we develop an algorithm that decides the finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes. Similar to Calvanese's approach, the core idea behind our algorithm is to translate a given satisfiability problem into a set of linear inequalities that can then be solved by linear programming methods. In this translation, we use variables to represent the number of elements described by so-called *mosaics*: a mosaic is an abstraction of domain elements which describes the (unary) type of a domain element together with its "neighborhood", i.e., the numbers and types of (relevant) role successors. Using a rather strict notion of mosaics and an appropriate data structure to represent them allows us to keep the number of mosaics exponential in the size of the input. This yields an exponential bound on the number of variables and also on the size of systems of inequalities. Thus, we improve the best-known 2-ExpTime upper bound to a tight ExpTime one.

However, this bound is exponential only if we assume unary coding of numbers in number restrictions, and it is not clear whether our translation can be modified to yield an ExpTime upper bound in the case of binary coding. Thus, we use a different strategy to attack binary coding: in Section 4, we give a polynomial reduction of finite $\mathcal{ALCQI}$-concept satisfiability w.r.t. TBoxes to finite satisfiability of $\mathcal{ALCFI}$-concept satisfiability w.r.t. TBoxes, where $\mathcal{ALCFI}$ is obtained from $\mathcal{ALCQI}$ by allowing only numbers up to two to be used in number restrictions. Since finite model reasoning in $\mathcal{ALCFI}$ is in ExpTime by the results from Section 3 (the coding of numbers is not an issue here), we obtain a tight ExpTime bound for finite model reasoning in $\mathcal{ALCQI}$ with numbers coded in binary. Note that we cannot use existing reductions from $\mathcal{ALCQI}$ to $\mathcal{ALCFI}$ since these fail for finite model reasoning [11].

In Section 5, we extend our result to a more general reasoning problem, namely the finite consistency of ABoxes w.r.t. TBoxes. Intuitively, ABoxes describe a particular state of affairs, a "snapshot" of the world. Finite $\mathcal{ALCQI}$-ABox consistency is another interesting reasoning task with important applications: whereas finite $\mathcal{ALCQI}$-concept satisfiability can be used to decide the consistency of conceptual database models and infer implicit IS-A relationships, $\mathcal{ALCQI}$-ABox consistency can be used as the core component of algorithms deciding containment of conjunctive queries w.r.t. conceptual database models—a task that DLs have successfully been used for and that calls for finite model reasoning [21; 22]. Using a reduction to (finite) concept satisfiability, we are able to show that this reasoning task is also ExpTime-complete, independently of the way in which numbers are coded.

Finally, in Section 6, we discuss related work.

2 Preliminaries

We introduce syntax and semantics of $\mathcal{ALCQI}$, discuss the inference problems we are interested in, and introduce some useful notation.

Definition 1 ($\mathcal{ALCQI}$ Syntax) *Let* $\mathbf{R}$ *and* $\mathbf{C}$ *be disjoint and countably infinite sets of* role *and* concept names. *A* role *is either a role name* $R \in \mathbf{R}$ *or the inverse* R^- *of a role name* $R \in \mathbf{R}$. *The set of $\mathcal{ALCQI}$-concepts is the smallest set satisfying the following properties:*

- *each concept name* $A \in \mathbf{C}$ *is an $\mathcal{ALCQI}$-concept;*
- *if* C *and* D *are $\mathcal{ALCQI}$-concepts,* R *is a role, and* n *a natural number, then* $\neg C$, $C \sqcap D$, $C \sqcup D$, $(\leqslant n\ R\ C)$, *and* $(\geqslant n\ R\ C)$ *are also $\mathcal{ALCQI}$-concepts.*

A concept equation *is of the form* $C \doteq D$ *for* C, D *two $\mathcal{ALCQI}$-concepts. A* TBox *is a finite set of concept equations.*

We will refer to concepts of the form $(\leqslant n\ R\ C)$ as *atmost restrictions* and to concepts of the form $(\geqslant n\ R\ C)$ as *atleast restrictions*. As usual, we use the standard abbreviations $\rightarrow$ and $\leftrightarrow$ as well as $\exists R.C$ for $(\geqslant 1\ R\ C)$, $\forall R.C$ for $(\leqslant 0\ R\ \neg C)$, $\top$ to denote an arbitrary propositional tautology, and $\bot$ as abbreviation for $\neg\top$. The fragment $\mathcal{ALCFI}$ of $\mathcal{ALCQI}$ is obtained by admitting only atmost restrictions $(\leqslant n\ R\ C)$ with $n \in \{0, 1\}$ and only atleast restrictions $(\geqslant n\ R\ C)$ with $n \in \{1, 2\}$.

Definition 2 ($\mathcal{ALCQI}$ Semantics) *An interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a mapping that assigns*

- *to each concept name A, a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and*
- *to each role name R, a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.*

The interpretation of inverse roles and complex concepts is then defined as follows, with $\#S$ denoting the cardinality of the set S:

$$\begin{aligned}
(R^{-})^{\mathcal{I}} &= \{\langle e, d\rangle \mid \langle d, e\rangle \in R^{\mathcal{I}}\} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\leqslant n\ R\ C)^{\mathcal{I}} &= \{d \mid \#\{e \in C^{\mathcal{I}} \mid \langle d, e\rangle \in R^{\mathcal{I}}\} \leq n\} \\
(\geqslant n\ R\ C)^{\mathcal{I}} &= \{d \mid \#\{e \in C^{\mathcal{I}} \mid \langle d, e\rangle \in R^{\mathcal{I}}\} \geq n\}
\end{aligned}$$

A domain element $d \in \Delta^{\mathcal{I}}$ is an instance *of a concept C if $d \in C^{\mathcal{I}}$; moreover, a domain element $d' \in \Delta^{\mathcal{I}}$ is an R-*neighbour *of d, for R a role, if $(d, d') \in R^{\mathcal{I}}$.*

An interpretation $\mathcal{I}$ satisfies *a concept equation $C \doteq D$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$, and $\mathcal{I}$ is called a* model *of a TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all concept equations in $\mathcal{T}$.*

A concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ *if there is a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$. A concept C is* finitely satisfiable w.r.t. a TBox $\mathcal{T}$ *if there is a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$ and $\Delta^{\mathcal{I}}$ finite.*

To see that satisfiability and finite satisfiability do not coincide, consider the concept $C = \neg A \sqcap \exists R.A$ and the TBox $\{A \doteq \exists R.A \sqcap (\leqslant 1\ R^{-}\ \top)\}$. It is not hard to see that C is satisfiable w.r.t. $\mathcal{T}$, but only in infinite models: each model contains an infinite, acyclic R-chain. Thus, $\mathcal{ALCQI}$ does not enjoy the finite model property.

The second important reasoning problem on concepts and TBoxes, subsumption of concepts w.r.t. TBoxes, has already been mentioned in the introduction: a concept C is *(finitely) subsumed* by a concept D w.r.t. a TBox $\mathcal{T}$ if we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each (finite) model $\mathcal{I}$ of $\mathcal{T}$. It is well known that subsumption can be reduced to (un)satisfiability, as C is subsumed by D w.r.t. $\mathcal{T}$ if and only if

$$\neg(C \sqcap D) \rightsquigarrow \neg C \sqcup \neg D \qquad \neg(C \sqcup D) \rightsquigarrow \neg C \sqcap \neg D$$

$$\neg\neg C \rightsquigarrow C \qquad \neg(\leqslant n\ R\ C) \rightsquigarrow (\geqslant n+1\ R\ C)$$

$$\neg(\geqslant n\ R\ C)) \rightsquigarrow (\leqslant n-1\ R\ C) \text{ if } n > 0$$

$$\neg(\geqslant n\ R\ C)) \rightsquigarrow \bot \qquad \text{if } n = 0$$

Fig. 1. The NNF rewrite rules.

$C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$. Since this holds both for the infinite and the finite case, in this paper we will concentrate on satisfiability and just note here that all complexity bounds obtained in this paper also apply to subsumption (despite the implicit complementation in the reduction, since we will only be dealing with deterministic complexity classes).

In the remainder of this paper, we will w.l.o.g. only consider concepts and TBoxes that are in a restricted syntactic form: concepts are assumed to be in *negation normal form (NNF)*, i.e., negation is only allowed in front of concept names. Every $\mathcal{ALCQI}$-concept can be transformed in linear time into an equivalent one in NNF by exhaustively applying the rewrite rules displayed in Figure 1. We use $\dot\neg C$ to denote the NNF of $\neg C$. TBoxes are assumed to be of the rather simple form $\{\top \doteq C\}$ with C in NNF. This can be done w.l.o.g. since an interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T} = \{C_i \doteq D_i \mid 1 \leq i \leq n\}$ iff it is a model of $\{\top \doteq \bigsqcap_{1 \leq i \leq n}(C_i \leftrightarrow D_i)\}$.

We now introduce some convenient notation used throughout this paper. For each role R, we use $\mathsf{Inv}(R)$ to denote R^- if R is a role name, and S if $R = S^-$. For a given concept C and TBox $\mathcal{T}$, we use $\mathsf{cnam}(C, \mathcal{T})$ to denote the set of concept names appearing in C and $\mathcal{T}$, $\mathsf{rnam}(C, \mathcal{T})$ to denote the set of role names appearing in C and $\mathcal{T}$, and $\mathsf{rol}(C, \mathcal{T})$ to denote the set

$$\mathsf{rnam}(C, \mathcal{T}) \cup \{R^- \mid R \in \mathsf{rnam}(C, \mathcal{T})\}.$$

3 Unary Coding of Numbers

In this section, we present a decision procedure for finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes that runs in deterministic exponential time, provided that numbers in number restrictions are coded unarily. In Section 4, we will generalise this upper bound to binary coding of numbers.

It is easily seen that combinatorics is an important issue when deciding finite satisfiability of $\mathcal{ALCQI}$-concepts. To illustrate this, consider the TBox

$$\mathcal{T} := \{A \doteq (\geqslant 2\ R\ B),\ B \doteq (\leqslant 1\ R^-\ A)\}. \qquad (*)$$

In any (finite) model of $\mathcal{T}$, there are at least twice as many objects satisfying B as there are objects satisfying A. This kind of combinatorics is not an issue if infinite domains are admitted: in this case, we can always find a model where all concepts have the same number of instances, namely countably infinitely many.

As observed by Calvanese in [17], the combinatorial issues of finite model reasoning in description logics can be addressed by using systems of inequalities. More precisely, for deciding the finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes, we will convert a given conept C_0 and TBox $\mathcal{T}$ into a system of linear inequalities that describes the induced combinatorial constraints. This is done in a such way that there is a correspondence between non-negative integer solutions of the equation system and finite models of the input. In this way, checking finite satisfiability of the input concept and TBox corresponds to checking whether the constructed system of inequalities has a non-negative integer solution. To obtain an ExpTime upper bound as desired, we have to be careful to ensure that the system of inequalities can be constructed in time exponential in the size of the input, and that the existence of solutions can be checked in polynomial time.

Equation systems that handle combinatorial constraints can be conveniently formulated in terms of types, which we introduce next. Along with types, we define the closure of an $\mathcal{ALCQI}$-concept C_0 and a TBox $\mathcal{T}$, which is, intuitively, the set of concepts that are "relevant" for deciding the (finite) satisfiability of C_0 w.r.t. $\mathcal{T}$.

Definition 3 (Closure, Type) *Let C_0 be a concept and $\mathcal{T} = \{\top \doteq C_\mathcal{T}\}$ a TBox. The* closure $\mathsf{cl}(C_0, \mathcal{T})$ *of C_0 and $\mathcal{T}$ is the smallest set of $\mathcal{ALCQI}$-concepts such that*

- *C_0, $C_\mathcal{T}$, and all sub-concepts of C_0 and $C_\mathcal{T}$ are in $\mathsf{cl}(C_0, \mathcal{T})$;*
- *if $C \in \mathsf{cl}(C_0, \mathcal{T})$, then $\dot\neg C$, the NNF of $\neg C$, is also in $\mathsf{cl}(C_0, \mathcal{T})$.*

A type T for C_0 and $\mathcal{T}$ *is a subset $T \subseteq \mathsf{cl}(C_0, \mathcal{T})$ such that, for all $D, E \in \mathsf{cl}(C_0, \mathcal{T})$, we have*

(1) $D \in T$ iff $\dot\neg D \notin T$,
(2) if $D \sqcap E \in \mathsf{cl}(C_0, \mathcal{T})$, then $D \sqcap E \in T$ iff $D \in T$ and $E \in T$,
(3) if $D \sqcup E \in \mathsf{cl}(C_0, \mathcal{T})$, then $D \sqcup E \in T$ iff $D \in T$ or $E \in T$, and
(4) $C_\mathcal{T} \in T$.

We use $\mathsf{type}(C_0, \mathcal{T})$ to denote the set of all types for C_0 and $\mathcal{T}$.

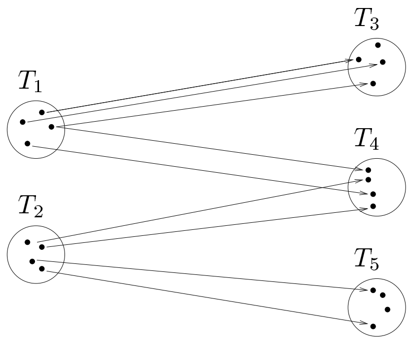

Fig. 2. Problems with types.

For interpretations $\mathcal{I}$, we call a domain element $d \in \Delta^{\mathcal{I}}$ an instance *of a type T if $d \in C^{\mathcal{I}}$ for all $C \in T$. Moreover, we use $t(d)$ to denote the type that d is an instance of.*[2]

A first idea to convert a finite satisfiability problem into an equational problem could be to introduce one variable x_T for each type T for the input concept C_0 and TBox $\mathcal{T}$, and then to formulate a suitable system of inequalities for C_0 and $\mathcal{T}$ such that each non-negative integer solution δ of the equation system corresponds to a model where each type T has exactly $\delta(x_T)$ instances.

However, it turns out that this approach is too naive: assume that T_1 to T_5 are types for C_0 and $\mathcal{T}$, and that the following holds:

- $(\geqslant 1\ R\ C) \in T_1$ and $(\geqslant 1\ R\ D) \in T_2$,
- $(\leqslant 1\ R^{-}\ \top) \in T_3 \cap T_4 \cap T_5$,
- $C \in T_3 \cap T_4$ and $D \in T_4 \cap T_5$.

Observe that (instances of) T_1 can "use" (instances of) T_3 and T_4 to satisfy the concept $(\geqslant 1\ R\ C) \in T_1$, and T_2 can "use" T_4 and T_5 to satisfy the concept $(\geqslant 1\ R\ D) \in T_2$, a situation depicted in Figure 2. Similarly as for our initial example $(*)$, we get that (i) there have to be at least as many instances of T_3 and T_4 as there are instances of T_1, and (ii) there have to be at least as many instances of T_4 and T_5 as there are instances of T_2. Thus, it is likely that a system of inequalities for C_0 and $\mathcal{T}$ will include

$$x_{T_1} \leq x_{T_3} + x_{T_4} \text{ and } x_{T_2} \leq x_{T_4} + x_{T_5}. \qquad (**)$$

Ignoring the existence of possible additional inequalities for a second, we obtain $x_{T_1} = x_{T_2} = x_{T_4} = 1$ and $x_{T_3} = x_{T_5} = 0$ as an integer solution. Trying to construct a model with a_1, a_2, and a_4 instances of T_1, T_2, and T_4, respectively, we have to use a_4 as a witness of a_1 being an instance of $(\geqslant 1\ R\ C)$ and a_2 being an instance of $(\geqslant 1\ R\ D)$. Since this clearly violates the $(\leqslant 1\ R^{-}\ \top)$

[2] This type is obviously unique, and thus $t(d)$ well defined.

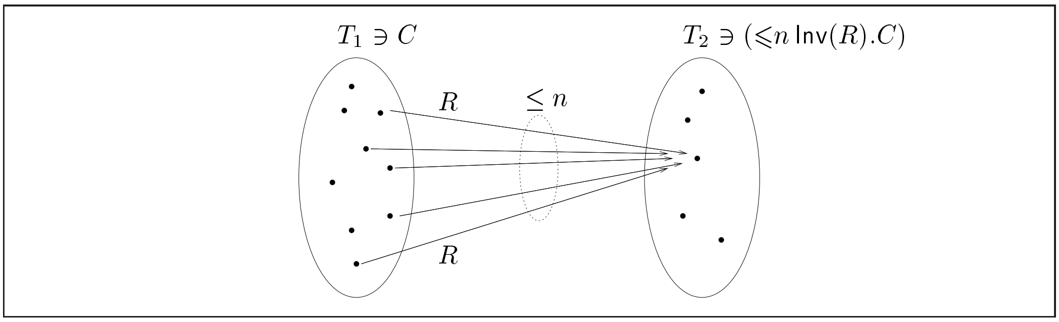

Fig. 3. Illustration of the lim function.

concept in T_4, we do not have an easy correspondence between models and integer solutions as sketched above. Intuitively, the problem is that, above, we have considered Points (i) and (ii) separately although they both speak about T_4. Unfortunately, it seems impossible to resolve this problem by adding additional inequalities of size at most exponential in the size of the input.

One possible view on the sketched problem, which is also taken by Calvanese in [17], is that types do not provide enough information about domain elements. Intuitively, it seems necessary to also record, for each role R, the type and number of R-neighbours. If this is done, in the above example ($**$), we can distinguish instances of T_1 and T_2 that have R-neighbors of type T_4 from those that do not. It is then possible to refine the given equations such that "infeasible solutions" such as the one discussed are ruled out. Thus, we now develop a refinement of types that allows to describe such additional information. We start with introducing a convenient notation that will play a rather prominent role throughout this paper.

Definition 4 (lim **function)** *Let C_0 be a concept, $\mathcal{T}$ a TBox, R a role, and T_1, T_2 types for C_0 and $\mathcal{T}$. Then we write*

$$\lim_R(T_1, T_2)$$

if $C \in T_1$ and $(\leqslant n\ \mathsf{Inv}(R)\ C) \in T_2$ for some $C \in \mathsf{cl}(C_0, \mathcal{T})$ and $n \in \mathbb{N}$.

Intuitively, $\lim_R(T_1, T_2)$ holds if, for each instance of T_2, there can be only a limited number of "incoming R-edges" from instances of T_1. This situation is illustrated in Figure 3, where the left ellipse contains all instances of type T_1 and the right ellipsis contains all instances of type T_2. Note that, in the initial example ($*$), we have $\lim_R(T_1, T_2)$ for all types T_1, T_2 such that T_1 contains A and T_2 contains B.

Our generalization of a type to also include the type and number of R-neighbours is called a mosaic, and is defined as follows.

Definition 5 (Mosaic) *Let T be a type and $\bowtie \in \{\leqslant, \geqslant\}$. Then we use the* *following abbreviations:*

$$\mathsf{max}^{\bowtie}(T) := \max\{n \mid (\bowtie n\ R\ C) \in T\}$$

$$\mathsf{sum}^{\bowtie}(T) := \sum_{(\bowtie n\ R\ C) \in T} n.$$

A mosaic *for a concept C_0 and a TBox $\mathcal{T}$ is a triple $M = (T_M, L_M, E_M)$ where*

- $T_M \in \mathsf{type}(C_0, \mathcal{T})$,
- L_M *and* E_M *are functions from* $\mathsf{rol}(C_0, \mathcal{T}) \times \mathsf{type}(C_0, \mathcal{T})$ *to* $\mathbb{N}$.

such that the following conditions are satisfied:

(M1) *if* $L_M(R, T) > 0$, *then* $\lim_R(T_M, T)$ *and not* $\lim_{\mathsf{Inv}(R)}(T, T_M)$,
(M2) *if* $E_M(R, T) > 0$, *then* $\lim_{\mathsf{Inv}(R)}(T, T_M)$,
(M3) *if* $(\leqslant n\ R\ C) \in T_M$, *then* $n \geq \sum_{\{T \mid C \in T\}} E_M(R, T)$,
(M4) $\#\{(R, T) \mid L_M(R, T) > 0\} \leq \mathsf{sum}^{\geqslant}(T_M)$ *and* $\max(\mathsf{ran}(L_M)) \leq \mathsf{max}^{\geqslant}(T_M)$, *where* $\mathsf{ran}(f)$ *denotes the range of the function* f.

If $\mathcal{I}$ is an interpretation, $d \in \Delta^{\mathcal{I}}$, and $M = (T_M, L_M, E_M)$ a mosaic for C_0 and $\mathcal{T}$, then d is an instance *of M if the following holds, for all $R \in \mathsf{rol}(C_0, \mathcal{T})$ and $T \in \mathsf{type}(C_0, \mathcal{T})$:*

- $t(d) = T_M$, *i.e. d is an instance of T_M;*
- *if* $\lim_R(T_M, T)$ *and not* $\lim_{\mathsf{Inv}(R)}(T, T_M)$, *then* $L_M(R, T)$ *is the minimum of* $\mathsf{max}^{\geqslant}(T_M)$ *and* $\#\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in R^{\mathcal{I}}$ *and* $t(e) = T\}$*;*
- *if* $\lim_{\mathsf{Inv}(R)}(T, T_M)$, *then* $E_M(R, T) = \#\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in R^{\mathcal{I}}$ *and* $t(e) = T\}$.

It follows immediately from this definition that each domain element d is an instance of exactly one mosaic. The definition of "instance" shows how mosaics are used to describe domain elements: while T_M is simply the type of d in $\mathcal{I}$, L_M and E_M are used to describe the number of neighbours of d of certain types that are reachable from d via some role R, up to the limit $\mathsf{max}^{\geqslant}(T_M)$ in the L_M case (to keep the number of mosaics "small"). More precisely, we distinguish three possibilities for the R relationship between T_M and a type T:

(1) $\lim_R(T_M, T)$ and not $\lim_{\mathsf{Inv}(R)}(T, T_M)$. Then each instance of T_M may have an unrestricted number of R-neighbours of type T since, by definition of lim, $(\leqslant n\ R\ C) \in T_M$ implies $C \notin T$. However, each instance of T has a limit on the number of $\mathsf{Inv}(R)$-neighbours of type T_M: there is some $(\leqslant n\ \mathsf{Inv}(R)\ C) \in T$ with $C \in T_M$. Thus, we must be careful not to violate this limit when using instances of T as "witnesses" to satisfy atleast restrictions $(\geqslant n\ R\ D) \in T_M$ with $D \in T$ (such a violation is exactly what is happening in the example $(**)$ above). To this end, we record in

L_M the minimal number of R-neighbours of type T that an instance of M has ("L" for "lower bound"). In the equation systems to be defined later, this lower bound will be used to take care of atleast restrictions in T_M.

(2) $\lim_{\mathsf{Inv}(R)}(T, T_M)$. Then an instance d of T_M may only have a limited number of R-neighbours of type T. To prevent the violation of this limit, we need to record an upper bound on the number of d's R-neighbours of type T in M. On the other hand, there may be atleast restrictions in T_M that need witnesses of type T. Thus, we also want to record a lower bound on the number of d's R-neighbours of type T in M. Summing up, we use E_M to record the *exact* number of d's R-neighbours of type T ("E" for "exact bound").

(3) Not $\lim_R(T_M, T)$ and not $\lim_{\mathsf{Inv}(R)}(T, T_M)$. Then each instance of T_M may have an unrestricted number of R-neighbours of type T and each instance of T may have an unrestricted number of $\mathsf{Inv}(R)$-neighbours of type T_M. Intuitively, R-neighbours of type T are "uncritical" for M and thus their number need not be recorded in the mosaic (we shall see later that even without stating a lower bound, it is easy to satisfy atleast restrictions in T_M using witnesses in T).

The conditions (M1) to (M4) of mosaics can thus be understood as follows: (M1) and (M2) ensure that L_M and E_M record information for the "correct" types as described above; (M3) ensures that atmost restrictions are not violated—it suffices to consider only E_M here since $(\leqslant n\ R\ C) \in T_M$ and $C \in T$ implies $L_M(R, T) = 0$ by (M1) and definition of lim; finally, (M4) puts upper bounds on L_M to ensure that there exists only a limited number of mosaics.

To use mosaics in systems of inequalities, we introduce one variable x_M for each mosaic M for the input C_0 and $\mathcal{T}$, instead of for each type as sketched before. The intuition behind variables, however, is slightly different from the type-based case: the goal is to ensure that each non-negative integer solution δ of the equation system corresponds to a *pre*-model in which each mosaic M has exactly $\delta(x_M)$ instances. Intuitively, pre-models differ from models in that, for any role R and domain elements d, e, they admit *multiple R-edges* between d and e.

Definition 6 (Pre-model) *A* pre-interpretation $\mathcal{I}$ *is a pair* $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$ *where* $\Delta^\mathcal{I}$ *is a non-empty set and* $\cdot^\mathcal{I}$ *is a mapping that assigns*

- *to each concept name A, a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and*
- *to each role name R, a function $R^\mathcal{I} : (\Delta^\mathcal{I} \times \Delta^\mathcal{I}) \to \mathbb{N}$.*

Complex concepts and roles are interpreted as for standard interpretations,

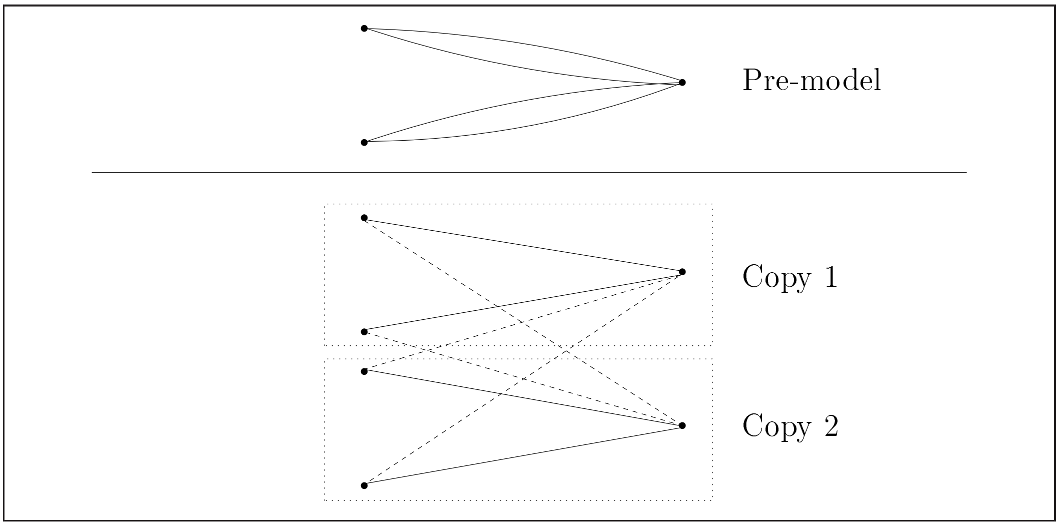

Fig. 4. The copying construction.

with the following exceptions:

$$(R^-)^\mathcal{I}(d,e) = R^\mathcal{I}(e,d),$$
$$(\leqslant n\ R\ C)^\mathcal{I} = \{d \mid \textstyle\sum_{e \in C^\mathcal{I}} R^\mathcal{I}(d,e) \leq n\}, \text{ and}$$
$$(\geqslant n\ R\ C)^\mathcal{I} = \{d \mid \textstyle\sum_{e \in C^\mathcal{I}} R^\mathcal{I}(d,e) \geq n\}.$$

A pre-interpretation $\mathcal{I}$ *is a* pre-model *of a concept* C_0 *and a TBox* $\mathcal{T}$ *iff* $C_0^\mathcal{I} \neq \emptyset$ *and* $C \doteq D \in \mathcal{T}$ *implies* $C^\mathcal{I} = D^\mathcal{I}$.

It is straightforward to adapt the notion "instance of mosaic" to pre-models by taking into account the multiple edges when defining L_M and E_M: we only have to replace $\#\{e \in \Delta^\mathcal{I} \mid (d,e) \in R^\mathcal{I}$ and $t(e) = T\}$ with $\sum_{e \in T^\mathcal{I}} R^\mathcal{I}(d,e)$.

The following theorem shows that we may safely consider pre-models instead of models when checking satisfiability.

Theorem 7 *A concept* C_0 *and a TBox* $\mathcal{T}$ *have a finite pre-model iff* C_0 *and* $\mathcal{T}$ *have a finite (standard) model.*

The "if" direction is trivial since every standard model can be conceived as a pre-model. A formal proof of the "only if" direction can be found in Appendix A. Intuitively, to obtain a finite standard model from a finite pre-model $\mathcal{I}$ for C_0 and $\mathcal{T}$, we take a finite number of "disjoint copies" of $\mathcal{I}$, and then bend some role relationships back and forth to eliminate multiple edges. This construction is illustrated in Figure 4: if the maximum multiplicity of edges in the pre-model is n, we take n disjoint copies of it and "bend" the ith edge between two elements d and e in the jth copy to go to (the copy of) e in the $((j+i) \mod n)$th copy. This ensures that, for any role R, type T, and domain element d of the resulting model $\mathcal{I}'$, d has exactly the same number of R-neighbours of type T as its corresponding domain element in the pre-model $\mathcal{I}$. As a consequence, $\mathcal{I}'$ is still a model of C_0 and $\mathcal{T}$.

Let us now come back to the system of inequalities. As already stated, the variables represent the number of instances that mosaics have in a pre-model. We use inequalities to ensure that we can "connect" the instances of the mosaics via roles such that

(a) the lower bounds on numbers of successors stored in L_M are satisfied,
(b) the exact numbers of successors stored as $E_M(R, T)$ are satisfied, where we have to distinguish the following two cases
   (i) $\lim_{\mathsf{Inv}(R)}(T, T_M)$ and $\lim_R(T_M, T)$, and
   (ii) $\lim_{\mathsf{Inv}(R)}(T, T_M)$ and *not* $\lim_R(T_M, T)$.
(c) all atleast concepts are satisfied.

Note that we do not need to worry about the atmost-concepts as they are ensured by (M3) together with Point (b) above. We first give the inequalities and then relate them to Points (a) to (c) above.

Definition 8 (Equation System) *For C_0 an $\mathcal{ALCQI}$-concept and $\mathcal{T}$ a TBox, we introduce a variable x_M for each mosaic M for $C_0, \mathcal{T}$ and define the system of inequalities $\mathcal{E}_{C_0,\mathcal{T}}$ by taking (i) the inequality*

$$\sum_{\{M \mid C_0 \in T_M\}} x_M \geq 1, \tag{E1}$$

(ii) for each pair of types T, $T' \in \mathsf{type}(C_0, \mathcal{T})$ and role R such that $\lim_R(T, T')$ and not $\lim_{\mathsf{Inv}(R)}(T', T)$ the inequality

$$\sum_{\{M \mid T_M = T\}} L_M(R, T') \cdot x_M \leq \sum_{\{M \mid T_M = T'\}} E_M(\mathsf{Inv}(R), T) \cdot x_M, \tag{E2}$$

and (iii) for each pair of types T, $T' \in \mathsf{type}(C_0, \mathcal{T})$ and role R such that $\lim_R(T, T')$ and $\lim_{\mathsf{Inv}(R)}(T', T)$ the inequality

$$\sum_{\{M \mid T_M = T\}} E_M(R, T') \cdot x_M = \sum_{\{M \mid T_M = T'\}} E_M(\mathsf{Inv}(R), T) \cdot x_M. \tag{E3}$$

We give a brief overview of the purpose of the inequalities, and refer to the proof of Lemma 10 below for the full picture. Inequality (E1) simply guarantees the existence of an instance of C_0, and inequality (E2) deals with Point (a) from above. Point (b) is comprised of two subcases, and Point (b.i) is dealt with by inequality (E3). In contrast, Point (b.ii) and (c) cannot be dealt with by a simple inequality since they rather require a "conditional" inequality. To address these two points, we introduce the notion of admissible solutions.

Definition 9 (Admissible) *A solution of* $\mathcal{E}_{C_0,\mathcal{T}}$ *is* admissible *if it is a non-negative integer solution and satisfies the following side-conditions:*

(i) for each pair of types $T,\ T' \in \mathsf{type}(C_0, \mathcal{T})$ *and role* R *such that* $\lim_R(T, T')$ *and not* $\lim_{\mathsf{Inv}(R)}(T', T)$*,*

$$\text{if} \quad \sum_{\{M \mid T_M = T'\}} E_M(\mathsf{Inv}(R), T) \cdot x_M > 0, \quad \text{then} \quad \sum_{\{M \mid T_M = T\}} x_M > 0. \tag{A1}$$

(ii) for each mosaic M *and each role* R*, if* $x_M > 0$*,* $(\geqslant n\ R\ C) \in T_M$*, and*

$$m = \sum_{\{T \mid C \in T\}} L_M(R, T) + \sum_{\{T \mid C \in T\}} E_M(R, T) < n,$$

then (A2)

$$\sum_{\substack{\{M' \mid\ C \in T_{M'},\ \textit{not}\ \lim_R(T_M, T_{M'}),\\ \textit{and not}\ \lim_{\mathsf{Inv}(R)}(T_{M'}, T_M)\}}} x_{M'} > 0,$$

Now Point (b.ii) is addressed by the side-condition (A1). The fact that we require only the existence of a *single* instance in the post-condition is due to the fact that we work in pre-models and can simply introduce an appropriate multiple edge to satisfy requirements for larger numbers of instances. Finally, Point (c) from above is ensured using side-condition (A2).

The following lemma shows that our inqualities and side-conditions are indeed appropriate.

Lemma 10 *The system of inequalities* $\mathcal{E}_{C_0,\mathcal{T}}$ *has an admissible solution iff* C_0 *is finitely satisfiable w.r.t.* $\mathcal{T}$*.*

Intuitively, the proof of Lemma 10 proceeds as follows: for the "if" direction we simply take a finite model $\mathcal{I}$ for C_0 and $\mathcal{T}$ (as every model is also a pre-model), and then define an admissible solution for the equation system by taking, for each variable x_M, the number of instances of M in $\mathcal{I}$. For the "only if" direction, we construct a pre-model for $\mathcal{I}$ and $\mathcal{T}$ by reserving domain elements for each mosaic as indicated by an admissible solution of $\mathcal{E}_{C_0,\mathcal{T}}$, and then refer to the inequalities and side-conditions to show that we can indeed turn the reserved domain elements into instances of the corresponding mosaic by connecting them via roles in an appropriate way. It then remains to refer to Lemma 7 for the existance of a finite (standard) model. As the "only if" direction nicely illustrates the purpose of the individual inequalities and side-conditions, we give the proof here. The proof of the "if" direction can be found in Appendix A.

Proof. We only prove the "only if" direction here. Let $\{\hat{x}_M \mid M \text{ a mosaic}\}$ be an admissible solution of $\mathcal{E}_{C_0,\mathcal{T}}$. We construct a finite pre-interpretation $\mathcal{I}$ from this solution and then show that it is a pre-model of C_0 and $\mathcal{T}$. For each mosaic M, fix a set $\hat{M}$ (of instances) such that $\#\hat{M} = \hat{x}_M$ and $M \neq M'$ implies $\hat{M} \cap \hat{M}' = \emptyset$. We define

$$\Delta^{\mathcal{I}} = \bigcup \hat{M}.$$

In the following, for all $e \in \Delta^{\mathcal{I}}$, we use $m(e)$ to denote the mosaic M with $e \in \hat{M}$, and $t(e)$ to denote the type $T_{m(e)}$. For each concept name $A \in \mathbf{C}$, we put

$$A^{\mathcal{I}} := \{e \in \Delta^{\mathcal{I}} \mid A \in t(e)\}.$$

Role names $R \in \mathbf{R}$ are harder to deal with. More precisely, in the construction of their interpretation, we distinguish between the three cases identified on Page 11. We start with Case (1): for each role $R \in \mathsf{rol}(C_0, \mathcal{T})$ and each pair of types $T, T' \in \mathsf{type}(C_0, \mathcal{T})$ such that $\lim_R(T, T')$ but not $\lim_{\mathsf{Inv}(R)}(T', T)$, we construct a mapping

$$\gamma^R_{T,T'} : \bigcup_{\{M \mid T_M = T\}} \hat{M} \times \bigcup_{\{M \mid T_M = T'\}} \hat{M} \to \mathbb{N}$$

(such mappings will henceforth be called *multiplicity mappings*) such that

(1) for each e with $t(e) = T$, we have

$$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid t(e') = T'\}} \gamma^R_{T,T'}(e, e') \geq L_{m(e)}(R, T');$$

(2) for each e' with $t(e') = T'$, we have

$$\sum_{\{e \in \Delta^{\mathcal{I}} \mid t(e) = T\}} \gamma^R_{T,T'}(e, e') = E_{m(e')}(\mathsf{Inv}(R), T).$$

Intuitively, the $\gamma^R_{T,T'}$ function is the "part" of $R^{\mathcal{I}}$ that deals with edges from elements of type T to elements of type T'. The construction proceeds as follows. First define two sets

$$\Delta_T := \{(e, i) \in \Delta^{\mathcal{I}} \times \mathbb{N} \mid t(e) = T \text{ and } i < L_{m(e)}(R, T')\}$$
$$\Delta_{T'} := \{(e, i) \in \Delta^{\mathcal{I}} \times \mathbb{N} \mid t(e) = T' \text{ and } i < E_{m(e)}(\mathsf{Inv}(R), T)\}$$

By Equation $(E2)$, we find a (total) injection f from Δ_T to Δ'_T. We define a multiplicity mapping r by setting $r(d, e) := \sharp\{(i, j) \in \mathbb{N}^2 \mid f(e, i) = (d, j)\}$. It is easily checked that, by setting $\gamma^R_{T,T'} := r$, we satisfy Condition (1) from above, but only the following weakening of Condition (2):

(2′) for each e' with $t(e') = T'$, we have

$$\sum_{\{e \in \Delta^{\mathcal{I}} \mid t(e)=T\}} \gamma^R_{T,T'}(e, e') \leq E_{m(e')}(\mathsf{Inv}(R), T).$$

If Condition (2) is satisfied accidentally, we are done. If it is not, then we can "augment" r appropriately to satisfy Condition (2) without destroying Condition (1). This is realised in two steps. First, if r does not accidentally satisfy (2), then there is an e' with $t(e') = T'$ and

$$\sum_{\{e \in \Delta^{\mathcal{I}} \mid t(e)=T\}} \gamma^R_{T,T'}(e, e') < E_{m(e')}(\mathsf{Inv}(R), T).$$

Then $\hat{x}_{m(e')} \neq 0$ and $E_{m(e')}(\mathsf{Inv}(R), T) > 0$. Hence, by side-condition (A1), there exists a mosaic M such that $\hat{M} \neq \emptyset$ and $T_M = T$. Fix an $e_M \in \hat{M}$. Second, for each e' with $t(e') = T'$, we define

$$\mathsf{miss}(e') := E_{m(e')}(\mathsf{Inv}(R), T) - \sum_{\{e \in \Delta^{\mathcal{I}} \mid t(e)=T\}} \gamma^R_{T,T'}(e, e').$$

We can now define $\gamma^R_{T,T'}$:

$$\gamma^R_{T,T'}(d, e') := \begin{cases} r(d, e') + \mathsf{miss}(e') \text{ if } d = e_M \\ r(d, e') \text{ otherwise.} \end{cases}$$

It is readily checked that Conditions (1) and (2) are now both satisfied. We have thus finished the construction of $\gamma^R_{T,T'}$.

Now we deal with Case (2) from Page 11: for each role name R and each pair of types $T, T' \in \mathsf{type}(C_0, \mathcal{T})$ such that $\lim_R(T, T')$ and $\lim_{R^-}(T', T)$, we construct a multiplicity mapping $\lambda^R_{T,T'}$ such that

(1) for each e with $t(e) = T$, we have

$$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid t(e')=T'\}} \lambda^R_{T,T'}(e, e') = E_{m(e)}(R, T');$$

(2) for each e' with $t(e') = T'$, we have

$$\sum_{\{e \in \Delta^{\mathcal{I}} \mid t(e)=T\}} \lambda^R_{T,T'}(e, e') = E_{m(e')}(\mathsf{Inv}(R), T).$$

The construction is is similar to that of $\gamma^R_{T,T'}$, but simpler: First define two sets

$$\Delta_T := \{(e, i) \in \Delta^{\mathcal{I}} \times \mathbb{N} \mid t(e) = T \text{ and } i < E_{m(e)}(R, T')\}$$
$$\Delta_{T'} := \{(e, i) \in \Delta^{\mathcal{I}} \times \mathbb{N} \mid t(e) = T' \text{ and } i < E_{m(e)}(\mathsf{Inv}(R), T)\}$$

By Equation ($E3$), we find a bijection f from Δ_T to Δ'_T. We then define $\lambda^R_{T,T'} := \sharp\{(i,j) \in \mathbb{N}^2 \mid f(e,i) = (d,j)\}$. It is easily checked that Conditions (1) and (2) are satisfied, and thus we are done.

Finally, we address the simplest case from Page 11: Case (3). Let $n \in \mathbb{N}$ be a supremum of the numbers used inside number restrictions in C_0 and $\mathcal{T}$. For each role name R and each pair of types $T, T' \in \mathsf{type}(C_0, \mathcal{T})$ such that neither $\lim_R(T,T')$ nor $\lim_{R^-}(T',T)$, we define a multiplicity mapping $\omega^R_{T,T'}$ by setting $\omega^R_{T,T'}(d,e) := n$ for all d, e with $t(e) = T$ and $t(e') = T'$.

We are now ready to assemble the interpretation $R^{\mathcal{I}}$ of role names: for any two $d, e \in \Delta^{\mathcal{I}}$ with $t(e) = T$ and $t(e') = T'$, set

$$R^{\mathcal{I}}(d,e) := \begin{cases} \gamma^R_{T,T'}(d,e) \text{ if } \lim_R(T,T') \text{ and not } \lim_{\mathsf{Inv}(R)}(T',T) \\ \gamma^{R^-}_{T',T}(e,d) \text{ if not } \lim_R(T,T') \text{ and } \lim_{\mathsf{Inv}(R)}(T',T) \\ \lambda^R_{T,T'}(d,e) \text{ if } \lim_R(T,T') \text{ and } \lim_{\mathsf{Inv}(R)}(T',T) \\ \omega^R_{T,T'}(d,e) \text{ if neither } \lim_R(T,T') \text{ nor } \lim_{\mathsf{Inv}(R)}(T',T) \end{cases}$$

It remains to show that $\mathcal{I}$ is a pre-model of C_0 and $\mathcal{T}$. To this end, we first establish a claim showing that all lower bounds L_M of mosaics are met in $\mathcal{I}$.

Claim 1: For all $e \in \Delta^{\mathcal{I}}$ with $m(e) = M$ and $t(e) = T$, roles R, and types T' with $\lim_R(T,T')$ and not $\lim_{\mathsf{Inv}(R)}(T',T)$, we have

$$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid t(e') = T'\}} R^{\mathcal{I}}(e,e') \geq L_M(R,T'). \qquad (*)$$

Proof: Let e, R, and T be as in the claim. We distinguish two cases:

- R is a role name. By construction of $R^{\mathcal{I}}$, we have $R^{\mathcal{I}}(e,e') = \gamma^R_{T,T'}(e,e')$ for all e' with $t(e') = T'$. Thus Property (1) of $\gamma^R_{T,T'}$ immediately yields $(*)$.
- $R = S^-$ for some role name S. By construction of $S^{\mathcal{I}}$ and the semantics of inverse roles, we have $R^{\mathcal{I}}(e,e') = S^{\mathcal{I}}(e',e) = \gamma^{S^-}_{T,T'}(e,e')$. Thus Property (1) of $\gamma^{S^-}_{T,T'}$ yields $(*)$.

The next claim addresses all exact bounds E_M.

Claim 2: For all $e \in \Delta^{\mathcal{I}}$ with $m(e) = M$ and $t(e) = T$, roles R, and types T' with $\lim_{\mathsf{Inv}(R)}(T',T)$, we have

$$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid t(e') = T'\}} R^{\mathcal{I}}(e,e') = E_M(R,T'). \qquad (*)$$

Proof: Let e, R, and T be as in the claim. We establish the claim using a case distinction:

- *Not* $\lim_R(T, T')$ and R is a role name. By construction of $R^{\mathcal{I}}$, we have $R^{\mathcal{I}}(e, e') = \gamma^{R^-}_{T',T}(e', e)$ for all e' with $t(e') = T'$. Thus Property (2) of the multiplicity mapping $\gamma^{R^-}_{T',T}$ yields $(*)$.
- *Not* $\lim_R(t(e), T')$ and $R = S^-$ for some role name S. By construction of $S^{\mathcal{I}}$ and the semantics of inverse roles, we have $R^{\mathcal{I}}(e, e') = S^{\mathcal{I}}(e', e) = \gamma^{S}_{T',T}(e', e)$. Thus, we again obtain $(*)$ by Property (2) of $\gamma^{S}_{T',T}$.
- $\lim_R(t(e), T')$ and R is a role name. By construction of $R^{\mathcal{I}}$, we have $R^{\mathcal{I}}(e, e') = \lambda^{R}_{T,T'}(e, e')$ for all e' with $t(e') = T'$. Thus Property (1) of $\lambda^{R}_{T,T'}$ yields $(*)$.
- $\lim_R(t(e), T')$ and $R = S^-$ for some role name S. By construction of $S^{\mathcal{I}}$ and the semantics of inverse roles, we have $R^{\mathcal{I}}(e, e') = S^{\mathcal{I}}(e', e) = \lambda^{S}_{T',T}(e', e)$. Thus Property (2) of $\lambda^{R}_{T,T'}$ yields $(*)$.

We can now prove the claim that is central for showing that $\mathcal{I}$ is a pre-model of the input concept C_0 and the input TBox $\mathcal{T}$:

Claim 3: For all $C \in \mathsf{cl}(C_0, \mathcal{T})$ and all $e \in \Delta^{\mathcal{I}}$, $C \in t(e)$ implies $e \in C^{\mathcal{I}}$.

The proof is by induction on the *norm* of concepts C as introduced in the proof of Theorem 7. Let $e \in \Delta^{\mathcal{I}}$ such that $C \in t(e)$.

- C is a concept name. Then $e \in C^{\mathcal{I}}$ follows from the definition of $\mathcal{I}$.
- $C = \neg D$. Since every concept in $\mathsf{cl}(C_0, \mathcal{T})$ is in NNF, D is a concept name. If $\neg D \in t(e)$, then $D \notin t(e)$ by definition of types. Thus $e \in (\neg D)^{\mathcal{I}}$ by definition of $\mathcal{I}$.
- For $C = D \sqcap E$ or $C = D \sqcup E$, the claim follows immediately from the definition of types and the induction hypothesis.
- $C = (\leqslant n\ R\ D)$. We show that

  $$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid D \in t(e')\}} R^{\mathcal{I}}(e, e') \leq n. \qquad (*)$$

  It then follows that $e \in C^{\mathcal{I}}$ as required, as we can show that $D \notin t(e')$ implies $e' \notin D^{\mathcal{I}}$: by definition of types, $D \notin t(e')$ implies $\dot{\neg} D \in t(e')$. Since we are performing induction on the norm of concepts, induction hypothesis thus yields $e' \in (\dot{\neg} D)^{\mathcal{I}}$, and $e' \notin D^{\mathcal{I}}$ follows by the semantics.

  It thus remains to establish $(*)$, which is simple: $C \in t(e)$ and $D \in t(e')$ implies $\lim_{\mathsf{Inv}(R)}(t(e'), t(e))$. Thus by Claim 2 we can rewrite $(*)$ as

  $$\sum_{\{T \mid D \in T\}} E_{m(e)}(R, T) \leq n.$$

  This, however, is ensured by Property ($\mathsf{M3}$) of mosaics.
- $C = (\geqslant n\ R\ D)$. We show that

  $$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid D \in t(e')\}} R^{\mathcal{I}}(e, e') \geq n. \qquad (**)$$

  It then clearly follows from the induction hypothesis that $e \in C^{\mathcal{I}}$ as required.

Claims 1 and 2 together with Properties (M1) and (M2) of mosaics imply that

$$\sum_{\{e' \in \Delta^{\mathcal{I}} \mid D \in t(e')\}} R^{\mathcal{I}}(e, e') \geq \sum_{\{T \mid D \in T\}} L_{m(e)}(R, T) + \sum_{\{T \mid D \in T\}} E_{m(e)}(R, T)$$

If the right-hand side of this inequality is greater or equal to n, then we are done. Otherwise, (A2) ensures that there exists a mosaic M such that $D \in T_M$, not $\lim_R(t(e), T_M)$, not $\lim_{\mathsf{Inv}(R)}(T_M, t(e))$, and $\hat{x}_M \neq 0$, i.e. there is an $e' \in \hat{M}$. First assume that R is a role name. By construction of $R^{\mathcal{I}}$, we have $R^{\mathcal{I}}(e, e') = \omega^R_{T,T'} \geq n$. Thus, $(**)$ is satisfied and we are done. Now let $R = S^-$ for a role name S. Then we have $R^{\mathcal{I}}(e, e') = S^{\mathcal{I}}(e', e) = \omega^S_{T',T} \geq n$ and are also done.

As a consequence, $\mathcal{I}$ is a pre-model of C_0 and $\mathcal{T} = \{\top \doteq C_{\mathcal{T}}\}$: by Equation (E1) and due to the fact that $\hat{x}_M > 0$ implies $\#\hat{M} > 0$, there is a mosaic M such that $C_0 \in T_M$ and $\#\hat{M} > 0$. Fix an $e \in \hat{M}$. Claim 3 implies that $e \in C_0^{\mathcal{I}}$ and thus $\mathcal{I}$ is a pre-model of C_0. Moreover, by definition of types, we have $C_{\mathcal{T}} \in T_M$ for each mosaic M. This fact together with Claim 3 implies that $\mathcal{I}$ is a pre-model of $\mathcal{T}$. □

To establish the intended ExpTime upper bound, it now remains to show that (i) the size of the constructed equation system $\mathcal{E}_{C_0,\mathcal{T}}$ is (at most) exponential in the size of C_0 and $\mathcal{T}$, and (ii) the existence of admissible solutions can be checked in polynomial time.

We start with defining the size of concepts and TBoxes. First, the *size w.r.t. unary coding* of concepts is defined inductively as follows:

$$|A|_u = 1 \text{ for } A \text{ a concept name,}$$
$$|\neg C|_u = 1 + |C|_u, \qquad |C_1 \sqcap C_2|_u = |C_1 \sqcup C_2|_u = |C_1|_u + |C_2|_u$$
$$|(\leqslant n\ R\ C)|_u = |(\geqslant n\ R\ C)|_u = n + 1 + |C|_u$$

The size of a TBox $\mathcal{T}$ is defined as $|C_{\mathcal{T}}|_u$. It can easily be shown that the cardinality of $\mathsf{cl}(C_0, \mathcal{T})$ is linear in the size of C_0 and $\mathcal{T}$.

Now we determine the number of mosaics for C_0 and $\mathcal{T}$. Let n be the size of C_0 plus the size of $C_{\mathcal{T}}$ w.r.t. unary coding. The cardinality of $\mathsf{type}(C_0, \mathcal{T})$ is exponential in n. For mosaics, (M2) and (M3) imply

$$\#\{(R, T) \mid E_M(R, T) > 0\} \leq \mathsf{sum}^{\leqslant}(T_M)$$

and $\max(\mathsf{ran}(E_M)) \leq \mathsf{max}^{\leqslant}(T_M)$, whereas (M4) implies analogous bounds for L_M. Since $\mathsf{max}^{\bowtie}(T)$ and $\mathsf{sum}^{\bowtie}(T)$ are linear in n for $\bowtie \in \{\leqslant, \geqslant\}$, each mosaic M can be represented by T_M and a vector of length $2n$ of pairs of the form

(k, T) for $k \leq n$ and T a type. This implies the existence of a constant c such that the number of mosaics is bounded by $2^{(cn^2)}$.

Since the number of mosaics is exponential in the size of C_0 and $\mathcal{T}$, we can easily infer similar bounds for the number of inequalities and side-conditions of $\mathcal{E}_{C_0,\mathcal{T}}$. Before we continue, however, let us analyze what bounds are needed. To do this, we show that the existence of an admissible solution for systems of inequalities $\mathcal{E}_{C_0,\mathcal{T}}$ can be decided in time polynomial in certain parameters of $\mathcal{E}_{C_0,\mathcal{T}}$.

First we need some prerequisites. We assume linear inequalities to be of the form $\Sigma_i c_i x_i \geq b$. Such an inequality is called *positive* if $b \geq 0$. A system of linear inequalities is described by a tuple $(V, \mathcal{E})$, where V is a set of variables and $\mathcal{E}$ a set of inequalities. Such a system is called *simple* if all inequalities are positive and all coefficients are (possibly negative) integers.

A *side condition* for an inequality system $(V, \mathcal{E})$ is a constraint of the form

$$x > 0 \Longrightarrow x_1 + \cdots + x_\ell > 0, \text{ where } x, x_1, \ldots x_\ell \in V.$$

Let $(V, \mathcal{E})$ be an inequality system and I a set of side conditions for $(V, \mathcal{E})$. We say that $(V, \mathcal{E})$ admits an *I-admissible solution* if it admits a solution satisfying all constraints from I.

It is not hard to check that the inequality systems from Definition 8 are simple and that the conditions (A1) and (A2) can be polynomially transformed into side conditions:

- (E1) is already simple,
- (E2) can obviously be transformed into $\sum \ldots - \sum \ldots \geq 0$,
- the equality (E3) is transformed into two inequalities of the form $\sum \ldots - \sum \ldots \geq 0$,
- each implication due to (A1) can be transformed into polynomially many side conditions as follows: since we are interested in non-negative solutions only, we can use a separate implication for each summand appearing in the premise. Next, the coefficients on the left-hand sides of the premise are omitted by dropping those side-conditions whose coefficient is zero and replacing all other coefficients with 1.
- (A2) is already in the form of a side condition.

The following proposition states that the existence of I-admissible integer solutions can be checked in time polynomial in several parameters. It is a generalization of Lemma 6.1.5 in [23].

Proposition 11 *Let $(V, \mathcal{E})$ be a simple system of inequalities in which all coefficients and constants are from the interval $[-a; a]$ of integers, and let I*

be a set of side conditions for $(V, \mathcal{E})$*. Then the existence of an integer, non-negative, and* I*-admissible solution for* $(V, \mathcal{E})$ *can be decided in (deterministic) time polynomial in* $\#V + \#\mathcal{E} + \#I + a$*.*

It is now easy to obtain the desired ExpTime upper bound. First, note that the number of variables and the number of inequalities in $\mathcal{E}_{C_0,\mathcal{T}}$ is at most exponential in the size of C_0 and $\mathcal{T}$ due to our bound on the number of mosaics. Second, the coefficients and constants appearing in $\mathcal{E}_{C_0,\mathcal{T}}$ are linear in the size of C_0 and $\mathcal{T}$ due to (M2) to (M4). When transforming $\mathcal{E}_{C_0,\mathcal{T}}$ into simple inequalities and side conditions, these properties are preserved. Thus, Lemmas 10 and 11 yield an ExpTime upper bound for the satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes. The corresponding lower bound is a consequence of the ExpTime-hardness of unrestricted satisfiability of $\mathcal{ALC}$ w.r.t. TBoxes [20; 3] and the fact that this DL has the finite model property.

Theorem 12 *Finite satisfiability of* $\mathcal{ALCQI}$*-concepts w.r.t. TBoxes is* ExpTime*-complete if numbers are coded in unary.*

If numbers in number restrictions are coded binarily, the algorithm developed in this section does no longer yield an ExpTime upper bound: in this case, the number of mosaics is double exponential in the size of the input concept and TBox. Since it is not clear whether and how the presented algorithm can be modified in order to yield an ExpTime upper bound for the case of binary coding, we resort to a different approach to attacking this problem: in the next section, we reduce finite $\mathcal{ALCQI}$-satisfiability to the finite satisfiability of $\mathcal{ALCFI}$-concepts. Since the employed reduction is polynomial, in this way we obtain an ExpTime upper bound for the finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes, even if numbers are coded in binary.

4 Binary Coding of Numbers

In this section, we prove that finite $\mathcal{ALCQI}$-concept satisfiability w.r.t. TBoxes is decidable in ExpTime even if numbers are coded in binary, where the size w.r.t. binary coding $|C|_b$ of a concept C is defined as the size w.r.t. unary coding, the only difference being that

$$|(\leqslant n\ R\ C)|_b = |(\geqslant n\ R\ C)|_b = \log(n) + 1 + |C|_b.$$

The proof is by a polynomial reduction to finite $\mathcal{ALCFI}$-concept satisfiability w.r.t. TBoxes. Since, in the case of $\mathcal{ALCFI}$, the size of numbers appearing in number restrictions is constant (regardless of the coding), the results presented in the previous section imply that finite $\mathcal{ALCFI}$-concept satisfiability

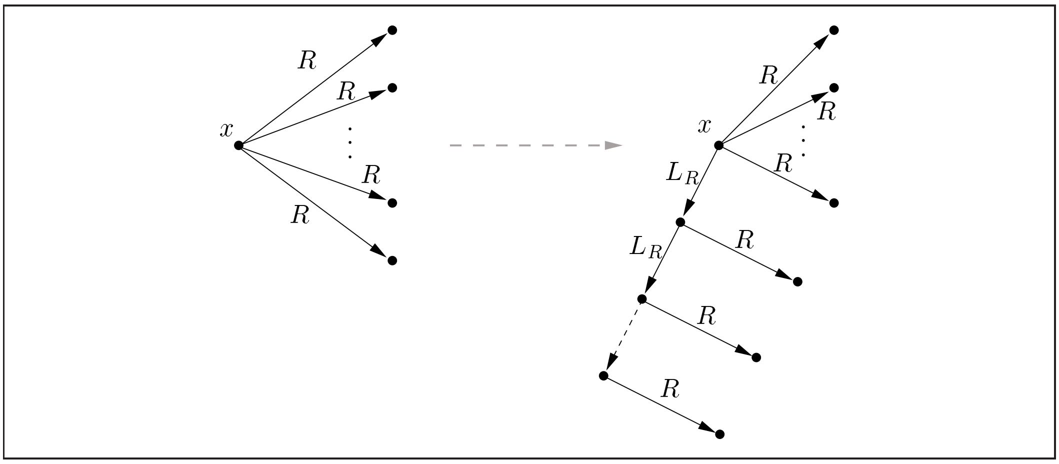

Fig. 5. Representing role neighbour relationships.

w.r.t. TBoxes is ExpTime-complete. Thus, this logic is a suitable target for reduction. In contrast to existing reductions of $\mathcal{ALCQI}$ to $\mathcal{ALCFI}$, which only work in the case of potentially infinite models (such as the one presented in [11]), we have to take special care to deal with finite (and thus non-tree) models.

Before we go into technical details, let us describe the intuition behind the reduction. The general idea is to replace counting via qualified number restrictions with counting via concept names: to count up to a number n, we reserve concept names $B_0, \ldots, B_{\lceil \log(n) \rceil}$ representing the bits of numbers between 0 and n. For the actual counting, we can then use well-known (propositional logic) formulae that encode incrementation. But how can we use this approach to count the number of role neighbour? Intuitively, we rearrange the neighbours of each domain element in a way that allows to replace qualifying number restrictions with the combination of (i) functionality of roles as provided by $\mathcal{ALCFI}$ and (ii) counting via concept names. Consider, for example, the domain element x and its R-neighbours displayed on the left-hand side of Figure 5. Ignoring the "direct" R-neighbours of x on the right-hand side for a moment, we have rearranged three R-neighbours along an auxiliary path that is built using a new role L_R. Employing the $(\leqslant 1\ S\ \top)$ constructor of $\mathcal{ALCFI}$, we can ensure that each node on this path has precisely one L_R-predecessor, at most one L_R-neighbour, and precisely one R-neighbour. The counting via concept names is then performed along the domain elements on L_R-paths.

However, we cannot gather all original R-neighbours of x on the L_R-path. The reason for this is as follows: assume we are at some domain element on the L_R-path descending from x and move along this domain element's outgoing R-edge. The reduction ensures that we either reach a "real" domain element (such as x) or arrive on an $L_{\mathsf{Inv}(R)}$-path. If the latter is the case, we have to ensure that, moving up the $L_{\mathsf{Inv}(R)}$-path, we will finally reach a "real" domain element. To do this, we count the lengths of auxiliary paths via concept names:[3] once we have moved up to node 0 of the path, its predecessor must be "real". Since, however, we do not know how many R-neighbours an object had in the original model, we do not know how many bits to reserve for this counting. The solution is to gather only those R-neighbours of x on the L_R-path which are constrained by a $(\leqslant n\ R\ C)$ concept applying to x or which are witnesses for a $(\geqslant n\ R\ C)$ concept applying to x—this helps since the number of such domain elements is known in advance. All other domain elements can remain "direct" neighbours of x since there is no need to count them.

Fix an $\mathcal{ALCQI}$-concept C and an $\mathcal{ALCQI}$-TBox $\mathcal{T}$ whose finite satisfiability is to be decided. W.l.o.g., we assume C and $\mathcal{T}$ to be in NNF. In order to translate C and $\mathcal{T}$ to $\mathcal{ALCFI}$, we introduce some additional concept and role names:

(1) a fresh (i.e., not appearing in C or $\mathcal{T}$) concept name Real;
(2) for each $R \in \mathsf{rol}(C, \mathcal{T})$, a fresh concept name H_R and a fresh role name L_R;
(3) for each concept $D \in \mathsf{cl}(C, \mathcal{T})$ of the form $(\bowtie n\ R\ E)$, where $\bowtie$ is used as a placeholder for $\geqslant$ or $\leqslant$, we reserve a fresh concept name X_D;
(4) for each concept $D \in \mathsf{cl}(C, \mathcal{T})$ that appears inside a qualifying number restriction $(\bowtie n\ R\ D) \in \mathsf{cl}(C, \mathcal{T})$, we reserve fresh concept names $B_{D,0}, \ldots, B_{D,k}$, where $k = \lceil \log(\mathsf{num}_D) \rceil$ and

$$\mathsf{num}_D = \max\{n \mid (\bowtie n\ R\ D) \in \mathsf{cl}(C, \mathcal{T})\} + 1;$$

(5) for each role $R \in \mathsf{rol}(C, \mathcal{T})$, we reserve fresh concept names $B_{R,0}, \ldots, B_{R,k}$, where $k = \lceil \log(\mathsf{depth}_R) \rceil$ and

$$\mathsf{depth}_R = \sum_{(\bowtie n\ R\ C) \in \mathsf{cl}(C, \mathcal{T})} n.$$

The concept name Real is used to distinguish "real" domain elements from domain elements on auxiliary paths. The concept names H_R are used to "mark" objects on auxiliary paths for the role R: when following an L_R-path, all encountered objects (apart from the root representing a "real" domain element) will be instances of H_R. The concept names $B_{R,i}$ are used to count the length of auxiliary L_R-paths as described above. The concept names $B_{D,i}$ are also employed for counting: they are used to count the "occurrence" of R-neighbours in D along L_R-paths and will thus help to replace $\mathcal{ALCQI}$-concepts of the form $(\bowtie\ n\ R\ D)$. Note that the number of newly introduced concept and role names is polynomial in the size of C and $\mathcal{T}$. We will use $\overline{B_D}$ to refer to the number encoded by the concept names $B_{D,0}, \ldots, B_{D,\lceil \log(\mathsf{num}_D) \rceil}$ and $\overline{B_R}$ to refer to the number encoded by the concept names $B_{R,0}, \ldots, B_{R,\lceil \log(\mathsf{depth}_R) \rceil}$. Moreover, we will use the following abbreviations:

[3] This counter is a different one than the ones mentioned above

- $(\overline{B_R} = i)$ to denote the $\mathcal{ALCFI}$-concept expressing that $\overline{B_R}$ equals i (and similar for $\overline{B_D} = i$ and the comparisons "<" and ">");
- $\mathsf{incr}(\overline{B_R}, S)$ to denote the $\mathcal{ALCFI}$-concept expressing that, for all S-successors, the number $\overline{B_R}$ is incremented by 1 modulo depth_R (and similar for $\mathsf{incr}(\overline{B_D}, S)$). More precisely, the concept $\mathsf{incr}(\overline{B_R}, S)$ is defined as follows (with n abbreviating $\lceil \log(\mathsf{depth}_R) \rceil$):

$$
\begin{aligned}
&(B_{R,0} \rightarrow \forall S.\neg B_{R,0}) \sqcap (\neg B_{R,0} \rightarrow \forall S.B_{R,0}) \sqcap \\
&\bigsqcap_{k=1..n} \Big(\bigsqcap_{j=0..k-1} B_{R,j} \Big) \rightarrow \Big((B_{R,k} \rightarrow \forall S.\neg B_{R,k}) \sqcap (\neg B_{R,k} \rightarrow \forall S.B_{R,k}) \Big) \sqcap \\
&\bigsqcap_{k=1..n} \Big(\bigsqcup_{j=0..k-1} \neg B_{R,j} \Big) \rightarrow \Big((B_{R,k} \rightarrow \forall S.B_{R,k}) \sqcap (\neg B_{R,k} \rightarrow \forall S.\neg B_{R,k}) \Big).
\end{aligned}
$$

- $\mathsf{eq}(\overline{B_D}, S)$ to denote the $\mathcal{ALCFI}$-concept expressing that, for all S-successors, the number $\overline{B_R}$ is not changed. Formally, $\mathsf{eq}(\overline{B_R}, S)$ is defined as follows (with n abbreviating $\lceil \log(\mathsf{depth}_R) \rceil$):

$$
\bigsqcap_{i=1..n} ((B_{D,i} \rightarrow \forall L_R.B_{D,i}) \sqcap (\neg B_{D,i} \rightarrow \forall L_R.\neg B_{D,i}))\Big)
$$

We inductively define a translation $\gamma(C)$ of the concept C into a Boolean formula (which is also an $\mathcal{ALCFI}$-concept):

$$
\begin{aligned}
\gamma(A) &:= A, \text{ for } A \in \mathsf{cnam}(C, \mathcal{T}) & \gamma(\neg D) &:= \neg\gamma(D) \\
\gamma(D \sqcap E) &:= \gamma(D) \sqcap \gamma(E) & \gamma(D \sqcup E) &:= \gamma(D) \sqcup \gamma(E) \\
\gamma(\geqslant n\ R\ D) &:= X_{(\geqslant n\ R\ D)} & \gamma(\leqslant n\ R\ D) &:= X_{(\leqslant n\ R\ D)}
\end{aligned}
$$

Now set $\sigma(C) := \gamma(C) \sqcap \mathsf{Real}$ and, for $\mathcal{T} = \{\top \doteq C_\mathcal{T}\}$,

$$
\sigma(\mathcal{T}) := \{\top \doteq \mathsf{Real} \rightarrow \gamma(C_\mathcal{T})\} \cup \mathsf{Aux}(C, \mathcal{T}),
$$

where the TBox $\mathsf{Aux}(C, \mathcal{T})$ is defined in Figure 6 in which we use $D \sqsubseteq E$ as abbreviation for $\top \doteq D \rightarrow E$, and in which all $\bigsqcup$ and $\bigsqcap$ that have only a concept as index range over all concepts in $\mathsf{cl}(C, \mathcal{T})$ of the specified form.

The first three concept equations ensure the behaviour sketched above of Real, H_R, and the counting concepts B_R and B_D. The last but one concept equation ensures that the counting concepts B_D are updated correctly along an L_R path. To guarantee that a "real" element d satisfies "number restrictions" $X_{(\bowtie n\ R\ D)}$, the fourth concept equation ensures that we see enough R-neighbours in D for atleast restrictions $(\geqslant n\ R\ D)$ along an L_R path starting at d, whereas the last concept equation guarantees that we do not see too many such neighbours along an L_R path for atmost restrictions $(\leqslant n\ R\ D)$. The following Lemma states that σ is a reduction from finite $\mathcal{ALCQI}$-concept satisfiability to finite $\mathcal{ALCFI}$-concept satisfiability.

$$
\begin{aligned}
\top \doteq \bigsqcap_{R \in \mathsf{rol}(C,\mathcal{T})} \Big(& \forall R.(\mathsf{Real} \sqcup H_{\mathsf{Inv}(R)}) \sqcap \\
& \forall L_R.H_R \sqcap \\
& (\leqslant 1\ L_R\ \top) \sqcap \\
& \bigsqcap_{(\bowtie n\ S\ D)} \Big(X_{(\bowtie n\ S\ D)} \leftrightarrow \forall L_R.X_{(\bowtie n\ S\ D)} \Big) \sqcap \\
& \bigsqcap_{A \in \mathsf{cnam}(C,\mathcal{T})} (A \leftrightarrow \forall L_R.A) \Big) \sqcap \\
\bigsqcap_{D} & \neg\gamma(D) \rightarrow \gamma(\dot{\neg}(D))
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{Real} \sqsubseteq \bigsqcap_{R \in \mathsf{rol}(C,\mathcal{T})} \Big(& \forall L_R.(\overline{B_R} = 0) \sqcap \\
& (\leqslant 0\ L_R^-\ \top) \Big) \sqcap \\
\bigsqcap_{(\bowtie n\ R\ D)} & \Big(X_{(\bowtie n\ R\ D)} \rightarrow \forall L_R.(\overline{B_D} = 0) \Big) \sqcap \\
\bigsqcap_{(\leqslant n\ R\ D)} & \Big(X_{(\leqslant n\ R\ D)} \rightarrow \forall R.\neg\gamma(D) \Big) \sqcap \\
\bigsqcap_{\substack{(\geqslant n\ R\ D) \\ \text{with } n>0}} & \Big(X_{(\geqslant n\ R\ D)} \rightarrow \exists L_R.\top \Big)
\end{aligned}
$$

$$
\begin{aligned}
H_R \sqsubseteq\ & (=1\ R\ \top) \sqcap \\
& (=1\ L_R^-\ \top) \sqcap \\
& \mathsf{incr}(\overline{B_R}, L_R) \sqcap \\
& (\overline{B_R} = 0) \rightarrow \exists L_R^-.\mathsf{Real} \sqcap \\
& (\overline{B_R} > 0) \rightarrow \exists L_R^-.H_R \sqcap \\
& (\overline{B_R} = (\mathsf{depth}_R - 1)) \rightarrow (\leqslant 0\ L_R\ \top)
\end{aligned}
$$

$$
H_R \sqsubseteq \bigsqcap_{(\geqslant n\ R\ D)} \Big((X_{(\geqslant n\ R\ D)} \sqcap \overline{B_D} < n \sqcap \forall R.\neg\gamma(D)) \rightarrow \exists L_R.\top \Big)
$$

$$
\begin{aligned}
H_R \sqsubseteq \bigsqcap_{(\bowtie n\ R\ D)} \Big(& \exists R.\gamma(D) \rightarrow \mathsf{incr}(\overline{B_D}, L_R) \sqcap \\
& \forall R.\gamma(\dot{\neg} D) \rightarrow \mathsf{eq}(\overline{B_D}, L_R) \Big)
\end{aligned}
$$

$$
H_R \sqsubseteq \bigsqcap_{(\leqslant n\ R\ D)} \Big((X_{(\leqslant n\ R\ D)} \sqcap \overline{B_D} = n) \rightarrow \forall R.\neg\gamma(D) \Big)
$$

Fig. 6. The TBox $\mathsf{Aux}(C, \mathcal{T})$.

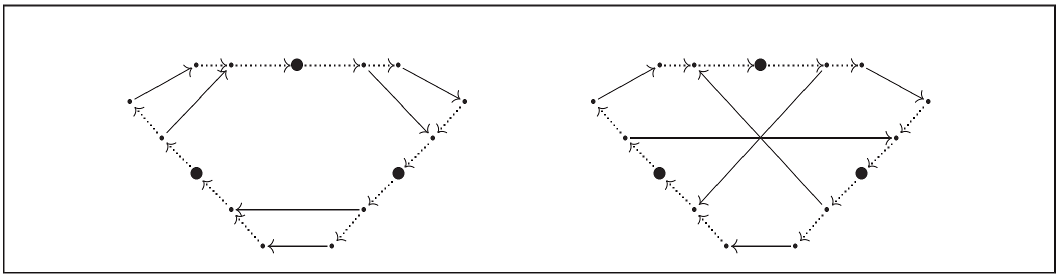

Fig. 7. Two models for $\sigma(C)$ and $\sigma(\mathcal{T})$.

Lemma 13 *A concept C is finitely satisfiable w.r.t. a TBox $\mathcal{T}$ iff $\sigma(C)$ is finitely satisfiable w.r.t. $\sigma(\mathcal{T})$.*

Intuitively, the proof of the above lemma proceeds as follows: for the "only if" direction, we simply take a finite model of C and $\mathcal{T}$, define all elements in the model as instances of the concept Real, then form the auxiliary paths adding new elements to the model, define the interpretations of the auxiliary concepts and roles, and manipulate the interpretation of the original roles as described above to obtain a finite model of $\sigma(C)$ and $\sigma(\mathcal{T})$.

The "if" direction needs more work. We first note that a straightforward construction of a model of C and $\mathcal{T}$ from a model of $\sigma(C)$ and $\sigma(\mathcal{T})$ by moving all the origins of role relationships from the auxiliary paths to the instance e of Real where the auxiliary path starts does not work. Let us call this naive approach "spooling". To see that spooling fails, consider the two models of $\sigma(C)$ and $\sigma(\mathcal{T})$ given in Figure 7, where

$$\mathcal{T} = \{\top = (\leqslant 2\ R\ C) \sqcap (\leqslant 2\ R^{-}\ C)\}.$$

The thick points represent real elements, the dotted edges denote auxiliary paths, and the solid edges denote real role relationships. Now, if we apply spooling to the model depicted at the left of Fig. 7, we do not obtain a model of C and $\mathcal{T}$ since each node has exactly one incoming and one outgoing R edge. So, to prove this part of Lemma 13, we first show that, if $\sigma(C)$ is finitely satisfiable w.r.t. $\sigma(\mathcal{T})$, then there is a *singular* finite model of $\sigma(C)$ and $\sigma(\mathcal{T})$: intuitively, in a singular model, an auxiliary path for a role R and an auxiliary path for $\mathsf{Inv}(R)$ are connected via at most one R-edge. In Figure 7, the left model is not singular, whereas the right one is. Then we show that, if we apply spooling to a singular model of $\sigma(C)$ and $\sigma(\mathcal{T})$, we indeed obtain a model of C and $\mathcal{T}$.

The complete proof of Lemma 13 can be found in Appendix B. Interestingly, to show the existence of a singular model, we use the same copying construction that we used in the proof of Theorem 7, and thus this encoding trick cannot be easily extended to work for logics that are not closed under taking disjoint copies of models such as $\mathcal{ALCQI}$ with nominals or C2.

Lemma 13 together with the fact that $\sigma(C)$ and $\sigma(\mathcal{T})$ are computable in polynomial time proves that finite satisfiability of $\mathcal{ALCQI}$ concepts w.r.t. TBoxes is polynomially reducible to finite satisfiability of $\mathcal{ALCFI}$ concepts w.r.t. TBoxes. By Theorem 12 we obtain the following theorem:

Theorem 14 *Finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes is* ExpTime*-complete if numbers are coded in binary.*

5 ABox Consistency

In this section, we extend the complexity bounds obtained in Sections 3 and 4 to a more general reasoning task: finite $\mathcal{ALCQI}$-ABox consistency. As noted in the introduction, ABoxes can be understood as describing a"snapshot" of the world.

Definition 15 (ABox) *Let O be a countably infinite set of* object names. *An* ABox assertion *is an expression of the form $a : C$ or $(a, b) : R$, where a and b are object names, C is an $\mathcal{ALCQI}$-concept, and R a role. An* ABox *is a finite set of ABox assertions.*

Interpretations $\mathcal{I}$ are extended to ABoxes as follows: additionally, the interpretation function $\cdot^{\mathcal{I}}$ maps each object name to an element of $\Delta^{\mathcal{I}}$ such that $a \neq b$ implies $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all $a, b \in O$ (the so-called unique name assumption*). An interpretation $\mathcal{I}$* satisfies *an assertion $a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and an assertion $(a, b) : R$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. It is a* model *for an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. An ABox is called* finitely consistent *w.r.t a TBox $\mathcal{T}$ if it has a finite model that is also a model of $\mathcal{T}$.*

In the following, we will polynomially reduce finite $\mathcal{ALCQI}$-ABox consistency to finite $\mathcal{ALCQI}$-concept satisfiability. Thus, we prove that $\mathcal{ALCQI}$-ABox consistency is ExpTime-complete independently of the way in which numbers are coded. We start with fixing some notation.

Let $\mathcal{A}$ be an ABox and $\mathcal{T}$ a TBox. Analogously to what was done in previous sections, we use $\mathsf{rnam}(\mathcal{A}, \mathcal{T})$ to denote the set of role names appearing in $\mathcal{A}$ and $\mathcal{T}$, $\mathsf{rol}(\mathcal{A}, \mathcal{T})$ to denote the set of roles and their inverses appearing in $\mathcal{A}$ and $\mathcal{T}$, and $\mathsf{obj}(\mathcal{A})$ to denote the set of object names appearing in $\mathcal{A}$. For each object name $a \in \mathsf{obj}(\mathcal{A})$ and role $R \in \mathsf{rol}(\mathcal{A}, \mathcal{T})$, $N_{\mathcal{A}}(a, R)$ denotes the set of *R-neighbours* of a in $\mathcal{A}$, i.e.

$$N_{\mathcal{A}}(a, R) = \{b \in \mathsf{obj}(\mathcal{A}) \mid (a, b) : R \in \mathcal{A} \text{ or } (b, a) : \mathsf{Inv}(R) \in \mathcal{A}\}$$

We use $\mathsf{cl}(\mathcal{A}, \mathcal{T})$ to denote the smallest set containing all sub-concepts of concepts appearing in $\mathcal{A}$ and $\mathcal{T}$ that is closed under $\dot{\neg}$. It can easily be shown that the cardinality of $\mathsf{cl}(\mathcal{A}, \mathcal{T})$ is linear in the sizes of $\mathcal{A}$ and $\mathcal{T}$. The notion of types can straightforwardly be extended to ABoxes.

Definition 16 (Type) *A* type *T for an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ is defined as in Definition 3, where $\mathsf{cl}(C_0, \mathcal{T})$ is replaced with $\mathsf{cl}(\mathcal{A}, \mathcal{T})$.*

The *size* of an ABox assertion $a : C$ is the length of the concept C; the *size* of an ABox assertion $(a, b) : R$ is 1; finally, the *size* of an ABox $\mathcal{A}$ is the sum of the size of all assertions in $\mathcal{A}$. The number of types for an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ is thus clearly exponential in the size of $\mathcal{A}$ and $\mathcal{T}$.

The central notion in the reduction of finite $\mathcal{ALCQI}$-ABox consistency to finite $\mathcal{ALCQI}$-concept satisfiability is that of a reduction candidate:

Definition 17 (Reduction Candidate) *Let $\mathcal{A}$ be an ABox and $\mathcal{T}$ a TBox. A* reduction candidate *for $\mathcal{A}$ and $\mathcal{T}$ is a function t that maps each object name a appearing in $\mathcal{A}$ to a type $t(a)$ for $\mathcal{A}$ and $\mathcal{T}$ such that $a : C \in \mathcal{A}$ implies $C \in t(a)$.*

Let t be a reduction candidate for $\mathcal{A}$ and $\mathcal{T}$. For each object name $a \in \mathsf{obj}(\mathcal{A})$, role $R \in \mathsf{rol}(\mathcal{A}, \mathcal{T})$, and type $T \in \mathsf{ran}(t)$ we use $\#_t^{\mathcal{A}}(a, R, T)$ to denote the number of object names b such that $b \in N_{\mathcal{A}}(a, R)$ and $t(b) = T$.

Now, for each object name $a \in \mathsf{obj}(\mathcal{A})$, we define a reduction concept *$C_t^{\mathcal{A}}(a)$ as follows:*

$$C_t^{\mathcal{A}}(a) := \bigsqcap_{C \in t(a)} C \quad \sqcap \bigsqcap_{\substack{T \in \mathsf{ran}(t) \\ \#_t^{\mathcal{A}}(a,R,T) > 0}} (\geqslant \#_t^{\mathcal{A}}(a, R, T)\ R\ (\bigsqcap_{C \in T} C)).$$

The reduction candidate t is called realisable *iff, for every object name $a \in \mathsf{obj}(\mathcal{A})$, the reduction concept $C_t^{\mathcal{A}}(a)$ is finitely satisfiable w.r.t. $\mathcal{T}$.*

The intuition behind this definition is as follows: for realisable reduction candidates, we can "join" models of the individual reduction concepts to a model of the ABox. Vice versa, each model of the ABox is also a model of all reduction concepts of a realisable reduction candidate.

Note that the definition of reduction concepts exploits the unique name assumption: If we find n different R-neighbours of an object name a in an ABox $\mathcal{A}$ that are all assigned the same type T by the reduction candidate, then the reduction concept $C_t^{\mathcal{A}}(a)$ for a requires (via the atleast restriction) that, for each domain element satisfying it, there are at least n different domain elements of type T that are reachable via the role R. If we drop the unique name assumption, this requirement is too strong since different R-neighbours of a in $\mathcal{A}$ can be interpreted as the same domain element.

The following lemma fixes the relationship between ABoxes and reduction candidates. A proof can be found in Appendix C.

Lemma 18 *Let $\mathcal{A}$ be an ABox and $\mathcal{T}$ a TBox. $\mathcal{A}$ is finitely consistent w.r.t. $\mathcal{T}$ iff there exists a realisable reduction candidate for $\mathcal{A}$ and $\mathcal{T}$.*

It is now easy to establish a tight complexity bound for finite $\mathcal{ALCQI}$-ABox consistency.

Theorem 19 *Finite $\mathcal{ALCQI}$-ABox consistency w.r.t. TBoxes is* ExpTime-*complete if numbers are coded in binary.*

Proof. Let $\mathcal{A}$ be an ABox and $\mathcal{T}$ a TBox. Since the number of types for $\mathcal{A}$ and $\mathcal{T}$ is exponential in the size of $\mathcal{A}$ and $\mathcal{T}$ and the number of object names used in $\mathcal{A}$ is linear in the size of $\mathcal{A}$, the number of reduction candidates for $\mathcal{A}$ and $\mathcal{T}$ is exponential in the size of $\mathcal{A}$ and $\mathcal{T}$. Thus, to decide finite consistency of $\mathcal{A}$ w.r.t. $\mathcal{T}$, we may simply enumerate all reduction candidates for $\mathcal{A}$ and $\mathcal{T}$ and check them for realisability: by Lemma 18, $\mathcal{A}$ is finitely consistent w.r.t. $\mathcal{T}$ if we find a realisable reduction candidate. Since the size of each reduction concept is polynomial in the size of $\mathcal{A}$ and $\mathcal{T}$, by Theorem 14, the resulting algorithm can be executed in deterministic time exponential in $\mathcal{A}$ and $\mathcal{T}$.
$\square$

Note that we make the unique name assumption only to allow for simpler proofs. Indeed, it is not crucial for obtaining an ExpTime upper bound: if we want to decide finite consistency of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ without the unique name assumption, we may use the following approach: enumerate all possible partitionings of the object names used in $\mathcal{A}$. For each partitioning, choose a representative for each partition and then replace each object name with the representative of its partition. Obviously, the ABox $\mathcal{A}$ is finitely consistent w.r.t. $\mathcal{T}$ *without* the unique name assumption if and only if one of the resulting ABoxes is finitely consistent w.r.t. $\mathcal{T}$ *with* the unique name assumption. Since the number of partitionings is exponential in the number of ABox objects, this yields an ExpTime upper bound for finite ABox consistency without the unique name assumption.

6 Related Work

The results presented here are closely related to investigations that have been performed in two different areas: on the one hand, the complexity of finite model reasoning has been investigated for a variety of conceptual database models that can express infinity. For example, in [24], it is shown that finite satisfiability in SERM schemata can be decided in polynomial time, where a SERM schema roughly corresponds to an entity-relationship (ER) schema with cardinality constraints, but without IS-A links between entities or relationships. In [25], an ExpTime upper bound is proved for finite satisfiability of CR models, where CR is the extension of SERM with IS-A links between entities and relationships. In [26], this ExpTime upper bound is extended to the finite satisfiability of CAR models, where CAR provides, in addition, full Boolean operators on classes and relations of arity larger than 2. A last piece of work to be mentioned is [27], where the complexity of a variety of reasoning problems on (several combinations of) integrity constraints on relational databases are investigated, both in unrestricted and in finite models. For the integrity constraints considered (unary inclusion dependencies and functional dependencies), it turns out that validity of implications between (various combinations of) these constraints often depends on whether we consider unrestricted or finite models, but their complexity is mostly the same.

On the other hand, the complexity of finite model reasoning has been investigated for other first order and modal logics. Most prominently, the two variable fragment of first order logic with counting quantifiers (C2) lacks the finite model property, but both reasoning in the unrestricted case and in finite models are decidable [15; 28] and even of the same complexity, namely NExptime-complete; see [28] for the unrestricted case, [19] for reasoning in finite models, [18] for both cases, and [29] for numbers inside counting quantifiers being coded in binary. As mentioned in the introduction, $\mathcal{ALCQI}$ can be polynomially translated into C2, which yields a NExptime upper bound for $\mathcal{ALCQI}$. As we have shown in this paper, neither this bound nor the one that was established in the first decidability result for $\mathcal{ALCQI}$ [17] are tight. Another example to be mentioned here is the full μ-calculus, i.e., the extension of $\mathcal{ALC}$ with fixpoints and inverse roles. Even without any nested fixpoints, this logic lacks the finite model property because, roughly spoken, it allows to express that (i) there exists an infinite R-path, and (ii) R^- is well-founded. These two constraints together are satisfiable only in an infinite, acyclic R-path, and thus only in infinite models. For the $\nu\mu$-fragment of this logic, finite satisfiability has recently been shown to be ExpTime -complete [30], meeting the complexity bounds for the unrestricted case [31].

The common pattern that seems to recur in various cases is that unrestricted and finite model reasoning are often both decidable, and quite often of the same complexity, even though they might ask for different reasoning techniques. An exception to the latter point is the Stellar fragment, a clausal formalism closely related to the two-variable fragment of first order logic with counting quantifiers: in [18], systems of linear equations are used both for reasoning in unrestricted and finite models.

Finally, we would like to point out that, similar to the case of unrestricted model reasoning, the complexity of finite model reasoning is, in many natural cases, insensitive to the coding of numbers in number restrictions. For example, C2 is NExptime-complete logic that is insensitive in this sense,both for unrestricted and finite model reasoning [29]. In this paper, we have given an example for an ExpTime-complete logic for which finite model reasoning is insensitive to the coding of numbers. The corresponding proof for the unrestricted case can be found in [11]. Finally, examples of Pspace-complete logics for which the (only interesting) unrestricted case is insensitive to the coding of numbers can be found in [13].

7 Outlook

In this paper, we have determined finite model reasoning in the description logic $\mathcal{ALCQI}$ to be ExpTime-complete. This shows that reasoning w.r.t. finite models is not harder than reasoning w.r.t. unrestricted models, which is also known to be ExpTime-complete [11]. We hope that, ultimately, this research will lead to the development of finite model reasoning systems that behave equally well as existing DL reasoners performing reasoning w.r.t. unrestricted models such as FaCT and RACER [8; 9]. Note, however, that the current algorithm is *best-case* ExpTime since it constructs an exponentially large system of inequalities. It can thus not be expected to have an acceptable runtime behaviour if implemented in a naive way. Nevertheless, we believe that the use of equation systems and linear programming is indispensable for finite model reasoning in $\mathcal{ALCQI}$. Thus, efforts to obtain efficient reasoners should perhaps concentrate on methods to avoid best-case exponentiality such as on-the-fly construction of equation systems. Moreover, the reductions presented in Section 4 and 5 can also not be expected to exhibit an acceptable run-time behaviour and it would thus be interesting to try to replace them by more "direct" methods.

Another option for future work is the following: while finite $\mathcal{ALCQI}$-concept satisfiability w.r.t. TBoxes is sufficient for reasoning about conceptual database models as described in the introduction, finite $\mathcal{ALCQI}$-ABox consistency is not yet sufficient for deciding the containment of conjunctive queries w.r.t. a given conceptual model—an intermediate reduction step is required. For unrestricted models, this problem was proven to be in 2-ExpTime [21], and it would be interesting to find out whether this blow-up is avoidable, both for the unrestricted and the finite model case.

References

[1] C. Lutz, U. Sattler, L. Tendera, The complexity of finite model reasoning in description logics, in: Proceedings of the 19th Conference on Automated Deduction (CADE-19), Vol. 2741 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2003.

[2] M. Schmidt-Schauß, G. Smolka, Attributive concept descriptions with complements, Artificial Intelligence 48 (1) (1991) 1–26.

[3] K. Schild, A correspondence theory for terminological logics: Preliminary report, in: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91), Sydney, 1991, pp. 466–471.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.

[5] D. Calvanese, M. Lenzerini, D. Nardi, Description logics for conceptual data modeling, in: J. Chomicki, G. Saake (Eds.), Logics for Databases and Information Systems, Kluwer Academic Publisher, 1998, pp. 229–263.

[6] I. Horrocks, P. F. Patel-Schneider, F. van Harmelen, Reviewing the design of DAML+OIL: An ontology language for the semantic web, in: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-02), 2002.
URL `download/2002/AAAI02IHorrocks.pdf`

[7] I. Horrocks, P. F. Patel-Schneider, F. van Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, Journal of Web Semantics 1 (1).

[8] I. Horrocks, Using an Expressive Description Logic: FaCT or Fiction?, in: Proceedings of the Sixth International Conference on the Principles of Knowledge Representation and Reasoning (KR-98), Morgan Kaufmann, Los Altos, 1998.

[9] V. Haarslev, R. Möller, RACER system description, in: Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-01), Vol. 2083 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2001.

[10] E. Franconi, G. Ng, The i.com tool for intelligent conceptual modelling, in: Working Notes of the ECAI2000 Workshop on Knowledge Representation Meets Databases (KRDB2000), CEUR (`http://ceur-ws.org/`), 2000.

[11] G. De Giacomo, M. Lenzerini, Tbox and Abox reasoning in expressive description logics, in: Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR-96), Morgan Kaufmann, Los Altos, 1996, pp. 316–327.

[12] B. Thalheim, Fundamentals of cardinality constraints, in: Proceedings of the Eleventh International Conference on the Entity-Relationship Approach (ER-92), no. 645 in Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 7–23.

[13] S. Tobies, Complexity results and practical algorithms for logics in knowledge representation, Ph.D. thesis, RWTH Aachen, electronically available at `http://www.bth.rwth-aachen.de/ediss/ediss.html` (2001).

[14] J. F. A. K. van Benthem, Modal Logic and Classical Logic, Bibliopolis, Naples, Italy, 1983.

[15] E. Grädel, M. Otto, E. Rosen, Two-variable logic with counting is decidable, in: Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97), 1997, available via `http://speedy.informatik.rwth-aachen.de/WWW/papers.html`.

[16] L. Pacholski, W. Szwast, L. Tendera, Complexity of two-variable logic with counting, in: Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97), 1997.

[17] D. Calvanese, Finite model reasoning in description logics, in: Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR-96), Morgan Kaufmann, Los Altos, 1996.

[18] I. Pratt-Hartmann, Counting quantifiers and the stellar fragment, available at The Mathematics Preprint Server, `www.mathpreprints.com` (2003).

[19] J. Łopuszański, L. Pacholski, Finite satisfibility of two-variables logic with counting, submitted (2003).

[20] M. J. Fischer, R. E. Ladner, Propositional dynamic logic of regular programs, Journal of Computer and System Science 18 (1979) 194–211.

[21] D. Calvanese, G. De Giacomo, M. Lenzerini, On the decidability of query containment under constraints, in: Proceedings of the Seventeenth ACM SIGACT SIGMOD Symposium on Principles of Database Systems (PODS-98), 1998, pp. 149–158.

[22] I. Horrocks, U. Sattler, S. Tessaris, S. Tobies, How to decide query containment under constraints using a description logic, in: A. Voronkov (Ed.), Proceedings of the Seventh International Conference on Logic for Programming and Automated Reasoning (LPAR 2000), no. 1955 in Lecture Notes in Artificial Intelligence, Springer-Verlag, 2000.

[23] D. Calvanese, Unrestricted and finite model reasoning in class-based representation formalisms, Ph.D. thesis, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza" (1996).

[24] M. Lenzerini, P. Nobili, On the satisfiability of dependency constraints in entity-relationship schemata, Information Systems 15 (4) (1990) 453–461.

[25] D. Calvanese, M. Lenzerini, On the interaction between isa and cardinality constraints, IEEE Computer Society Press, 1994, pp. 204–213.

[26] D. Calvanese, M. Lenzerini, Making object-oriented schemas more expressive, in: Proceedings of the Thirteenth ACM SIGACT SIGMOD Symposium on Principles of Database Systems (PODS-94), ACM Press and Addison Wesley, 1994, pp. 243–254.

[27] S. Cosmadakis, P. Kanellakis, M. Vardi, Polynomial-time implication problems for unary inclusion dependencies, Journal of the ACM 37 (1) (1990) 15–46.

[28] L. Pacholski, W. Szwast, L. Tendera, Complexity results for first-order two-variable logic with counting, SIAM Journal of Computing 29 (4) (2000) 1083–1117.

[29] I. Pratt-Hartmann, Complexity of the two-variable fragment with (binary coded) counting quantifiers, submitted.

[30] M. Bojanczyk, Two-way alternating automata and finite models, in: Proceedings of the 29th International Colloquium on Automata, Languages, and Programming, Vol. 2380 of Lecture Notes in Computer Science, Springer-Verlag, 2002.

[31] M. Y. Vardi, Reasoning about the past with two-way automata, in: Proceedings of the 25th International Colloquium on Automata, Languages, and Programming, Vol. 1443 of Lecture Notes in Computer Science, Springer-Verlag, 1998, pp. 628–641.

[32] S. Tobies, The complexity of reasoning with cardinality restrictions and nominals in expressive description logics, Journal of Artificial Intelligence Research 12 (2000) 199–217.

[33] C. H. Papadimitriou, On the complexity of integer programming, Journal of the ACM 28 (2) (1981) 765–769.

[34] A. Schrijver, Theory of linear and integer programming, Wiley, 1986.

A Proofs for Section 3

We first prove Theorem 7 and then Lemma 10.

Theorem 7 A concept C_0 and a TBox $\mathcal{T}$ have a finite pre-model iff C_0 and $\mathcal{T}$ have a finite (standard) model.

Proof. Since the "if" direction is trivial, we concentrate on "only if". Thus, let $\mathcal{I}$ be a finite pre-model for C_0 and $\mathcal{T}$. We use n to denote the maximum multiplicity of edges in $\mathcal{I}$, i.e.

$$n := \max\{R^{\mathcal{I}}(d,e) \mid d,e \in \Delta^{\mathcal{I}} \text{ and } R \text{ used in } C_0 \text{ or } \mathcal{T}\}.$$

Since $\mathcal{I}$ is finite, n is clearly well-defined. Next, define a (standard) interpretation $\mathcal{J}$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}} \times \{0,\ldots,n-1\}$;
- $A^{\mathcal{J}} := A^{\mathcal{I}} \times \{0,\ldots,n-1\}$ for concept names A;
- $R^{\mathcal{J}} := \{((d,i),(e,j)) \mid \exists k < R^{\mathcal{I}}(d,e) : j = i + k \mod n\}$ for role names R.

The following claim clearly implies that $\mathcal{J}$ is a model of C_0 and $\mathcal{T}$ as desired:

Claim: for all $C \in \mathsf{cl}(C_0,\mathcal{T})$ and $d \in \Delta^{\mathcal{I}}$, $d \in C^{\mathcal{I}}$ implies $(d,i) \in C^{\mathcal{J}}$ for all $i \leq n$.

The proof is by induction on the norm $||\cdot||$ of concepts C, which is defined inductively as follows:

$$\begin{aligned}
||A|| := ||\neg A|| &:= 0 \text{ for } A \text{ concept name}\\
||C_1 \sqcap C_2|| := ||C_1 \sqcup C_2|| &:= 1 + ||C_1|| + ||C_2||\\
||(\geqslant n\ R\ D)|| := ||(\leqslant n\ R\ D)|| &:= 1 + ||D||
\end{aligned}$$

The induction start and the Boolean cases are trivial by definition of $\mathcal{J}$ and using the induction hypothesis. Hence we only treat the number restrictions explicitly:

- $C = (\leqslant n\ R\ D)$. Let $d \in C^{\mathcal{I}}$ and fix an $i \in \{0,\ldots,n-1\}$. We have to show that $(d,i) \in C^{\mathcal{J}}$. From the semantics, we obtain

$$\sum_{e \in D^{\mathcal{I}}} R^{\mathcal{I}}(d,e) \leq n \qquad (*)$$

By construction, for each $e \in \Delta^{\mathcal{I}}$ we have that

$$\sharp\{j \in \{0,\ldots,n-1\} \mid ((d,i),(e,j) \in R^{\mathcal{J}}\} = R^{\mathcal{I}}(d,e). \qquad (**)$$

Since we are doing induction on the norm, the induction hypothesis yields that $e \in (\dot{\neg} D)^{\mathcal{I}}$ implies $(e, j) \in (\dot{\neg} D)^{\mathcal{J}}$ for all $e \in \Delta^{\mathcal{I}}$ and $j \leq n$. Together with $(*)$ and $(**)$, this clearly yields that $(d, i) \in C^{\mathcal{J}}$ as desired.

- $C = (\geqslant n\ R\ D)$. Similar to the previous case.

□

Next, we prove the "if" direction of Lemma 10.

Lemma 20 *If C_0 is finitely satisfiable w.r.t. $\mathcal{T}$, then the system of inequalities $\mathcal{E}_{C_0,\mathcal{T}}$ has an admissible solution.*

Proof. Let $\mathcal{I}$ be a finite model of C_0 w.r.t. $\mathcal{T}$. From $\mathcal{I}$, we can construct an admissible solution of $\mathcal{E}_{C_0,\mathcal{T}}$. For $e \in \Delta^{\mathcal{I}}$, we use $t(e)$ to refer to the unique *type* of which e is an instance, and $m(e)$ to refer to the unique *mosaic* of which e is an instance, as has been defined in Definitions 3 and 5, respectively. Moreover, we use $M^{\mathcal{I}}$ to refer to $\{e \in \Delta^{\mathcal{I}} \mid m(e) = M\}$ and $T^{\mathcal{I}}$ to refer to $\{e \in \Delta^{\mathcal{I}} \mid t(e) = T\}$. Next, we set $\hat{x}_M := \#M^{\mathcal{I}}$ and prove the following claim:

Claim: $\{\hat{x}_M \mid M \text{ a mosaic}\}$ is an admissible solution of $\mathcal{E}_{C_0,\mathcal{T}}$.

Equation $(E1)$ is satisfied since $\mathcal{I}$ is a model of C_0: there is some $e_0 \in C_0^{\mathcal{I}}$ implying, by definition of $m(\cdot)$, that we have $\hat{x}_{m(e_0)} \geq 1$ and $C_0 \in T_{m(e_0)}$.

For $(E2)$, let T, T' be types, R a role with $\lim_R(T, T')$ and *not* $\lim_{\mathsf{Inv}(R)}(T', T)$, and fix some $e_M \in M^{\mathcal{I}}$ for each $M^{\mathcal{I}} \neq \emptyset$ as follows:

- if $T_M = T$, choose an $e_M \in M^{\mathcal{I}}$ with a minimal number of R-neighbours in $T'^{\mathcal{I}}$, and
- if $T_M \neq T$, choose an arbitrary $e_M \in M^{\mathcal{I}}$.

We claim that the following (in)equalities hold, which clearly implies (E2).

$$
\begin{aligned}
\sum_{\{M \mid T_M = T\}} L_M(R, T') \cdot \hat{x}_M = & \sum_{\{M \mid T_M = T \wedge M^{\mathcal{I}} \neq \emptyset\}} L_M(R, T') \cdot \hat{x}_M \leq \\
& \sum_{\{M \mid T_M = T \wedge M^{\mathcal{I}} \neq \emptyset\}} \#\{e' \in T'^{\mathcal{I}} \mid \langle e_M, e' \rangle \in R^{\mathcal{I}}\} \cdot \hat{x}_M \leq \\
& \sum_{\{M \mid T_M = T' \wedge M^{\mathcal{I}} \neq \emptyset\}} \#\{e \in T^{\mathcal{I}} \mid \langle e_M, e \rangle \in \mathsf{Inv}(R)^{\mathcal{I}}\} \cdot \hat{x}_M = \\
& \sum_{\{M \mid T_M = T'\}} E_M(\mathsf{Inv}(R), T) \cdot \hat{x}_M
\end{aligned}
$$

The first equality is obvious. The first inequality is due to the definition of m, which implies that, for each instance e of M, $L_M(R, T')$ is a lower bound for the number of e's R-neighbours in $T'^{\mathcal{I}}$. The second inequality holds mainly by a simple graph-theoretic reason: the number ρ of R edges from $T^{\mathcal{I}}$ into $T'^{\mathcal{I}}$ coincides the number of $\mathsf{Inv}(R)$ edges from $T'^{\mathcal{I}}$ into $T^{\mathcal{I}}$. Next, we have chosen e_M with $T_M = T$ to have a minimal number of R-neighbours in $T'^{\mathcal{I}}$, and thus the left-hand term is a lower bound for ρ. Finally, since each $e \in M^{\mathcal{I}}$ with $T_M = T'$ has the same number $E_M(\mathsf{Inv}(R), T)$ of incoming R-edges from T by definition of $M^{\mathcal{I}}$, the right-hand term coincides with ρ, and thus the second inequality holds. Finally, the last equality follows by definition of the sets $M^{\mathcal{I}}$.

Equation $(E3)$ is satisfied with a similar yet simpler argument: let T, T' be types, R a role with $\lim_R(T, T')$ and $\lim_{\mathsf{Inv}(R)}(T', T)$, and fix some $e_M \in M^{\mathcal{I}}$ for each $M^{\mathcal{I}} \neq \emptyset$. Then we have

$$\sum_{\{M \mid T_M = T\}} E_M(R, T') \cdot \hat{x}_M = \sum_{\{M \mid T_M = T \wedge M^{\mathcal{I}} \neq \emptyset\}} \#\{e' \in T'^{\mathcal{I}} \mid \langle e_M, e' \rangle \in R^{\mathcal{I}}\} \cdot \hat{x}_M =$$

$$\sum_{\{M \mid T_M = T' \wedge M^{\mathcal{I}} \neq \emptyset\}} \#\{e \in T^{\mathcal{I}} \mid \langle e_M, e \rangle \in \mathsf{Inv}(R)^{\mathcal{I}}\} \cdot \hat{x}_M =$$

$$\sum_{\{M \mid T_M = T'\}} E_M(\mathsf{Inv}(R), T) \cdot \hat{x}_M$$

using similar arguments as for the (E2) case.

Now for the admissibility of our solution. Obviously, it is a non-negative integer solution. For $(A1)$, consider types T, T' and a role R with $\lim_R(T, T')$, *not* $\lim_{\mathsf{Inv}(R)}(T', T)$, and

$$\sum_{\{M \mid T_M = T'\}} E_M(\mathsf{Inv}(R), T) \cdot \hat{x}_M > 0.$$

Hence there is, by definition of $M^{\mathcal{I}}$, some $\langle e', e \rangle \in \mathsf{Inv}(R)^{\mathcal{I}}$ with $e' \in T'^{\mathcal{I}}$ and $e \in T^{\mathcal{I}}$. Hence we have

$$\sum_{\{M \mid T_M = T\}} \hat{x}_M > 0,$$

and thus $(A1)$ is satisfied.

Finally, for $(A2)$, let M be a mosaic with $\hat{x}_M > 0$, $(\geqslant n R.C) \in T_M$, and

$$m = \sum_{\{T \mid C \in T\}} L_M(R, T) + \sum_{\{T \mid C \in T\}} E_M(R, T) < n.$$

Hence there is some $e_M \in T_M^{\mathcal{I}}$ and $e_1, \ldots, e_n$ with $e_i \neq e_j$ for all $i \neq j$ and, for all $1 \leq i \leq n$, $\langle e_M, e_i \rangle \in R^{\mathcal{I}}$ and $e_i \in C^{\mathcal{I}}$. By definition of $m(e)$, $m < n$ implies that there is some ℓ with $1 \leq \ell \leq n$ such that *not* $\lim_{\mathsf{Inv}(R)}(t(e_M), t(e_\ell))$ and *not* $\lim_R(t(e_\ell), t(e_M))$. Since $C \in t(e_\ell)$, the claim yields

$$\sum_{\substack{\{M' \mid C \in T_{M'},\ \text{not } \lim_R(T_M, T_{M'}),\\ \text{and not } \lim_{\mathsf{Inv}(R)}(T_{M'}, T_M)\}}} \hat{x}_{M'} \geq 1,$$

and $(A2)$ is satisfied. □

We now prove Proposition 11. In the proof, we use the following lemma that was established by Calvanese in [23] and builds on results of Papadimitriou [33].

Lemma 21 [23] *Let $(V, \mathcal{E})$ be a system of $m = \#\mathcal{E}$ linear inequalities in $n = \#V$ variables, in which all coefficients and constants are from the interval $[-a; a]$ of integers. Then, if $(V, \mathcal{E})$ has a solution in $\mathbb{N}^n$, it also has one in $\{0, 1, \ldots, H(V, \mathcal{E})\}^n$, where $H(V, \mathcal{E}) = (n + m)(ma)^{2m+1}$.*

The proof of Proposition 11 is closely related to the proof of Lemma 6.1.5 in [23].

Proposition 11 Let $(V, \mathcal{E})$ be a simple system of inequalities in which all coefficients and constants are from the interval $[-a; a]$ of integers, and let I be a set of side conditions for $(V, \mathcal{E})$. Then the existence of an integer, non-negative, and I-admissible solution for $(V, \mathcal{E})$ can be decided in (deterministic) time polynomial in $\#V + \#\mathcal{E} + \#I + a$.

Proof. For a positive integer k, we use $\mathcal{E}_I(k)$ to denote the set of inequalities

$$\{x \leq k \cdot (x_1 + \cdots + x_j) \mid x > 0 \Longrightarrow x_1 + \cdots + x_j > 0 \in I\}.$$

It is readily checked that every non-negative solution of $(V, \mathcal{E} \cup \mathcal{E}_I(k))$ is a (non-negative and) I-admissible solution of $(V, \mathcal{E})$. We prove the following claim:

Claim: There is an integer $k_\mathcal{E}$ exponential in $\#V + \#\mathcal{E} + \#I$ such that $(V, \mathcal{E})$ admits a non-negative, integer, and I-admissible solution iff $(V, \mathcal{E} \cup \mathcal{E}_I(k_\mathcal{E}))$ admits a non-negative (rational) solution.

Proof: Let $n = \#V$, $m = \#\mathcal{E}$, and $r = \#I$. Then we choose

$$k_\mathcal{E} = a \cdot (2n + m + r)(n + m + r)^{2(n+m+r)+1}.$$

It remains to show that $k_\mathcal{E}$ is as required:

For the "if" direction, let S be a non-negative solution of $(V, \mathcal{E} \cup \mathcal{E}_I(k_\mathcal{E}))$. As noted above, S is also a (non-negative and) I-admissible solution of $(V, \mathcal{E})$. Since all inequalities in $(V, \mathcal{E})$ are positive, we can convert S into an integer solution by multiplying S with the smallest common multiplier of the denominators in S.

Now for the "only if" direction: assume that there exists an integer, non-negative, and I-admissible solution S of $(V, \mathcal{E})$, and let $S(x)$ denote the value S assigns to x. Set

$$\mathcal{E}_S = \{x_1 + \cdots + x_j > 0 \mid x > 0 \Longrightarrow x_1 + \cdots + x_j > 0 \in I \text{ and } S(x) > 0\} \cup \{x = 0 \mid S(x) = 0\}.$$

Obviously, S is also an (integer and non-negative) solution of the system $(V, \mathcal{E} \cup \mathcal{E}_S)$. By Lemma 21, there exists a non-negative integer solution S' of $(V, \mathcal{E} \cup \mathcal{E}_S)$ which is bounded by $h = H(V, \mathcal{E} \cup \mathcal{E}_S)$. It is readily checked that the solution S' is also an (integer and non-negative) solution of $(V, \mathcal{E} \cup \mathcal{E}_I(n))$ for any $n \geq h$. It remains to note that, since $\mathcal{E}_S$ contains at most one inequality for each variable in V and each implication in I, we have $h \leq k_\mathcal{E}$.

In view of the claim just established, it is now easy to show that the existence of a non-negative integer and I-admissible solution for a simple system of inequalities $(V, \mathcal{E})$ and a set of side conditions I can be decided in time polynomial in $\#V + \#\mathcal{E} + \#I + a$: we may clearly view $(V, \mathcal{E} \cup \mathcal{E}_I(k_\mathcal{E}))$ as a linear programming problem. Since $k_\mathcal{E}$ is exponential in $\#V + \#\mathcal{E} + \#I + a$, the binary representation of $k_\mathcal{E}$ is polynomial in $\#V + \#\mathcal{E} + \#I + a$. Thus, the existence of a rational (non-negative) solution for $(V, \mathcal{E} \cup \mathcal{E}_I(k_\mathcal{E}))$ can be checked in (deterministic) time polynomial in $\#V + \#\mathcal{E} + \#I + a$ [34]. □

B Proofs for Section 4

In this section, we prove Lemma 13. For the sake of readability, we split the two directions of this lemma into two separate lemmas. To address individual concept equations of the TBox $\mathsf{Aux}(C, \mathcal{T})$ displayed in Figure 6, throughout this section we will use Ei to refer to the i'th concept equation and E$i.j$ to refer to its j'th line.

Lemma 22 *If $\sigma(C)$ is finitely satisfiable w.r.t. $\sigma(\mathcal{T})$, then C is finitely satisfiable w.r.t. $\mathcal{T}$.*

Proof. The proof strategy is to take a finite model of $\sigma(C)$ and $\sigma(\mathcal{T})$ and transform it into a finite model of C and $\mathcal{T}$. For this purpose, instead of taking an arbitrary model, we first select a special, so-called singular one. We first define the notion of singularity. Let $\mathcal{I}$ be a finite model of $\sigma(C)$ and $\sigma(\mathcal{T})$. For each domain element $d \in \mathsf{Real}^\mathcal{I}$ and each $R \in \mathsf{rol}(C, \mathcal{T})$, we inductively define a sequence of domain elements $h_0^{d,R}, \ldots, h_{\ell_{d,R}}^{d,R}$ as follows:

- set $h_0^{d,R} = d$;

- set $h^{d,R}_{i+1}$ to the L_R-neighbour of $h^{d,R}_i$ (which is unique due to E1.3) if it exists. Otherwise, $\ell_{d,R} = i$.

The constructed sequence is finite due to the use of the $\overline{B_R}$ counter in E2.1, E3.3, and E3.6. Moreover, by E1.2 we have $h^{d,R}_i \in H^{\mathcal{I}}_R$ for $0 < i \leq \ell_{d,R}$, which we will often use (implicitly) throughout the remaining proof. The model $\mathcal{I}$ is called *singular* if, for all roles $R \in \mathsf{rol}(C, \mathcal{T})$ and nodes $d, e \in \mathsf{Real}^{\mathcal{I}}$, we have

$$\#\{(i,j) \mid i \leq \ell_{d,R}, j \leq \ell_{e,\mathsf{Inv}(R)}, \text{ and } (h^{d,R}_i, h^{e,\mathsf{Inv}(R)}_j) \in R^{\mathcal{I}}\} \leq 1.$$

Intuitively, in a singular model, an L_R-path and an $L_{\mathsf{Inv}(R)}$-path are connected via at most one R edge, and thus the operation of contracting L_R edges always results in a simple graph, i.e. no two vertices are connected by more than one edge.

Claim 1. If $\sigma(C)$ is finitely satisfiable w.r.t. $\sigma(\mathcal{T})$, then there is a finite, singular model of $\sigma(C)$ and $\sigma(\mathcal{T})$.

Proof: Let $\mathcal{I}$ be a finite model of $\sigma(C)$ and $\sigma(\mathcal{T})$. Fix an injective mapping δ from $\Delta^{\mathcal{I}}$ to $\{0, \ldots, (\#\Delta^{\mathcal{I}} - 1)\}$. Then we construct a new (finite) interpretation $\mathcal{J}$ by copying $\mathcal{I}$ sufficiently often and "bending R edges" from one copy of $\mathcal{I}$ into others. More precisely, $\mathcal{J}$ is defined as follows:

$$\begin{aligned}
\Delta^{\mathcal{J}} &:= \{\langle d, i\rangle \mid d \in \Delta^{\mathcal{I}} \text{ and } i < \#\Delta^{\mathcal{I}}\};\\
A^{\mathcal{J}} &:= \{\langle d, i\rangle \in \Delta^{\mathcal{J}} \mid d \in A^{\mathcal{I}}\} \text{ for all concept names } A \in \mathsf{cnam}(\sigma(C), \sigma(\mathcal{T}));\\
L^{\mathcal{J}}_R &:= \{(\langle d, i\rangle, \langle e, i\rangle) \in \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}} \mid (d, e) \in L^{\mathcal{I}}_R\}\\
&\qquad \text{for all role names } L_R \text{ with } R \in \mathsf{rol}(C, \mathcal{T});\\
R^{\mathcal{J}} &:= \{(\langle d, i\rangle, \langle e, (\delta(d) + i \mod \#\Delta^{\mathcal{I}})\rangle) \mid (d, e) \in R^{\mathcal{I}}\}\\
&\qquad \text{for all role names } R \text{ appearing in } C \text{ or } \mathcal{T}.
\end{aligned}$$

It is straightforward to check that $\mathcal{J}$ is a singular model of $\sigma(C)$ and $\sigma(\mathcal{T})$, which finishes the proof of Claim 1.

Now let $\mathcal{I}$ be a singular, finite model of $\sigma(C)$ and $\sigma(\mathcal{T})$ and fix, for each $d \in \mathsf{Real}^{\mathcal{I}}$ and $R \in \mathsf{rol}(C, \mathcal{T})$, a sequence of domain elements $h^{d,R}_0, \ldots, h^{d,R}_{\ell_{d,R}}$ as above. We use $\mathcal{I}$ to define an interpretation $\mathcal{J}$ as follows:

$$\begin{aligned}
\Delta^{\mathcal{J}} &:= \mathsf{Real}^{\mathcal{I}}\\
A^{\mathcal{J}} &:= A^{\mathcal{I}} \cap \mathsf{Real}^{\mathcal{I}}\\
R^{\mathcal{J}} &:= \{(d, e) \in \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}} \mid \exists i \leq \ell_{d,R},\ j \leq \ell_{e,\mathsf{Inv}(R)} : (h^{d,R}_i, h^{e,\mathsf{Inv}(R)}_j) \in R^{\mathcal{I}}\}
\end{aligned}$$

It remains to establish the following claim:

Claim 2. For all $d \in \Delta^{\mathcal{J}}$ and $D \in \mathsf{cl}(C, \mathcal{T})$, $d \in \gamma(D)^{\mathcal{I}}$ implies $d \in D^{\mathcal{J}}$.

For assume that Claim 2 is true. Since $\mathcal{I}$ is a model of $\sigma(C)$, by definition of σ there exists a $d \in (\gamma(C) \sqcap \mathsf{Real})^{\mathcal{I}}$. Clearly we have $d \in \Delta^{\mathcal{J}}$ and thus Claim 2 yields $d \in C^{\mathcal{J}}$. Hence, $\mathcal{J}$ is a model of C. By definition of $\sigma(\mathcal{T})$ and the semantics, we have $\mathsf{Real}^{\mathcal{I}} = (\gamma(C_{\mathcal{T}}) \cap \mathsf{Real})^{\mathcal{I}}$. Together with Claim 2 and definition of $\mathcal{J}$, we obtain $\Delta^{\mathcal{J}} = C_{\mathcal{T}}^{\mathcal{J}}$ and thus $\mathcal{J}$ is a model of $\mathcal{T}$.

We prove Claim 2 by induction on the norm $|| \cdot ||$ of concepts D which is defined as in the proof of Theorem 7.

Let $d \in \Delta^{\mathcal{J}} \cap \gamma(D)^{\mathcal{I}}$ for some $D \in \mathsf{cl}(C, \mathcal{T})$. Then $d \in \mathsf{Real}^{\mathcal{I}}$. Since C and $\mathcal{T}$ are in NNF, D is also in NNF. We only treat the interesting cases:

- Let $D = (\geqslant n\ R\ E)$ and $d \in \gamma(D)^{\mathcal{I}} = (X_{(\geqslant n\ R\ E)})^{\mathcal{I}}$. By E1.4 and the choice of the elements $h_0^{d,R}, \ldots, h_{\ell_{d,R}}^{d,R}$, we have $h_i^{d,R} \in (X_{(\geqslant n\ R\ E)})^{\mathcal{I}}$ for $i \leq \ell_{d,R}$. Hence, by exploiting the counter $\overline{B_E}$ and its use in E2.3, E2.5, E4, and E5, it is straightforward to show that there exists a subset $I \subseteq \{1, \ldots, \ell_{d,R}\}$ of cardinality at least n such that, for each $i \in I$, there exists an $e_i \in \Delta^{\mathcal{I}}$ such that $(h_i^{d,R}, e_i) \in R^{\mathcal{I}}$ and $e_i \in \gamma(E)^{\mathcal{I}}$. By E1.1, we have $e_i \in \mathsf{Real}^{\mathcal{I}}$ or $e_i \in H_{\mathsf{Inv}(R)}$ for all $i \in I$. Using the counter $\overline{B_{\mathsf{Inv}(R)}}$ and E3.2 to E3.6, it is thus readily checked that, for each $i \in I$, there exists an $f_i \in \Delta^{\mathcal{I}}$ such that $f_i \in \mathsf{Real}^{\mathcal{I}}$ and e_i can be reached from f_i by repeatedly travelling along $\mathsf{Inv}(R)$-edges. Thus, e_i can be found among the elements $h_0^{f_i,\mathsf{Inv}(R)}, \ldots, h_{\ell_{f_i,\mathsf{Inv}(R)}}^{f_i,\mathsf{Inv}(R)}$. Since $\mathcal{I}$ is singular, it follows that we have $f_i \neq f_j$ for all $i, j \in I$ with $i \neq j$. Moreover, by definition of $\mathcal{J}$ we have $(d, f_i) \in R^{\mathcal{J}}$ for each $i \in I$:
  - · if R is a role name, then this is an immediate consequence of the definition of $\mathcal{J}$;
  - · if $R = S^-$ for some role name S, then $(f_i, d) \in S^{\mathcal{J}}$ by definition of $\mathcal{J}$. The semantics yields $(d, f_i) \in R^{\mathcal{J}}$.

  It thus remains to verify that $f_i \in E^{\mathcal{J}}$ for each $i \in I$. Clearly, $\gamma(E)$ is a Boolean formula over the set of concept names

  $$\mathsf{cnam}(C, \mathcal{T}) \cup \{X_F \mid F = (\bowtie n\ R\ F') \in \mathsf{cl}(C, \mathcal{T})\}.$$

  Since $e_i \in \gamma(E)^{\mathcal{I}}$, E1.4 and E1.5 thus yield $f_i \in \gamma(E)^{\mathcal{I}}$ for each $i \in I$. Since $f_i \in \mathsf{Real}^{\mathcal{I}}$, it remains to apply the induction hypothesis.
- Let $D = (\leqslant n\ R\ E)$ and $d \in \gamma(D)^{\mathcal{I}} = (X_{(\leqslant n\ R\ E)})^{\mathcal{I}}$. Assume that there exists a subset $W \subseteq \Delta^{\mathcal{J}}$ of cardinality greater than n such that, for each $e \in W$, we have $(d, e) \in R^{\mathcal{J}}$ and $e \in E^{\mathcal{J}}$. By definition of $\mathcal{J}$, this implies that, for each $e \in W$, there are $s_e \leq \ell_{d,R}$ and $t_e \leq \ell_{e,R}$ such that $(h_{s_e}^{d,R}, h_{t_e}^{e,\mathsf{Inv}(R)}) \in R^{\mathcal{I}}$:
  - · if R is a role name, then this is an immediate consequence of the definition of $\mathcal{J}$;
  - · if $R = S^-$ for some role name S, then $(d, e) \in R^{\mathcal{I}}$ implies $(e, d) \in S^{\mathcal{I}}$. By definition of $\mathcal{J}$, this means that there are $s_e \leq \ell_{d,R}$ and $t_e \leq \ell_{e,R}$ such that $(h_{t_e}^{e,S}, h_{s_e}^{d,R}) \in S^{\mathcal{I}}$. By semantics and since $S = \mathsf{Inv}(R)$, we obtain

$(h^{d,R}_{s_e}, h^{e,\mathsf{Inv}(R)}_{t_e}) \in R^{\mathcal{I}}$.

We clearly have $W \subseteq \mathsf{Real}^{\mathcal{I}}$. We prove the following three Properties:

(1) $e \neq e'$ implies $h^{d,R}_{s_e} \neq h^{d,R}_{s_{e'}}$ for all $e, e' \in W$. By definition of the $h^{\cdot,\cdot}_i$-sequences of domain elements and E2.2 and E3.2, $e \neq e'$ implies $h^{e,\mathsf{Inv}(R)}_{t_e} \neq h^{e',\mathsf{Inv}(R)}_{t_{e'}}$ for all $e, e' \in W$. Thus, E3.1 yields $h^{d,R}_{s_e} \neq h^{d,R}_{s'_e}$ if $e \neq e'$.

(2) $h^{e,\mathsf{Inv}(R)}_{t_e} \in \gamma(E)^{\mathcal{I}}$ for each $e \in W$. Suppose that $e \notin \gamma(E)^{\mathcal{I}}$. Then $e \in (\neg\gamma(E))^{\mathcal{I}}$ and, by E1.6, $e \in \gamma(\dot\neg E)^{\mathcal{I}}$. Since $e \in \mathsf{Real}^{\mathcal{I}}$ and we are performing induction on the norm of concepts rather than standard structural induction, the induction hypothesis yields $e \in (\dot\neg E)^{\mathcal{J}}$, a contradiction to $e \in E^{\mathcal{J}}$. Thus, $e \in \gamma(E)^{\mathcal{I}}$. Since $\gamma(E)$ is a Boolean formula, it follows from E1.4 and E1.5 that $h^{e,\mathsf{Inv}(R)}_{t_e} \in \gamma(E)^{\mathcal{I}}$.

(3) $s_e \neq 0$ for all $e \in W$. For assume that $s_e = 0$. Then $h^{d,R}_{s_e} = d$. By E2.4 and since $d \in (X_{(\leqslant n\ R\ E)})^{\mathcal{I}}$ and $(d, h^{e,\mathsf{Inv}(R)}_{t_e}) \in R^{\mathcal{I}}$, this yields $h^{e,\mathsf{Inv}(R)}_{t_e} \in (\neg(\gamma(E)))^{\mathcal{I}}$ in contradiction to Property 2.

Properties 1 to 3 imply the existence of a subset $I \subseteq \{1, \ldots, \ell_{d,R}\}$ of cardinality greater than n such that, for each $i \in I$, there exists an $e \in \Delta^{\mathcal{I}}$ with $(h^{d,R}_i, e) \in R^{\mathcal{I}}$ and $e \in \gamma(E)^{\mathcal{I}}$. Exploiting the concept $X_{(\leqslant n\ R\ E)}$ and the counter $\overline{B_E}$ and their use in E1.4, E2.3, E5, and E6, it is readily checked that this is a contradiction to $\mathcal{I}$ being a model of $\mathsf{Aux}(C, \mathcal{T})$.

□

Lemma 23 *If C is finitely satisfiable w.r.t. $\mathcal{T}$, then $\sigma(C)$ is finitely satisfiable w.r.t. $\sigma(\mathcal{T})$.*

Proof. Now for the "only if" direction: let $\mathcal{I}$ be a finite model of C and $\mathcal{T}$. For each $d \in \Delta^{\mathcal{I}}$ and each $R \in \mathsf{rol}(C, \mathcal{T})$, fix a subset $W_{d,R} \subseteq \Delta^{\mathcal{I}}$ of cardinality at most depth_R such that the following conditions are satisfied:

(1) $(d, e) \in R^{\mathcal{I}}$ for all $e \in W_{d,R}$;

(2) for all $(\geqslant n\ R\ D) \in \mathsf{cl}(C, \mathcal{T})$ with $d \in (\geqslant n\ R\ D)^{\mathcal{I}}$, we have

$$\#\{e \in W_{d,R} \mid e \in D^{\mathcal{I}}\} \geq n;$$

(3) for all $(\leqslant n\ R\ D) \in \mathsf{cl}(C, \mathcal{T})$ with $d \in (\leqslant n\ R\ D)^{\mathcal{I}}$, we have

$$\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in R^{\mathcal{I}} \text{ and } e \in D^{\mathcal{I}}\} \subseteq W_{d,R};$$

Using the semantics and the definition of depth_R, it is easy to show that such subsets indeed exist. Next, fix a linear ordering on $W_{d,R}$, i.e., an injective mapping $\nu_{d,R} : W_{d,R} \longrightarrow \{0, \ldots, \#W_{d,R} - 1\}$. We use these mappings to define a finite model $\mathcal{J}$ of $\sigma(C)$ w.r.t. $\sigma(\mathcal{T})$ as follows:

$$
\begin{aligned}
\Delta^{\mathcal{J}} &= \Delta^{\mathcal{I}} \cup \{x_{d,R,e} \mid d \in \Delta^{\mathcal{I}},\ R \in \mathsf{rol}(C,\mathcal{T}),\ \text{and } e \in W_{d,R}\};\\
A^{\mathcal{J}} &= A^{\mathcal{I}} \cup \{x_{d,R,e} \mid d \in A^{\mathcal{I}},\ R \in \mathsf{rol}(C,\mathcal{T}),\ \text{and } e \in W_{d,R}\}\\
&\qquad \text{for all } A \in \mathsf{cnam}(C,\mathcal{T});\\
X^{\mathcal{J}}_{(\bowtie n\ R\ D)} &= (\bowtie n\ R\ D)^{\mathcal{I}} \cup \{x_{d,R,e} \mid d \in (\bowtie n\ R\ D)^{\mathcal{I}} \text{ and } e \in W_{d,R}\}\\
&\qquad \text{for all } (\bowtie n\ R\ D) \in \mathsf{cl}(C,\mathcal{T});\\
\mathsf{Real}^{\mathcal{J}} &= \Delta^{\mathcal{I}};\\
H^{\mathcal{J}}_R &= \{x_{d,R,e} \mid d \in \Delta^{\mathcal{I}} \text{ and } e \in W_{d,R}\} \text{ for all } R \in \mathsf{rol}(C,\mathcal{T});\\
L_R &= \{(d, x_{d,R,e}) \mid d \in \Delta^{\mathcal{I}},\ e \in W_{d,R},\ \text{and } \nu_{d,R}(e) = 0\} \cup\\
&\qquad \{(x_{d,R,e}, x_{d,R,e'}) \mid d \in \Delta^{\mathcal{I}},\ e, e' \in W_{d,R},\ \text{and } \nu_{d,R}(e') = \nu_{d,R}(e) + 1\}\\
&\qquad \text{for all } R \in \mathsf{rol}(C,\mathcal{T});\\
R^{\mathcal{I}} &= \{(x_{d,R,e}, x_{e,R^-,d}) \mid d, e \in \Delta^{\mathcal{I}} \text{ with } e \in W_{d,R} \text{ and } d \in W_{e,R^-}\} \cup\\
&\qquad \{(x_{d,R,e}, e) \mid d, e \in \Delta^{\mathcal{I}} \text{ with } e \in W_{d,R} \text{ and } d \notin W_{e,R^-}\} \cup\\
&\qquad \{(d, x_{e,R^-,d}) \mid d, e \in \Delta^{\mathcal{I}} \text{ with } d \in W_{e,R^-} \text{ and } e \notin W_{d,R}\}\\
&\qquad \text{for all } R \in \mathsf{rnam}(C,\mathcal{T}).
\end{aligned}
$$

for each $R \in \mathsf{rol}(C,\mathcal{T})$, the counter $\overline{B_R}$ is defined as follows: $\overline{B_R} = 0$ for all instances of $\mathsf{Real}^{\mathcal{J}}$; for the instances of $H^{\mathcal{J}}_R$, we define $\overline{B_R}$ as follows:

$$\overline{B_R} = i \text{ for those } x_{d,R,e} \in H^{\mathcal{J}}_R \text{ with } \nu_{d,R}(e) = i;$$

for each concept $D \in \mathsf{cl}(C,\mathcal{T})$ that appears inside a qualifying number restriction $(\bowtie n\ R\ D) \in \mathsf{cl}(C,\mathcal{T})$, the counter $\overline{B_D}$ is defined as follows: $\overline{B_D} = 0$ for all instances of $\mathsf{Real}^{\mathcal{J}}$; for instances $x_{d,R,e}$ of $H^{\mathcal{J}}_R$, we set

$$\overline{B_D} = \#\{e' \in W_{d,R} \mid \nu_{d,R}(e') < \nu_{d,R}(e) \text{ and } e' \in D^{\mathcal{I}}\};$$

Since the translation $\sigma(C)$ of an $\mathcal{ALCQI}$-concept C is a Boolean formula, it is trivial to prove the following claim by structural induction (using the definition of $\mathcal{J}$):

Claim 3. For all $d \in \Delta^{\mathcal{I}}$ and $D \in \mathsf{cl}(C,\mathcal{T})$, $d \in D^{\mathcal{I}}$ implies $d \in \gamma(D)^{\mathcal{J}}$.

Since $\mathcal{I}$ is a model of C, Claim 3 clearly implies that there is a $d \in \Delta^{\mathcal{I}}$ such that $d \in \gamma(C)^{\mathcal{J}}$. By definition of $\mathsf{Real}^{\mathcal{J}}$, we thus have $d \in \sigma(C)^{\mathcal{J}}$ and thus $\mathcal{J}$ is a model of $\sigma(C)$. Moreover, also by Claim 3 $\mathcal{J}$ is a model of the TBox $\{\top \doteq \mathsf{Real} \rightarrow \gamma(C_{\mathcal{T}})\}$. It is tedious but straightforward to verify that $\mathcal{J}$ is also a model of the TBox $\mathsf{Aux}(C,\mathcal{T})$. Hence $\mathcal{J}$ is a model of $\sigma(\mathcal{T})$. □

C Proofs for Section 5

The goal of this section is to prove Lemma 18. Before we do this, we first establish a technical lemma showing that finitely satisfiable reduction concepts have finite models with certain, desirable properties.

Throughout this section, we will identify types T with the conjunction $\bigsqcap_{C \in T} C$ and write, e.g., $d \in T^{\mathcal{I}}$ for $d \in (\bigsqcap_{C \in T} C)^{\mathcal{I}}$.

Lemma 24 *Let $\mathcal{A}$ be an ABox, $\mathcal{T}$ a TBox, t a reduction candidate for $\mathcal{A}$ and $\mathcal{T}$, and a an object name used in $\mathcal{A}$. If the reduction concept $C_t^{\mathcal{A}}(a)$ is finitely satisfiable w.r.t. $\mathcal{T}$, then there exists a finite model $\mathcal{J}$ of $C_t^{\mathcal{A}}(a)$ and $\mathcal{T}$, and some $d \in (C_t^{\mathcal{A}}(a))^{\mathcal{J}}$ such that, for all roles R, $a \in N_{\mathcal{A}}(a, R)$ implies $(d, d) \in R^{\mathcal{J}}$.*

Proof. Let $\mathcal{I}$ be a finite model of $C_t^{\mathcal{A}}(a)$ and $\mathcal{T}$ and let $d \in (C_t^{\mathcal{A}}(a))^{\mathcal{I}}$. By definition of $C_t^{\mathcal{A}}(a)$, we have $d \in t(a)^{\mathcal{I}}$. We construct a new interpretation $\mathcal{J}$ that satisfies the condition given in the lemma. For each role name R with $a \in N_{\mathcal{A}}(a, R)$, fix

(1) a domain element $e_R \in \Delta^{\mathcal{I}}$ with $(d, e_R) \in R^{\mathcal{I}}$ and $e_R \in t(a)^{\mathcal{I}}$;
(2) a domain element $e_{R^-} \in \Delta^{\mathcal{I}}$ with $(d, e_{R^-}) \in (R^-)^{\mathcal{I}}$ and $e_{R^-} \in t(a)^{\mathcal{I}}$.

Such domain elements exist by construction of the reduction concept $C_t^{\mathcal{A}}(a)$, and since $a \in N_{\mathcal{A}}(a, R)$ implies $a \in N_{\mathcal{A}}(a, R^-)$. We construct the new interpretation $\mathcal{J}$ in two steps:

(1) Define a new interpretation $\mathcal{I}'$ as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}'} &= \Delta^{\mathcal{I}} \times \{0,1\}; \\
A^{\mathcal{I}'} &= \{(e,i) \mid e \in A^{\mathcal{I}} \text{ and } i \in \{0,1\}\} \text{ for all concept names } A; \\
R^{\mathcal{I}'} &= \{((e,i),(e',j)) \mid (e,e') \in R^{\mathcal{I}},\ i,j \in \{0,1\},\ \text{and } i \neq j\} \\
&\qquad \text{for all role names } R.
\end{aligned}$$

Using structural induction, it is readily checked that, for each $e \in \Delta^{\mathcal{I}}$ and $C \in \mathsf{cl}(\mathcal{A}, \mathcal{T})$,

$$e \in C^{\mathcal{I}} \text{ implies } (e,i) \in C^{\mathcal{I}'} \text{ for each } i \in \{0,1\}. \qquad (*)$$

Thus we have $(d, 0) \in (C_t^{\mathcal{A}}(a))^{\mathcal{I}'}$, where d is the initially chosen instance of $C_t^{\mathcal{A}}(a)$ (the same holds for $(d, 1)$). From now on, we focus on $(d, 0)$ as the "relevant" instance of $C_t^{\mathcal{A}}(a)$. Clearly, $(*)$ implies that $\mathcal{I}'$ is a model of $\mathcal{T}$.

(2) The interpretation $\mathcal{J}$ is now defined as follows:

$$\begin{aligned}
\Delta^{\mathcal{J}} &= \Delta^{\mathcal{I}'}; \\
A^{\mathcal{J}} &= A^{\mathcal{I}'} \text{ for all concept names } A; \\
R^{\mathcal{J}} &= R^{\mathcal{I}'} \text{ for all role names } R \text{ with } a \notin N_{\mathcal{A}}(a,R); \\
R^{\mathcal{J}} &= (R^{\mathcal{I}'} \setminus \{((d,0),(e_R,1)),((e_{R^-},1),(d,0))\}) \\
&\qquad \cup \{((d,0),(d,0)),((e_{R^-},1),(e_R,1))\} \\
&\qquad \text{for all role names } R \text{ with } a \in N_{\mathcal{A}}(a,R).
\end{aligned}$$

Using structural induction, we may check that, for each $x \in \Delta^{\mathcal{J}}$ and each $C \in \mathsf{cl}(\mathcal{A},\mathcal{T})$,

$$x \in C^{\mathcal{I}'} \text{ implies } x \in C^{\mathcal{J}}. \tag{$**$}$$

Note that we can show $(**)$ despite the different interpretation of the role names R with $a \in N_{\mathcal{A}}(a,R)$, which, intuitively, is due to the following reasons: (i) due to the choice of d, e_R, and e_{R^-} and to Property $(*)$, all of $(d,0)$, $(e_R,1)$ and $(e_{R^-},1)$ have type $t(a)$ in $\mathcal{I}'$. Thus, in constructing $\mathcal{J}$ we only remove and add R-neighbours and R^--neighbours that have type $t(a)$; (ii) we do not change the *number* of R-neighbours or R^--neighbours of type $t(a)$ for any domain element: in particular, by construction of $\mathcal{I}'$ the removed edges really exist in $\mathcal{I}'$, and the newly added edges are really new.

By $(**)$, $(d,0) \in (C_t^{\mathcal{A}}(a))^{\mathcal{J}}$ and $\mathcal{J}$ is a model of $\mathcal{T}$. To prove the lemma, it thus remains to show that, for each role R with $a \in N_{\mathcal{A}}(a,R)$, we have $((d,0),(d,0)) \in R^{\mathcal{J}}$. This is true by definition of $R^{\mathcal{J}}$ if R is a role name. If $R = S^-$ for some role name S, then $a \in N_{\mathcal{A}}(a,R)$ implies that $a \in N_{\mathcal{A}}(a,S)$. Thus $((d,0),(d,0)) \in S^{\mathcal{J}}$ by definition of $\mathcal{J}$. By semantics, we obtain $((d,0),(d,0)) \in R^{\mathcal{J}}$ as required. □

We are now ready to prove Lemma 18.

Lemma 18 Let $\mathcal{A}$ be an ABox and $\mathcal{T}$ a TBox. $\mathcal{A}$ is finitely consistent w.r.t. $\mathcal{T}$ iff there exists a realisable reduction candidate for $\mathcal{A}$ and $\mathcal{T}$.

Proof. The "only if" direction is simple: let $\mathcal{I}$ be a finite model of $\mathcal{A}$ and $\mathcal{T}$. We construct a reduction candidate t as follows:

$$\text{for each object } a \text{ in } \mathcal{A}, \text{ set } t(a) = \{D \in \mathsf{cl}(\mathcal{A},\mathcal{T}) \mid a^{\mathcal{I}} \in D^{\mathcal{I}}\}.$$

Exploiting the unique name assumption, it is then easily checked that, for every object a in $\mathcal{A}$, we have $a^{\mathcal{I}} \in (C_t^{\mathcal{A}}(a))^{\mathcal{I}}$, i.e. $\mathcal{I}$ is a finite model of $C_t^{\mathcal{A}}(a)$ and $\mathcal{T}$. Thus, t is realisable.

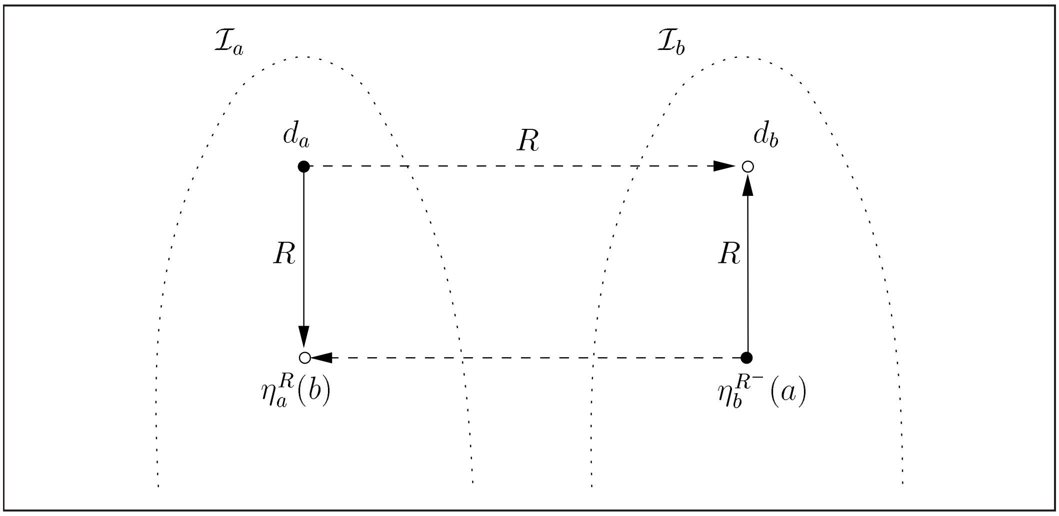

Fig. C.1. Connection of the models $\mathcal{I}_a$ and $\mathcal{I}_b$.

For the "if" direction, assume that there exists a realisable reduction candidate t for $\mathcal{A}$ and $\mathcal{T}$. This implies that, for each object name a used in $\mathcal{A}$, there is a finite model $\mathcal{I}_a$ of $C_t^{\mathcal{A}}(a)$ and $\mathcal{T}$. For each such model $\mathcal{I}_a$, fix a domain element $d_a \in \Delta^{\mathcal{I}_a}$ such that $d_a \in (C_t^{\mathcal{A}}(a))^{\mathcal{I}_a}$. By Lemma 24, we may w.l.o.g. assume that, for all object names a used in $\mathcal{A}$ and all roles R, $a \in N_{\mathcal{A}}(a,R)$ implies $(d_a, d_a) \in R^{\mathcal{I}_a}$. Moreover, we assume w.l.o.g. that $a \neq b$ implies $\Delta^{\mathcal{I}_a} \cap \Delta^{\mathcal{I}_b} = \emptyset$.

In the following, we use the models $\mathcal{I}_a$ to construct a (finite) model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$. First fix, for each object name a used in $\mathcal{A}$ and each role $R \in \mathsf{rol}(\mathcal{A},\mathcal{T})$, an injective function η_a^R from $N_{\mathcal{A}}(a,R)$ to $\Delta^{\mathcal{I}_a}$ such that, for all $b \in N_{\mathcal{A}}(a,R)$, we have the following:

(1) $\eta_a^R(b) \in t(b)^{\mathcal{I}}$;

(2) $(d_a, \eta_a^R(b)) \in R^{\mathcal{I}_a}$;

(3) if $b = a$, then $\eta_a^R(b) = d_a$.

To show that such functions indeed exist, fix an object name a and a role R. It suffices to construct, for each type $T \in \mathsf{ran}(t)$, an injective function $\eta_a^{R,T}$ from $N_{\mathcal{A}}(a,R) \cap \{b \mid t(b) = T\}$ to $\Delta^{\mathcal{I}_a}$ satisfying Properties (1) to (3), and then take the union of these individual functions since Property (1) ensures that the resulting function is still injective. Observe that, for each $T \in \mathsf{ran}(t)$, we can indeed find an injective function $\eta_a^{R,T}$ satisfying Properties (1) to (3) since (i) $C_t^{\mathcal{A}}(a)$ contains the conjunct $(\geqslant \#_t^{\mathcal{A}}(a,R,T)\ R\ (\bigsqcap_{C \in T} C))$, where $\#_t^{\mathcal{A}}(a,R,T)$ obviously is the cardinality of the set $N_{\mathcal{A}}(a,R) \cap \{b \mid t(b) = T\} = \mathsf{dom}(\eta_a^{R,T})$; and (ii) if $a \in N_{\mathcal{A}}(a,R)$, then $(d_a, d_a) \in R^{\mathcal{I}_a}$ by choice of $\mathcal{I}_a$.

Then define the interpretation $\mathcal{I}$ as follows:

$$
\begin{aligned}
\Delta^{\mathcal{I}} &:= \bigcup_{a \in \mathsf{obj}(\mathcal{A})} \Delta^{\mathcal{I}_a};\\
A^{\mathcal{I}} &:= \bigcup_{a \in \mathsf{obj}(\mathcal{A})} A^{\mathcal{I}_a} \text{ for all concept names } A;\\
R^{\mathcal{I}} &:= \begin{array}{l} \bigcup_{a \in \mathsf{obj}(\mathcal{A})} \Big[\Big(R^{\mathcal{I}_a} \setminus \big(\bigcup_{b \in N_{\mathcal{A}}(a,R)} \{(d_a, \eta_a^R(b))\} \cup \bigcup_{b \in N_{\mathcal{A}}(a,R^-)} \{(\eta_a^{R^-}(b), d_a)\}\big)\Big) \\ \qquad \cup \bigcup_{b \in N_{\mathcal{A}}(a,R)} \{(d_a, d_b), (\eta_b^{R^-}(a), \eta_a^R(b))\}\Big] \end{array}\\
&\qquad \text{for all role names } R;\\
a^{\mathcal{I}} &:= d_a \text{ for each object name } a \text{ used in } \mathcal{A}.
\end{aligned}
$$

Note that the interpretation of role names is well-defined: if $b \in N_{\mathcal{A}}(a, R)$, then $a \in N_{\mathcal{A}}(b, R^-)$, and thus $\eta_b^{R^-}(a)$ is defined.

We explain the idea behind the definition of $R^{\mathcal{I}}$ with the help of Figure C.1. Here we consider the connection of two interpretations $\mathcal{I}_a$ and $\mathcal{I}_b$, where a and b are ABox objects such that $b \in N_{\mathcal{A}}(a, R)$ (and thus also $a \in N_{\mathcal{A}}(b, R^-)$). The non-dashed edges are removed from $\mathcal{I}_a$ and $\mathcal{I}_b$ in Line 1 of the definition of $R^{\mathcal{I}}$, and are thus not part of the connected model. To compensate for this, we add the dashed edges to the connected model in Line 2 of the definition of $R^{\mathcal{I}}$. In the figure, all domain elements displayed as filled circles have the same type, and so do all domain elements displayed as non-filled circles (this is due to Property 1 of the $\eta_a^R(b)$ elements). It is thus readily checked that, after the modification, each domain element has the same number of R-neighbours and R^--neighbours of any given type as before.

Special care was taken in the case $a \in N_{\mathcal{A}}(a, R)$: if we had allowed $\eta_a^R(a) \neq d_a$ and $(d_a, d_a) \in R^{\mathcal{I}_a}$, then we would remove the edge between d_a and $\eta_a^R(a)$ in Line 1, but *not* compensate for this removal in Line 2: there, we only "add" an edge from d_a to itself that does already exist in $\mathcal{I}_a$. Clearly, such a modification might decrease the number of R-neighbours of a given type, which we want to avoid. This is the reason why we need Property 3 of the $\eta_a^R(b)$ elements (and Lemma 24, which ensures that setting $\eta_a^R(a) = d_a$ is always possible).

Using these arguments, it is not hard to prove the following claim using structural induction:

Claim: for each object name a used in $\mathcal{A}$, $d \in \Delta^{\mathcal{I}_a}$, and $C \in \mathsf{cl}(\mathcal{A}, \mathcal{T})$, $d \in C^{\mathcal{I}_a}$ implies $d \in C^{\mathcal{I}}$.

Using the claim, it is readily checked that $\mathcal{I}$ is indeed a (finite) model of $\mathcal{A}$ and $\mathcal{T}$:

(1) Let $a : C \in \mathcal{A}$. Then the claim together with $d_a \in (C_t^{\mathcal{A}}(a))^{\mathcal{I}_a}$ yields

$a^{\mathcal{I}} = d_a \in C^{\mathcal{I}}$ since $t(a)$ is a conjunct of $C_t^{\mathcal{A}}(a)$ and $a : C \in \mathcal{A}$ implies $C \in t(a)$.

(2) Let $(a, b) : R \in \mathcal{A}$. Then $b \in N_{\mathcal{A}}(a, R)$. If R is a role name, we thus have $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ by definition of $R^{\mathcal{I}}$ (second line). If $R = S^-$ for some role name S, then we have $a \in N_{\mathcal{A}}(b, S)$. Thus, $(b^{\mathcal{I}}, a^{\mathcal{I}}) \in S^{\mathcal{I}}$ by definition of $\mathcal{I}$, implying $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ by the semantics.

(3) Finally, the claim together with the fact that, for each object name a used in $\mathcal{A}$, $\mathcal{I}_a$ is a model of $\mathcal{T}$ clearly implies that $\mathcal{I}$ is also a model of $\mathcal{T}$.

□