The Complexity of Finite Model Reasoning in Description Logics

Carsten Lutz[1], Ulrike Sattler[1], and Lidia Tendera[2]

[1] Institute for Theoretical Computer Science, TU Dresden, Germany
email: *lastname*@tcs.inf.tu-dresden.de
[2] Institute of Mathematics and Informatics, Opole University, Poland
email: tendera@math.uni.opole.pl

Abstract. We analyze 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 $\mathcal{ALCQI}$.

1 Motivation

Description logics (DLs) are a family of logical formalisms that originated in the field of knowledge representation and are nowadays used in a wide range of applications [1]. Similar to many modal logics (to which DLs are closely related), most description logics enjoy the finite model property (FMP). This is, for example, the case for the basic propositionally closed DL $\mathcal{ALC}$ [12] that is well-known to be a notational variant of the multi-modal logic K [11]: satisfiability of $\mathcal{ALC}$-concepts (the DL equivalent of a formula) w.r.t. finite models coincides with the satisfiability of $\mathcal{ALC}$-concepts w.r.t. arbitrary models [11]. However, there also exist description logics that do not enjoy FMP. One example is the full μ-calculus, i.e., the extension of $\mathcal{ALC}$ with fixpoints and inverse roles (called inverse modalitities in modal logic). For the $\nu\mu$-fragment of this logic, finite satisfiability was shown to be in 2-ExpTime [2] (to the best of our knowledge, a matching lower bound is not yet known), whereas satisfiability in arbitrary models is known to be ExpTime-complete [15]. Another important example is the DL $\mathcal{ALCQI}$ which is obtained from $\mathcal{ALC}$ by adding qualifying number restrictions (corresponding to graded modalities in modal logic), inverse roles, and general TBoxes (roughly corresponding to the universal modality).

The fact that $\mathcal{ALCQI}$ lacks FMP becomes particularly important if we consider this logic's most prominent application, which is reasoning about conceptual database models [4]: if such a model is described by one of the standard formalisms—namely ER diagrams for relational databases and UML diagrams for object-oriented databases—then it can be translated into a DL TBox, i.e. a set of concept equations; afterwards, a description logic reasoner such as FaCT

and RACER can be used to detect inconsistencies and to infer implicit IS-A relationships between entities/classes. This useful and original application has already led to the implementation of tools that provide a GUI for specifying conceptual models, automatize the translation into description logics, and display the information returned by the DL reasoner [8]. However, it is well-known that there exist ER and UML diagrams which are satisfiable only in infinite models, but not in finite ones [13]. Since all available DL reasoning systems are performing reasoning w.r.t. arbitrary (as opposed to finite) models, this means that some inconsistencies and IS-A relationships will not be detected if these reasoners are used for reasoning about conceptual models.

The main reason for existing DL reasoners to perform only reasoning w.r.t. arbitrary models is that finite model reasoning in description logics such as $\mathcal{ALCQI}$ is not yet well-understood. The only known algorithm is presented by Calvanese in [5], where he proves that reasoning in $\mathcal{ALCQI}$ is decidable in 2-ExpTime. *The purpose of this paper is to improve the understanding of finite model reasoning in description logics by establishing tight* ExpTime *complexity bounds for finite model reasoning in the DL* $\mathcal{ALCQI}$.

In Section 3, we develop an algorithm that is capable of deciding *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 equations that can then be solved by linear programming methods. The main difference to Calvanese's approach is that our equation systems talk about different components of models, *mosaics*, which allows us to keep the size of equation systems exponential in the size of the input. In this way, we improve the best-known 2-ExpTime upper bound to a tight ExpTime one.

Since the approach presented in Section 3 presupposes unary coding of the numbers occurring in qualifying number restrictions, in Section 4 we consider finite model reasoning in $\mathcal{ALCQI}$ and numbers coded in binary. We give a polynomial reduction of $\mathcal{ALCQI}$-concept satisfiability w.r.t. TBoxes to the satisfiability of $\mathcal{ALCFI}$-concept satisfiability w.r.t. TBoxes, where $\mathcal{ALCFI}$ is obtained from $\mathcal{ALCQI}$ by allowing only the number 1 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 $\mathcal{ALCQI}$ and numbers coded in binary.

Finally, in Section 5 we consider the finite satisfiabiliy of ABoxes w.r.t. TBoxes. Intuitively, ABoxes describe a particular state of affairs, a "snapshot" of the world. By 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.

This paper is accompanied by a technical report that contains full proofs [10].

2 Preliminaries

We introduce syntax and semantics of $\mathcal{ALCQI}$.

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: (i) each concept name* $A \in \mathbf{C}$ *is an* $\mathcal{ALCQI}$*-concept; and (ii) 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.*

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 $\neg\top$. To avoid roles like $(R^-)^-$, we define a function Inv on roles such that $\mathsf{Inv}(R) = R^-$ if R is a role name, and $\mathsf{Inv}(R) = S$ if $R = S^-$. The fragment $\mathcal{ALCFI}$ of $\mathcal{ALCQI}$ is obtained by admitting only at-most restrictions $(\leqslant n\ R\ C)$ with $n \in \{0, 1\}$ and only at-least 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 which associates, with each concept name* A*, a set* $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ *and, with 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:*

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

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.*

Let us consider a witness for the fact that $\mathcal{ALCQI}$ lacks FMP: the concept $\neg A \sqcap \exists R.A$ is satisfiable w.r.t. the TBox $\{A \doteq \exists R.A \sqcap (\leqslant 1\ R^-\ \top)\}$, but each of its models contains an infinite R-chain.

There exists another important reasoning problem on concepts and TBoxes: *subsumption*. However, since subsumption can be reduced to (un)satisfiability and vice versa, we just note that all complexity bounds obtained in this paper also apply to subsumption.

In what follows, we will only consider TBoxes of the rather simple form $\{\top \doteq C\}$. 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 $\{\top \doteq \bigsqcap_{1 \leq i \leq n}(C_i \leftrightarrow D_i)\}$.

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 generalize this upper bound to binary coding of numbers.

As observed by Calvanese in [5], combinatorics is an important issue when deciding finite satisfiability of $\mathcal{ALCQI}$-concepts. To illustrate this, consider the TBox

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

It should be clear that, in any model of $\mathcal{T}$, there are at least twice as many objects satisfying $B \sqcap (\leqslant 1\ R^-\ A)$ as there are objects satisfying $A \sqcap (\geqslant 2\ R\ B)$. This simple example suggests that (i) *types* (i.e., sets of concepts satisfied by a particular object in a particular model) such as $\{A, (\geqslant 2\ R\ B)\}$ are a natural notion for dealing with finite satisfiability, and (ii) the combinatorics introduced by finite domains can be addressed with inequalities like $2 \cdot x_T \leq x_{T'}$, where the variable x_T describes the number of instance of a type T (e.g. $\{A, (\geqslant 2\ R\ B)\}$), while $x_{T'}$ describes the number of instances of another type T' (e.g. $\{B, (\leqslant 1\ R^-\ A)\}$). These combinatorial constraints are not an issue if infinite domains are admitted: in this case, we can always find a model where all types that have instances at all have the same number of instances, namely countably infinitely many.

Considering the above two points, a first idea to devise a decision procedure for finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes is to translate an input concept and TBox into a system of inequalities with one variable for each type, and then to use existing algorithms to check whether the equation system has a non-negative integer solution. For example, the satisfiability problem of the concept A w.r.t. the TBox $\mathcal{T}$ above can be translated into the two inequalities

$$\sum_{\{T \mid (\geqslant 2\ R\ B) \in T\}} 2 \cdot x_T \leq \sum_{\{T \mid (\leqslant 1\ \mathsf{Inv}(R)\ A) \in T\}} x_T \quad \text{and} \quad \sum_{\{T \mid A \in T\}} x_T > 0$$

where the sums range over all types induced by the input concept A and TBox $\mathcal{T}$. It is not hard to see that any non-negative integer solution to this equation system can be used to construct a finite model for A and $\mathcal{T}$ and vice versa.

Unfortunately, there is a problem with this approach: assume that the input concept and TBox induce types T_1 to T_5 as follows: $(\geqslant 1\ R\ C) \in T_1$, $(\geqslant 1\ R\ D) \in T_2$, $(\leqslant 1\ \mathsf{Inv}(R)\ \top) \in T_3 \cap T_4 \cap T_5$, $C \in T_3 \cap T_4$, and $D \in T_4 \cap T_5$. The translation described above yields the inequalities

$$x_{T_1} \leq x_{T_3} + x_{T_4} \text{ and } x_{T_2} \leq x_{T_4} + x_{T_5},$$

which have $x_{T_1} = x_{T_2} = x_{T_4} = 1$ 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 $(\geqslant 1\ R\ D)$. However, this violates the $(\leqslant 1\ \mathsf{Inv}(R)\ \top)$ concept in T_4.

This example illustrates that "counting types" does not suffice: conflicts may arise if a type containing an at-most restriction (T_4) can be used as a witness for at-least restrictions in more than one type (T_1 and T_2). In such a situation, it is thus necessary to (additionally) fix the types that are actually used as witnesses for at-least restrictions. We achieve this by defining systems of inequalities based on small chunks of models called *mosaics*, rather than based directly on types. Intuitively, a mosaic describes the type of an object and fixes the type of certain "important" witnesses.

Before defining mosaics, we introduce some preliminaries. In the remainder of this paper, we assume concepts (also those appearing inside TBoxes) 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 into an equivalent one in NNF by exhaustively applying de Morgan's rules and the equivalence between $\neg(\leqslant n\ R\ C)$ and $(\geqslant n+1\ R\ C)$, between $\neg(\geqslant (n+1)\ R\ C)$ and $(\leqslant n\ R\ C)$, and between $\neg(\geqslant 0\ R\ C)$ and $\bot$. We use $\dot{\neg}C$ to denote the NNF of $\neg C$. For a concept C_0 and a TBox $\mathcal{T} = \{\top \doteq C_{\mathcal{T}}\}$, $\mathsf{cl}(C_0, \mathcal{T})$ is the smallest set containing all sub-concepts of C_0 and $C_{\mathcal{T}}$ that is closed under $\dot{\neg}$. It can easily be shown that the cardinality of $\mathsf{cl}(C_0, \mathcal{T})$ is linear in the size of C_0 and $\mathcal{T}$. We use $\mathsf{rol}(C_0, \mathcal{T})$ to denote the set of role names R and their inverses R^- occurring in C_0 or $\mathcal{T}$.

Definition 3 (Types and Mosaics). *A* type *T for $C_0, \mathcal{T} = \{\top \doteq C_{\mathcal{T}}\}$ is a set $T \subseteq \mathsf{cl}(C_0, \mathcal{T})$ such that, for each $D, E \in \mathsf{cl}(C_0, \mathcal{T})$, we have*

(T1) *$D \in T$ iff $\dot{\neg}D \notin T$,*
(T2) *if $D \sqcap E \in \mathsf{cl}(C_0, \mathcal{T})$, then $D \sqcap E \in T$ iff $D \in T$ and $E \in T$,*
(T3) *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*
(T4) *$C_{\mathcal{T}} \in T$.*

We use $\mathsf{type}(C_0, \mathcal{T})$ to denote the set of all types over $C_0, \mathcal{T}$. Let T be a type and $\bowtie \in \{\leqslant, \geqslant\}$. Then we use the following abbreviations:

$$\mathsf{max}^{\bowtie}(T) := \max\{n \mid (\bowtie n\ R\ C) \in T\} \quad \text{and} \quad \mathsf{sum}^{\bowtie}(T) := \sum_{(\bowtie n\ R\ C) \in T} n.$$

For types T_1, T_2 and a role R, we write $\lim_R(T_1, T_2)$ (T_2 is a limited ressource *for T_1 w.r.t. R) 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 $n \in \mathbb{N}$. Finally, for a mapping f, we use $\mathsf{ran}(f)$ for the* range *of f.*

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

- *$T_M \in \mathsf{type}(C_0, \mathcal{T})$,*
- *L_M is a function from $\mathsf{rol}(C_0, \mathcal{T}) \times \mathsf{type}(C_0, \mathcal{T})$ to $\mathbb{N}$, and*
- *E_M is a function 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 $\max(\mathsf{ran}(L_M)) \leq \mathsf{max}^{\geqslant}(T_M)$.*

Consider a mosaic M and one of its "instances" d in some interpretation. While T_M is simply the type of d, L_M and E_M are used to describe certain "neighbors" of d, i.e. objects e reachable from d via a role. For a role R, there are three possibilities for the relationship between T_M and T, the type of e:

1. Not $\lim_R(T_M, T)$ and not $\lim_{\mathsf{Inv}(R)}(T, T_M)$. Then d may have an arbitrary number of R-neighbors of type T and every instance of T may have an arbitrary number of $\mathsf{Inv}(R)$-neighbors of type T_M. Intuitively, R-neighbors of type T are "uncritical" and not recorded in the mosaic.
2. $\lim_R(T_M, T)$ and not $\lim_{\mathsf{Inv}(R)}(T, T_M)$. Then d may have an arbitrary number of R-neighbors of type T, but every instance of T may only have a limited number of $\mathsf{Inv}(R)$-neighbors of type T_M. Thus, R-neighbors of type T are a limited ressource and we record in L_M the *minimal* number of R-neighbors of type T that d needs ("L" for "lower bound").
3. $\lim_{\mathsf{Inv}(R)}(T, T_M)$. Then d may only have a limited number of R-neighbors of type T. To prevent the violation of at-most restrictions in T_M, we record the *exact* number of d's R-neighbors of type T in E_M.

(M1) and (M2) ensure that L_M and E_M record information for the "correct" types as described above; (M3) ensures that at-most restrictions are not violated—by definition, this concerns only neighbors with E_M-types; finally, (M4) puts upper bounds on L_M to ensure that there exist only exponentially many mosaics (see below). At-least restrictions are not mentioned in the definition of mosaics and will be treated by the systems of inequalities to be defined later.

Now for the number of mosaics. The cardinality of $\mathsf{type}(C_0, \mathcal{T})$ is exponential in the size of C_0 and $\mathcal{T}$. Next, (M2) and (M3) imply $\#\{(R,T) \mid E_M(R,T) > 0\} \leq \mathsf{sum}^{\leqslant}(T_M)$ and $\max(\mathsf{ran}(E_M)) \leq \mathsf{max}^{\leqslant}(T_M)$. Analogous bounds for L_M are enforced by (M4). Now $\mathsf{max}^{\bowtie}(T)$ and $\mathsf{sum}^{\bowtie}(T)$ are linear in the size of C_0 and $\mathcal{T}$ for $\bowtie \in \{\leqslant, \geqslant\}$ since numbers are coded in unary, and thus the number of mosaics is bounded exponentially in the size of C_0 and $\mathcal{T}$.

We now define an system of inequalities for a concept C_0 and a TBox $\mathcal{T}$.

Definition 4 (Equation System). *For C_0 an $\mathcal{ALCQI}$-concept and $\mathcal{T}$ a TBox, we introduce a variable x_M for each mosaic M over $C_0, \mathcal{T}$ and define the equation system $\mathcal{E}_{C_0,\mathcal{T}}$ by taking (i) the equation*

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

(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 equation

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

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 equation

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

A solution of $\mathcal{E}_{C_0,\mathcal{T}}$ *is* admissible *if it is a non-negative integer solution and satisfies the following 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* $\lim_{\mathsf{Inv}(R)}(T',T)$*,*

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

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

$$x_M > 0, \textit{ and } \sum_{\{T \mid C \in T\}} L_M(R,T) + \sum_{\{T \mid C \in T\}} E_M(R,T) < n,$$

$$\textit{then} \sum_{\substack{\{M' \mid C \in T_{M'},\ \textit{not } \lim_R(T_M,T_{M'}),\\ \textit{and not } \lim_{\mathsf{Inv}(R)}(T_{M'},T_M)\}}} x_{M'} > 0 \tag{A2}$$

While inequality (E1) guarantees the existence of an instance of C_0, inequalities (E2) and (E3) enforce the lower and exact bounds on the number of neighbors as described by L_M and E_M. A special case is treated by condition (A1): in inequality (E2), it may happen that the left-hand side is zero while the right-hand side is non-zero. In this case, there is an instance of a mosaic M' with $T_{M'} = T'$ and $E_M(\mathsf{Inv}(R),T) > 0$ (counted on the right-hand side), but there is no instance of a mosaic M with $T_M = T$ (counted on the left-hand side)—thus we cannot find any neighbors as required by $E_M(\mathsf{Inv}(R),T)$. To cure this defect, condition (A1) ensures that, if the right-hand side of (E2) is non-zero, then there is at least one instance of a mosaic M with $T_M = T$.[1] Finally, (A2) takes care of at-least restrictions in types T_M: if the number of R-neighbors enforced by L_M and E_M is not enough for some $(\geqslant n\, R\, C) \in T_M$, then we make sure that there is at least one instance of a mosaic M' such that $C \in T_{M'}$ and, for instances of M (M'), the number of R-neighbors ($\mathsf{Inv}(R)$-neighbors) that are instances of M' (M) is not limited.[1]

Lemma 1. *C_0 is finitely satisfiable w.r.t. $\mathcal{T}$ iff the equation system $\mathcal{E}_{C_0,\mathcal{T}}$ has an admissible solution.*

Proof sketch: Concerning the only-if direction, it is possible to construct an admissible solution for $\mathcal{E}_{C_0,\mathcal{T}}$ from a model $\mathcal{I}$ of C_0 and $\mathcal{T}$. Intuitively, we associate, with each object $d \in \Delta^{\mathcal{I}}$, a mosaic $M(d)$: $T_{M(d)}$ contains all concepts from $\mathsf{cl}(C_0,\mathcal{T})$ that d is an instance of, and $L_{M(d)}$ and $E_{M(d)}$ are fixed using a choice function on the neighbors of d in $\mathcal{I}$. If necessary, the value of $L_{M(d)}$ is truncated in order to satisfy (M4).

For the if direction, we use an admissible solution δ of $\mathcal{E}_{C_0,\mathcal{T}}$ to construct a model of C_0 and $\mathcal{T}$ in two steps (in [10], these steps are actually merged): initially, each mosaic M has $\delta(M)$ instances and we use the inequalities and side

[1] To see why a single instance suffices, consult the proof sketch of Lemma 1.

conditions to define a relational structure such that (i) all at-most restrictions are satisfied and (ii) each instance of an at-least restriction $(\geqslant n\ R\ C)$ has at least 1 R-neighbor in C. Then, we take P disjoint copies of the initial model (for P the maximum number in C_0 and $\mathcal{T}$) and "bend" edges back and forth between the copies such that no at-most restrictions are violated and all at-least restrictions are satisfied. □

Since the number of mosaics is exponential in the size of C_0 and $\mathcal{T}$, the size of $\mathcal{E}_{C_0,\mathcal{T}}$ and of the admissibility condition is also exponential in the size of C_0 and $\mathcal{T}$. To prove an ExpTime upper bound for the finite satisfiability of $\mathcal{ALCQI}$-concepts, it thus remains to show that the existence of an admissible solution for the equation systems $\mathcal{E}_{C_0,\mathcal{T}}$ can be decided in deterministic polynomial time. Before we actually do this, we first fix some notation.

We assume linear inequalities to be of the form $\Sigma_i c_i x_i \geq b$. 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 using variables from V. Such a system is called *simple* if only non-negative integers occur on the right-hand side of inequalities and all coefficients are (possibly negative) integers. A *side condition* for an inequality system $(V, \mathcal{E})$ is a constraint of the form

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

It is not hard to check that the inequalities (Ei) can be polynomially transformed into simple ones, and that the conditions (Ai) can be transformed into side conditions: (E1) is already simple; each inequality from (E2) can be brought into the form $\sum \ldots - \sum \ldots \geq 0$; each equality from (E3) can be transformed into two inequalities of the same form; each implication from (A1) is transformed into polynomially many side conditions by using a separate side condition for each addend appearing in the premise (this is possible since we are interested in non-negative solutions only), replacing coefficients $E_M(\cdots) > 0$ with 1, and dropping conditions where $E_M(\cdots) = 0$; (A2) is already in the form of a side condition. The proof of the following lemma is by reduction to linear programming and can be found in [10].

Lemma 2. *Let $(V, \mathcal{E})$ be a simple equation system and I a set of side conditions for $(V, \mathcal{E})$. Then the existence of a non-negative integer solution for $(V, \mathcal{E})$ satisfying all constraints from I can be decided in (deterministic) time polynomial in $\#V + \#\mathcal{E} + \#I$.*

Since satisfiability of $\mathcal{ALC}$ w.r.t. TBoxes in arbitrary models is ExpTime-hard [7, 11] and this DL has the finite model property, combining Lemmas 1 and 2 yields the following theorem:

Theorem 1. *Finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes is* ExpTime*-complete if numbers are coded in unary.*

4 Binary Coding of Numbers

If numbers in number restrictions are coded binarily, the ExpTime upper bound from Theorem 1 does no longer apply: in this case, the number of mosaics is

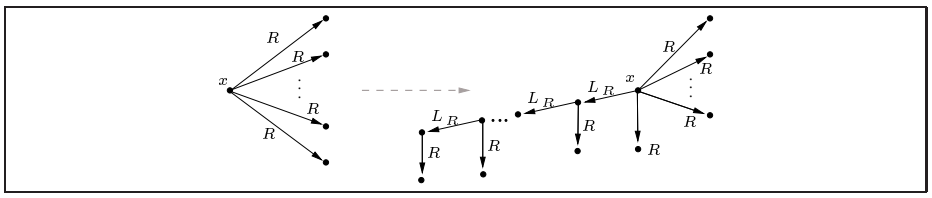

Fig. 1. Representing role successor relationships.

double exponential in the size of the input, and thus the algorithm used in Section 3 yields only a 2-ExpTime upper bound. Indeed, it is a non-trivial problem whether this algorithm can be adapted to binary coding. We have to leave this problem open and choose an alternative technique: a reduction of finite $\mathcal{ALCQI}$-concept satisfiability to the finite satisfiability of $\mathcal{ALCFI}$-concepts. This reduction is polynomial even for binary coding of numbers and its target logic is insensitive to the coding of numbers since it involves only the constants 0,1, and 2. Thus we have an ExpTime upper bound for the finite satisfiability of $\mathcal{ALCQI}$-concepts w.r.t. TBoxes even if numbers are coded in binary since we can first use the translation and then the ExpTime decision procedure from the previous section. Note that, in contrast to existing reductions of $\mathcal{ALCQI}$ to $\mathcal{ALCFI}$ which only work in the case of potentially infinite models [6], we have to take special care to deal with finite models.

The central idea behind our reduction 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 the binary coding of numbers between 0 and n. For the actual counting, we can then use well-known (propositional logic) formulas that encode incrementation. We use a TBox involving auxiliary concept names and roles L_R to re-arrange R-neighbors as shown in Figure 1: except for the root, each node on the auxiliary L_R-path attached to x has precisely one R-neighbor. Ignoring the root for a second, this means that we can count via concept names along the auxiliary objects on L_R-paths. However, we cannot gather all original R-neighbors of x on the L_R-path since we only count up to the sum of numbers occurring in the input concept and TBox. Since an object may have more R-neighbors than this, these "unrestricted" R-neighbors are not re-arranged, but attached to the root as shown in the upper right part of Figure 1.

Fix an $\mathcal{ALCQI}$-concept C_0 and an $\mathcal{ALCQI}$-TBox $\mathcal{T}$ whose finite satisfiability is to be decided. In the following, we use $\mathsf{cnam}(C_0, \mathcal{T})$ to denote the set of concept names appearing in C_0 and $\mathcal{T}$, $\mathsf{rnam}(C_0, \mathcal{T})$ to denote the set of role names appearing in C_0 and $\mathcal{T}$, and $\mathsf{rol}(C_0, \mathcal{T})$ with the same meaning as in Section 3. W.l.o.g., we assume C_0 and $\mathcal{T}$ to be in NNF. In order to translate C_0 and $\mathcal{T}$ to $\mathcal{ALCFI}$, we need to introduce some additional concept and role names:

1. a fresh (i.e., not appearing in C_0 or $\mathcal{T}$) concept name Real;
2. for each $R \in \mathsf{rnam}(C_0, \mathcal{T})$, a fresh concept name H_R and a fresh role name L_R;

3. for each concept $D \in \mathsf{cl}(C_0, \mathcal{T})$ of the form $(\bowtie n\ R\ C)$ (with $\bowtie \in \{\leqslant, \geqslant\}$), a fresh concept name X_D and fresh concept names $B^{\bowtie n}_{C,R,0}, \ldots, B^{\bowtie n}_{C,R,k}$, where $k = \lceil \log(n+1) \rceil$;
4. for each role $R \in \mathsf{rol}(C_0, \mathcal{T})$, fresh concept names $B_{R,0}, \ldots, B_{R,k}$, where $k = \lceil \log(\mathsf{depth}_R) \rceil$ and

$$\mathsf{depth}_R := \sum_{(\bowtie n\ R\ C) \in \mathsf{cl}(C_0, \mathcal{T})} n.$$

The concept name Real is used to distinguish "real" objects from auxiliary objects, and, for each role R, H_R identifies those auxiliary objects that are on an L_R-path. The concept names X_D are used as substitutes for $\mathcal{ALCQI}$'s at-least and at-most restrictions that are not available in $\mathcal{ALCFI}$. Counting with such a restriction $(\bowtie n\ R\ C)$ is replaced by counting via the concept names $B^{\bowtie n}_{C,R,i}$: they count the "occurrences" of R-neighbors in C along L_R-paths. The concept names $B_{R,i}$ are also used for counting, namely to count the length of auxiliary L_R paths.

Note that the number of newly introduced concept and role names is polynomial in the size of C_0 and $\mathcal{T}$. We will use $N^{\bowtie n}_{C,R}$ to refer to the $n$umber encoded by the concept names $B^{\bowtie n}_{C,R,0}, \ldots, B^{\bowtie n}_{C,R,\lceil \log(n+1) \rceil}$ and N_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:

- $(N_R = i)$ to denote the $\mathcal{ALCFI}$-concept (a Boolean formula) expressing that N_R equals i, and similar for $N^{\bowtie n}_{C,R} = i$ and the comparisons "$<$" and "$>$";
- $\mathsf{incr}(N_R, S)$ to denote the $\mathcal{ALCFI}$-concept expressing that, for all S-neighbors, the number N_R is incremented by 1 modulo depth_R, and similar for $\mathsf{incr}(N^{\bowtie n}_{C,R}, S)$. More precisely, these concepts are defined as follows:

$$(B_{R,0} \to \forall S.\neg B_{R,0}) \sqcap (\neg B_{R,0} \to \forall S.B_{R,0}) \sqcap$$

$$\bigsqcap_{k=1..n} \Big(\bigsqcap_{j=0..k-1} B_{R,j} \Big) \to \big((B_{R,k} \to \forall S.\neg B_{R,k}) \sqcap (\neg B_{R,k} \to \forall S.B_{R,k}) \big) \sqcap$$

$$\bigsqcap_{k=1..n} \Big(\bigsqcup_{j=0..k-1} \neg B_{R,j} \Big) \to \big((B_{R,k} \to \forall S.B_{R,k}) \sqcap (\neg B_{R,k} \to \forall S.\neg B_{R,k}) \big).$$

We can now inductively define a translation $\gamma(C_0)$ of the concept C_0 into an $\mathcal{ALCFI}$-concept (indeed even into a Boolean formula):

$$\begin{array}{rr} \gamma(A) := A & \gamma(\neg C) := \neg\gamma(C) \\ \gamma(C \sqcap D) := \gamma(C) \sqcap \gamma(D) & \gamma(C \sqcup D) := \gamma(C) \sqcup \gamma(D) \\ \gamma(\geqslant n\ R\ C) := X_{(\geqslant n\ R\ C)} & \gamma(\leqslant n\ R\ C) := X_{(\leqslant n\ R\ C)} \end{array}$$

Now set $\sigma(C_0) := \gamma(C_0) \sqcap \mathsf{Real}$ and, for $\mathcal{T} = \{\top \doteq C_\mathcal{T}\}$,

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

where the TBox $\mathsf{Aux}(C_0, \mathcal{T})$ is defined in Figure 2, in which we use $C \sqsubseteq D$ as abbreviation for $\top \doteq C \to D$, and in which all $\bigsqcup$ and $\bigsqcap$ range over all concepts

$$
\begin{aligned}
\top \doteq\ & \bigsqcap_{R \in \mathsf{rol}(C_0,\mathcal{T})} \forall R.(\mathsf{Real} \sqcup H_{\mathsf{Inv}(R)}) \sqcap \forall L_R.H_R \sqcap (\leqslant 1\ L_R\ \top) \sqcap \\
& \bigsqcap_{(\bowtie\ n\ R\ D)} \left(X_{(\bowtie\ n\ R\ D)} \leftrightarrow \forall L_R.X_{(\bowtie\ n\ R\ D)}\right) \sqcap \\
& \bigsqcap_{R \in \mathsf{rol}(C_0,\mathcal{T})} \bigsqcap_{A \in \mathsf{cnam}(C_0,\mathcal{T})} (A \leftrightarrow \forall L_R.A) \sqcap \\
& \bigsqcap_{D} \neg\gamma(D) \rightarrow \gamma(\dot{\neg}(D))
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{Real} \sqsubseteq\ & \bigsqcap_{R \in \mathsf{rol}(C_0,\mathcal{T})} \neg H_R \sqcap \forall L_R.(N_R = 0) \sqcap (\leqslant 0\ L_R^-\ \top) \sqcap \\
& \bigsqcap_{(\bowtie\ n\ R\ D)} \left(X_{(\bowtie\ n\ R\ D)} \rightarrow \forall L_R.(N_{D,R}^{\bowtie n} = 0)\right) \sqcap \\
& \bigsqcap_{(\leqslant\ n\ R\ D)} \left(X_{(\leqslant n\ R\ D)} \rightarrow \forall R.\neg\gamma(D)\right) \sqcap \\
& \bigsqcap_{\substack{(\geqslant\ n\ R\ D) \\ \text{with } n>0}} \left(X_{(\geqslant n\ R\ D)} \rightarrow \exists L_R.\top\right)
\end{aligned}
$$

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

$$
H_R \sqsubseteq \bigsqcap_{(\bowtie\ n\ R\ D)} (\exists R.\gamma(D) \rightarrow \mathsf{incr}(N_{D,R}^{\bowtie n}, L_R))
$$

$$
H_R \sqsubseteq \bigsqcap_{(\geqslant n\ R\ D)} (X_{(\geqslant\ n\ R\ D)} \sqcap N_{D,R}^{\geqslant n} < n \sqcap \forall R.\neg\gamma(D) \sqcap \forall L_R.\bot) \rightarrow \bot)
$$

$$
H_R \sqsubseteq \bigsqcap_{(\leqslant\ n\ R\ D)} \left((X_{(\leqslant\ n\ R\ D)} \sqcap N_{D,R}^{\leqslant n} = n \sqcap \exists R.\gamma(D)) \rightarrow \bot\right)
$$

Fig. 2. The TBox $\mathsf{Aux}(C_0, \mathcal{T})$.

in $\mathsf{cl}(C_0, \mathcal{T})$ of the form specified. In what follows, we will use CE i to refer to the i'th concept equation in Figure 2.

CE1, CE2, and CE3 enforce the proper behaviour of the concept names Real and H_R, and of the counting concepts $B_{C,R,i}^{\bowtie n}$ and $B_{R,i}$. CE4 ensures that the counting concepts $B_{C,R}^{\bowtie n}$ are updated correctly along L_R-paths. To guarantee that a "real" element d satisfies a number restriction $X_{(\bowtie\ n\ R\ C)}$, CE5 ensures that we see enough R-neighbors satisfying C along an L_R-path for $\bowtie\ = \geqslant$, whereas CE6 guarantees that we do not see too many such successors for $\bowtie\ = \leqslant$.

Lemma 3. *C_0 is finitely satisfiable w.r.t. $\mathcal{T}$ iff $\sigma(C_0)$ is finitely satisfiable w.r.t. $\sigma(\mathcal{T})$.*

Proof sketch: For the if direction, we take a *singular* finite model $\mathcal{I}$ of $\sigma(C_0)$ and $\sigma(\mathcal{T})$ and transform it into a finite model of C_0 and $\mathcal{T}$ where, intuitively, *singular* means the following: if d and d' are on an L_R-path starting at some $d_0 \in \mathsf{Real}^{\mathcal{I}}$, then there is no e with $(d, e), (d', e) \in R^{\mathcal{I}}$. From a finite model $\mathcal{I}'$ of

C_0 and $\mathcal{T}$, we can construct a singular one by making disjoint copies of $\mathcal{I}'$ and mutually "bending" edges that violate singularity from one copy into another one. The model $\mathcal{J}$ of C_0 and $\mathcal{T}$ is then obtained from $\mathcal{I}$ by keeping only instances of $\mathsf{Rea}|^{\mathcal{I}}$ and adding (d, e) to $R^{\mathcal{J}}$ if e is reachable via an L_R-path and one R-edge from d.

For the only-if direction, we take some finite model $\mathcal{I}$ of C_0 and $\mathcal{T}$ and build a finite model $\mathcal{J}$ of $\sigma(C_0)$ and $\sigma(\mathcal{T})$. For each $d \in \Delta^{\mathcal{I}}$ and $R \in \mathsf{ro}|(C_0, \mathcal{T})$, we fix a subset $W_{d,R} \subseteq \{e \mid (d, e) \in R^{\mathcal{I}}\}$ of cardinality at most depth_R such that (i) $W_{d,R}$ contains at least n witnesses for each $d \in (\geqslant n\ R\ D)^{\mathcal{I}}$, and (ii) if $d \in (\leqslant n\ R\ D)^{\mathcal{I}}$, then every R-neighbor of d in $D^{\mathcal{I}}$ is in $W_{d,R}$ (such sets obviously exist). Then we construct $\mathcal{J}$ by introducing auxiliary objects and, for each set $W_{d,R}$, arranging all the elements of $W_{d,R}$ as R-neigbors of the auxiliary objects on an L_R-path with root d. □

Taking together Theorem 1 and Lemma 3, we obtain the following result:

Theorem 2. *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. We should like to note that (finite) $\mathcal{ALCQI}$-ABox consistency has important applications: whereas finite $\mathcal{ALCQI}$-concept satisfiability algorithms can be used to decide the consistency of conceptual database models and infer implicit IS-A relationships as described in the introduction, $\mathcal{ALCQI}$-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 succesfully been used for and that calls for finite model reasoning [3, 9].

Definition 5. *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 a concept name, 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 *of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. An ABox is called* finitely consistent *if it has a finite model.*

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. For each object name a used in $\mathcal{A}$, $\mathsf{refl}_{\mathcal{A}}(a)$ denotes the set of role names R such that $\{(a,a) : R, (a,a) : R^-\} \cap \mathcal{A} \neq \emptyset$. For each object a and role $R \in \mathsf{rol}(\mathcal{A},\mathcal{T})$, $N_{\mathcal{A}}(a,R)$ denotes the set of object names b such that $b \neq a$ and $\{(a,b) : R,\ (b,a) : \mathsf{Inv}(R)\} \cap \mathcal{A} \neq \emptyset$.

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 be easily shown that the cardinality of $\mathsf{cl}(\mathcal{A},\mathcal{T})$ is linear in the size of $\mathcal{A}$ and $\mathcal{T}$. Moreover, $\mathsf{rol}(\mathcal{A},\mathcal{T})$ denotes the set of all roles (i.e., role names or inverses of role names) used in $\mathcal{A}$ or $\mathcal{T}$.

A *type* T for an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ is defined as in Definition 3 with the only exception that $\mathsf{cl}(C_0,\mathcal{T})$ is replaced with $\mathsf{cl}(\mathcal{A},\mathcal{T})$. In what follows, we will sometimes identify types T with the conjunction $\bigsqcap_{C\in T} C$ and write, e.g., $d \in T^{\mathcal{I}}$ for $d \in (\bigsqcap_{C\in T} C)^{\mathcal{I}}$. It is easily seen that the number of types for an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ is exponential in the size of $\mathcal{A}$ and $\mathcal{T}$.

A central notion for the reduction of finite $\mathcal{ALCQI}$-ABox consistency to finite $\mathcal{ALCQI}$-concept satisfiability is that of a *reduction candidate*: a mapping t that associates a type $t(a)$ with each object name a occurring in $\mathcal{A}$ such that $a : C \in \mathcal{A}$ implies $C \in t(a)$. For each reduction candidate t, object name 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 objects b such that $b \in N_{\mathcal{A}}(a,R)$ and $t(b) = T$. Then, for each object name a used in $\mathcal{A}$, we define its *t-reduction concept* $C_t^{\mathcal{A}}(a)$ as follows:

$$C_t^{\mathcal{A}}(a) := \quad t(a) \sqcap X \sqcap \bigsqcap_{R\in\mathsf{refl}_{\mathcal{A}}(a)} \exists R.(t(a)\sqcap X) \sqcap \bigsqcap_{R\in\mathsf{rol}(\mathcal{A},\mathcal{T})} \bigsqcap_{T\in\mathsf{ran}(t)} (\geqslant \#_t^{\mathcal{A}}(a,R,T)\ R\ (T \sqcap \neg X)),$$

where X is a fresh concept name not used in $\mathcal{A}$ and $\mathcal{T}$. Finally, the reduction candidate t is called *realizable* iff, for every object a used in $\mathcal{A}$, the reduction concept $C_t^{\mathcal{A}}(a)$ is finitely satisfiable w.r.t. $\mathcal{T}$. The following lemma describes the relationship between ABoxes and reduction candidates:

Lemma 4. *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 realizable reduction candidate for $\mathcal{A}$ and $\mathcal{T}$.*

Proof sketch: For the only-if direction, we take a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$. This model gives rise to a reduction candidate t in a straightforward way. By appropriately choosing an extension $X^{\mathcal{I}}$ for the fresh concept name X, we "almost" obtain a model for the reduction concepts $C_t^{\mathcal{A}}(a)$: there may exist object names a such that $a^{\mathcal{I}}$ is an R-neighbor of itself, but $R \notin \mathsf{refl}_{\mathcal{A}}(a)$. Since this interferes with the use of the concept name X, we need to take two disjoint copies of the original model and bend back and forth some edges.

For the if direction, we take a realizable reduction candidate t for $\mathcal{A}$ and $\mathcal{T}$ and finite models $\mathcal{I}_a$ of $C_t^{\mathcal{A}}(a)$ and $\mathcal{T}$, and use these to construct a finite model for $\mathcal{A}$ and $\mathcal{T}$. The general idea is to take the union of (disjoint) finite models for all reduction candidates and then bend some edges to satisfy ABox assertions $(a,b) : R$. Some special care needs to be taken to deal with reflexivity assertions $(a,a) : R$. □

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 realizability: by Lemma 4, $\mathcal{A}$ is finitely consistent w.r.t. $\mathcal{T}$ iff we find a realizable reduction type. Since the size of the reduction concepts is clearly polynomial in the size of $\mathcal{A}$ and $\mathcal{T}$, by Theorem 2 the resulting algorithm can be executed in deterministic time exponential in $\mathcal{A}$ and $\mathcal{T}$.

Theorem 3. *Finite $\mathcal{ALCQI}$-ABox consistency w.r.t. TBoxes is* ExpTime*-complete if numbers are coded in binary.*

Note that our choice of the unique name assumption is *not* crucial for this result: if we want to decide finite consistency of an ABox $\mathcal{A}$ 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 *without* the unique name assumption if and only if any of the resulting ABoxes is finitely consistent *with* the unique name assumption. Clearly, this yields an ExpTime upper bound for finite ABox consistency without the unique name assumption.

6 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. arbitrary models, which is known to be also ExpTime-complete [6]. We hope that, ultimately, this research will lead to the development of finite model reasoning systems that behave equally well as existing DL reasoners doing reasoning w.r.t. arbitrary models. Note, however, that the current algorithm is *best-case* ExpTime since it constructs an exponentially large equation system. 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.

Theoretically, there exist at least two interesting directions in which the presented research can be continued: first, while finite $\mathcal{ALCQI}$-concept satisfiabiltiy w.r.t. TBoxes is sufficient for reasoning about conceptual database models as described in the introduction, finite $\mathcal{ALCQI}$-ABox consistency it is not yet sufficient for deciding the containment of conjunctive queries w.r.t. a given conceptual model—an intermediate reduction step is required. It would thus be

interesting to analyze the complexity of query containment in finite models. We believe that it is possible to obtain an ExpTime upper bound by building on the results presented in Section 5. Secondly, it would be interesting to extend $\mathcal{ALCQI}$ with nominals, i.e. with concept names interpreted as singleton sets. Finite and standard reasoning in the resulting DL $\mathcal{ALCQOI}$ is known to be NExpTime-hard [14]. An extension in this direction is rather challenging since the results established in this paper crucially rely on the fact that adding disjoint copies of a model preserves the model's properties. Unfortunately, in the presence of nominals, this is no longer true.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2002.
2. M. Bojanczyk. Two-way alternating automata and finite models. In *Proc. of ICALP2002*, vol. 2380 of *LNCS*. Springer-Verlag, 2002.
3. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS-98*. ACM Press, 1998.
4. D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In *Logics for Databases and Information Systems*. Kluwer Academic Publisher, 1998.
5. D. Calvanese. Finite model reasoning in description logics. In *Proc. of KR-96*. Morgan Kaufmann, 1996.
6. G. De Giacomo and M. Lenzerini. Tbox and Abox reasoning in expressive description logics. In *Proc. of KR-96*. Morgan Kaufmann, 1996.
7. M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *J. of Computer and System Science*, 18:194–211, 1979.
8. E. Franconi and G. Ng. The i.com tool for intelligent conceptual modelling. In *Working Notes of the ECAI2000 Workshop KRDB2000*. CEUR, 2000.
9. I. Horrocks, U. Sattler, S. Tessaris, and S. Tobies. How to decide query containment under constraints using a description logic. In *Proc. of LPAR 2000*, vol. 1955 in LNAI. Springer-Verlag, 2000.
10. C. Lutz, U. Sattler, and L. Tendera. The complexity of finite model reasoning in description logics. LTCS-Report 02-05, TU Dresden, 2002. Available from http://lat.inf.tu-dresden.de/research/reports.html.
11. K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI-91*. Morgan Kaufmann, 1991.
12. M. Schmidt-Schauß and G. Smolka. Attributive Concept Descriptions with Complements. *Artificial Intelligence*, 48(1), 1991.
13. B. Thalheim. Fundamentals of cardinality constraints. In *Proc. of ER'92*, vol. 645 in LNCS. Springer Verlag, 1992.
14. S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Art. Intelligence Research*, 12:199–217, 2000.
15. M. Y. Vardi. Reasoning about the past with two-way automata. In *Proc. of ICALP'98*, vol. 1443 of *LNCS*. Springer-Verlag, 1998.