EPiC Series in Computing

Volume 50, 2017, Pages 6–19

GCAI 2017. 3rd Global Conference on Artificial Intelligence

Extending the Description Logic $\mathcal{ALC}$ with More Expressive Cardinality Constraints on Concepts*

Franz Baader and Andreas Ecke

Theoretical Computer Science, TU Dresden, Germany
franz.baader@tu-dresden.de

Abstract

We extend the terminological formalism of the well-known description logic $\mathcal{ALC}$ from concept inclusions (CIs) to more general constraints expressed in the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA). In QFBAPA one can formulate Boolean combinations of inclusion constraints and numerical constraints on the cardinalities of sets. Our new formalism extends, on the one hand, so-called cardinality restrictions on concepts, which have been introduced two decades ago, and on the other hand the recently defined statistical knowledge bases. Though considerably more expressive, our formalism has the same complexity (NExpTime) as cardinality restrictions on concepts. We will also introduce a restricted version of our formalism for which the complexity is ExpTime. This yields the until now unknown exact complexity of the consistency problem for statistical knowledge bases.

1 Introduction

Description Logics (DLs) [3] are a well-investigated family of logic-based knowledge representation languages, which are frequently used to formalize ontologies for application domains such as biology and medicine [6]. To define the important notions of such an application domain as formal concepts, DLs state necessary and sufficient conditions for an individual to belong to a concept. These conditions can be Boolean combinations of atomic properties required for the individual (expressed by concept names) or properties that refer to relationships with other individuals and their properties (expressed as role restrictions). For example, the concept of a motor vehicle can be formalized by the concept description $\mathit{Vehicle} \sqcap \exists \mathit{part}.\mathit{Motor}$, which uses the concept names *Vehicle* and *Motor* and the role name *part* as well as the concept constructors conjunction ($\sqcap$) and existential restriction ($\exists r.C$). The concept inclusion (CI)

$$\mathit{Motor\text{-}vehicle} \sqsubseteq \mathit{Vehicle} \sqcap \exists \mathit{part}.\mathit{Motor}$$

can then be used to state that every motor vehicle needs to belong to this concept description. Numerical constraints on the number of role successors (so-called number restrictions) have been

*Partially supported by DFG within the Research Unit 1513 Hybris.

used early on in DLs [5, 8, 7]. For example, using number restrictions, cars can be constrained to being motor vehicles with 3 or 4 wheels:

$$\mathit{Car} \sqsubseteq \mathit{Motor\text{-}vehicle} \sqcap (\leqslant 4\, \mathit{part}.\mathit{Wheel}) \sqcap (\geqslant 3\, \mathit{part}.\mathit{Wheel}).$$

Whereas number restrictions are local in the sense that they consider role successors of an individual under consideration (e.g. a particular motor vehicle), cardinality restrictions on concepts (CRs) [2, 17] are global, i.e., they consider all individuals in an interpretation. For example, the cardinality restriction $(\leqslant 45000000\, (\mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district}))$ states that at most 45 million cars are registered all over Germany. Such cardinality restrictions can express CIs ($C \sqsubseteq D$ is equivalent to $(\leqslant 0\, (C \sqcap \neg D))$), but are considerably more expressive. In particular, they increase the complexity of reasoning: for the prototypical DL $\mathcal{ALC}$, consistency w.r.t. CIs is ExpTime-complete [15], but consistency w.r.t. CRs is NExpTime-complete if the numbers occurring in the CRs are assumed to be encoded in binary [17]. With unary coding of numbers, consistency stays ExpTime-complete even w.r.t. CRs [17], but the above example considering 45 million cars clearly shows that unary coding is not appropriate if numbers with large values are employed.

Whereas CRs can only relate the cardinality of a concept to a fixed number, the recently introduced statistical KBs [11] consist of probabilistic conditional (PCs), which can relate the cardinality of a concept with that of a subconcept. For example, the PC

$$(\exists \mathit{maker}.\mathit{German} \mid \mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district})[.6, .7]$$

states that between 60% and 70% of the cars registered in Germany are German made. Since PCs can also express CIs, the complexity of reasoning in $\mathcal{ALC}$ w.r.t. PCs is ExpTime-hard, but the authors of [11] leave the exact complexity as an open problem.

In this paper, we introduce and investigate more general constraints on the cardinalities of concepts, which encompass both CRs and PCs, but are more expressive than both. The main idea is to use the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) [10] to formulate and combine constraints.[1] An example of a constraint expressible this way, but not expressible using CRs or PCs is $2 \cdot |\mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district} \sqcap \exists \mathit{fuel}.\mathit{Diesel}| < |\mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district} \sqcap \exists \mathit{fuel}.\mathit{Petrol}|$, which states that, in Germany, cars running on petrol outnumber cars running on diesel by a factor of more than two.

We will first show that, though considerably more expressive than CRs, our new formalism has the same complexity (NExpTime for binary coding of numbers) as CRs in the DL $\mathcal{ALC}$. We will then introduce a restricted version of this formalism, which can still express PCs, but not CRs, and show that this way we can reduce the complexity to ExpTime. This also solves the open problem of the exact complexity of reasoning w.r.t. PCs in $\mathcal{ALC}$. The NExpTime upper bound for the general case actually also follows from the NExpTime upper bound in [18] for a more expressive logic with n-ary relations and function symbols. We give our own proof for the case of cardinality constraints on concepts since it is simpler and sets the stage for our ExpTime result in the restricted case.

2 Preliminaries

Before defining our new terminological formalism in Section 3, we briefly introduce $\mathcal{ALC}$ concept inclusions, cardinality constraints on concepts, statistical knowledge bases, and QFBAPA.

[1] The idea of employing QFBAPA in DLs was already used in [1] to generalize number restrictions.

Given disjoint finite sets N_C and N_R of concept names and role names, respectively, the set of $\mathcal{ALC}$ concept descriptions is defined inductively:

- all concept names are $\mathcal{ALC}$ concept descriptions;
- if C, D are $\mathcal{ALC}$ concept descriptions and $r \in N_R$, then $\neg C$ (negation), $C \sqcup D$ (disjunction), $C \sqcap D$ (conjunction), $\exists r.C$ (existential restriction), $\forall r.C$ (value restriction) are $\mathcal{ALC}$ concept descriptions.

An $\mathcal{ALC}$ *concept inclusion (CI)* is of the form $C \sqsubseteq D$ where C, D are $\mathcal{ALC}$ concept descriptions. An $\mathcal{ALC}$ *TBox* is a finite set of $\mathcal{ALC}$ CIs.

The semantics of $\mathcal{ALC}$ is defined using the notion of an interpretation. An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the *domain* $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a function that assigns to every concept name A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every role name r a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This function is extended to $\mathcal{ALC}$ concept descriptions as follows:

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}, (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$;
- $(\exists r.C^{\mathcal{I}}) = \{x \in \Delta^{\mathcal{I}} \mid \text{there is a } y \in \Delta^{\mathcal{I}} \text{ with } (x, y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$;
- $(\forall r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \text{for all } y \in \Delta^{\mathcal{I}}, (x, y) \in r^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$.

The interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all CIs $C \sqsubseteq D \in \mathcal{T}$. The TBox $\mathcal{T}$ is *consistent* if it has a model. The $\mathcal{ALC}$ concept C is *satisfiable* w.r.t. $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. Given two $\mathcal{ALC}$ concept descriptions C, D, we say that C is *subsumed* by D w.r.t. $\mathcal{T}$ (written $C \sqsubseteq_{\mathcal{T}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{T}$. The concept descriptions C, D are equivalent w.r.t. $\mathcal{T}$ (written $C \equiv_{\mathcal{T}} D$) if $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. Equivalence w.r.t. the empty TBox (i.e., $\equiv_\emptyset$) is written as $\equiv$.

An $\mathcal{ALC}$ *cardinality restriction (CR)* is of the form $(\leqslant n\, C)$ or $(\geqslant n\, C)$ where C is an $\mathcal{ALC}$ concept description and n a non-negative integer. An $\mathcal{ALC}$ *CBox* is a finite set of $\mathcal{ALC}$ CRs. Given an interpretation $\mathcal{I}$, we say that $\mathcal{I}$ satisfies the CR $(\leqslant n\, C)$ if $|C^{\mathcal{I}}| \leq n$, and the CR $(\geqslant n\, C)$ if $|C^{\mathcal{I}}| \geq n$, where $|S|$ denotes the cardinality of the set S. The interpretation $\mathcal{I}$ is a *model* of a CBox $\mathcal{C}$ if it satisfies all CRs in $\mathcal{C}$. The CBox $\mathcal{C}$ is *consistent* if it has a model. Satisfiability and subsumption w.r.t. a CBox are defined in the obvious way.

CRs are more expressive than CIs. In fact, it is easy to see that the CR $(\leqslant 0\, (C \sqcap \neg D))$ is satisfied by the same interpretations as the CI $C \sqsubseteq D$. In addition, the CR $(\leqslant 1\, A)$ for a concept name A cannot be expressed by any $\mathcal{ALC}$ TBox. Basically, this is due to the bisimulation invariance of $\mathcal{ALC}$, which implies that any satisfiable concept C can be satisfied in models that make C arbitrarily large (see [4], Section 3). Consistency of $\mathcal{ALC}$ TBoxes is an ExpTime-complete problem [15]. CBoxes raise this complexity to NExpTime-complete if the numbers in the cardinality restrictions are coded in binary [17];[2] for unary coding, it stays ExpTime [17].

A *probabilistic* $\mathcal{ALC}$ *conditional (PC)* is of the form $(C \mid D)[\ell, u]$ where C, D are $\mathcal{ALC}$ concept descriptions and ℓ, u are rational numbers such that $0 \leq \ell \leq u \leq 1$. A *statistical* $\mathcal{ALC}$ *knowledge base* is a finite set of $\mathcal{ALC}$ PCs. To define the semantics of statistical KBs, the authors of [11] restrict the attention to finite interpretations, i.e., interpretations $\mathcal{I}$ where $\Delta^{\mathcal{I}}$ is finite. The finite interpretation $\mathcal{I}$ satisfies the PC $(C \mid D)[\ell, u]$ if $|D^{\mathcal{I}}| = 0$ or

$$\frac{|(C \sqcap D)^{\mathcal{I}}|}{|D^{\mathcal{I}}|} \in [\ell, u],$$

[2] In [17], NExpTime-hardness is actually only stated for $\mathcal{ALCQ}$, but it is easy to see that the argument used there also works for $\mathcal{ALC}$. The NExpTime-upper bound for the case of binary coding follows from the NExpTime upper bound for the two-variable fragment of first order logic with (binary coded) counting quantifiers [13], a result that was not yet known when [17] was published.

and it is a *model* of a statistical KB $\mathcal{S}$ if it satisfies all PCs in $\mathcal{S}$. The statistical KB $\mathcal{S}$ is *consistent* if it has a model. It implies the PC $(C \mid D)[\ell, u]$ (denote as $\mathcal{S} \models (C \mid D)[\ell, u]$) if this PC is satisfied in all models of $\mathcal{S}$. As shown in [11], the PC $(D \mid C)[1, 1]$ is satisfied by the same interpretations as the CI $C \sqsubseteq D$, and thus consistency of statistical $\mathcal{ALC}$ KBs is at least ExpTime-hard. In [11], finding a complexity upper bound for consistency is mentioned as an open problem.

Following the presentation in [1], we now briefly introduce the logic *QFBAPA* (more details can be found in [10]). In this logic one can build *set terms* by applying Boolean operations (intersection $\cap$, union $\cup$, and complement $\cdot^c$) to set variables as well as the constants $\emptyset$ and $\mathcal{U}$. Set terms s, t can then be used to state inclusion and equality constraints ($s = t, s \subseteq t$) between sets. *Presburger Arithmetic (PA) expressions* are built from integer variables, integer constants, and set cardinalities $|s|$ using addition as well as multiplication with an integer constant. They can be used to form numerical constraints of the form $k = \ell, k < \ell, N \,\mathsf{dvd}\, \ell$, where k, ℓ are PA expressions, N is an integer constant, and dvd stands for divisibility. A *QFBAPA formula* is a Boolean combination of set and numerical constraints. A *solution* σ of a QFBAPA formula ϕ assigns a finite set $\sigma(\mathcal{U})$ to $\mathcal{U}$, subsets of $\sigma(\mathcal{U})$ to set variables, and integers to integer variables such that ϕ is satisfied by this assignment. The evaluation of set terms, PA expressions, and set and numerical constraints w.r.t. σ is defined in the obvious way. For example, σ satisfies the numerical constraint $|s \cup t| = |s| + |t|$ for set variables s, t if $|\sigma(s) \cup \sigma(t)| = |\sigma(s)| + |\sigma(t)|$. Note that this is the case iff $\sigma(s)$ and $\sigma(t)$ are disjoint, which we could also have expressed using the set constraint $s \cap t \subseteq \emptyset$. A QFBAPA formula ϕ is *satisfiable* if it has a solution. In [10] it is shown that the satisfiability problem for QFBAPA formulae is NP-complete.

3 Extended Cardinality Constraints on Concepts

Basically, the more expressive constraints considered in this paper are QFBAPA formulae where $\mathcal{ALC}$ concept descriptions are used in place of set variables. However, since inclusion constraints $s \subseteq t$ can be expressed as cardinality constraints $|s \cap t^c| \leq 0$, we do not explicitly allow the use of set constraints. In addition, since $\mathcal{ALC}$ has all Boolean operations on concepts, we do not need Boolean operations on set terms. Consequently, we define *extended cardinality constraints on $\mathcal{ALC}$ concepts* as follows:

- *$\mathcal{ALC}$ cardinality terms* are built from integer constants and concept cardinalities $|C|$ for $\mathcal{ALC}$ concept descriptions C using addition and multiplication with integer constants;
- *extended $\mathcal{ALC}$ cardinality constraints* are of the form $k = \ell, k < \ell, N \,\mathsf{dvd}\, \ell$, where k, ℓ are $\mathcal{ALC}$ cardinality terms and N is an integer constant;
- an *extended $\mathcal{ALC}$ cardinality box (ECBox)* is a Boolean combination of extended $\mathcal{ALC}$ cardinality constraints.

When defining the semantics of ECBoxes, we restrict the attention to *finite* interpretations to ensure that cardinalities of concept descriptions are always well-defined non-negative integers. Given a finite interpretation $\mathcal{I}$, concept cardinalities are interpreted in the obvious way, i.e., $|C|^{\mathcal{I}} := |C^{\mathcal{I}}|$. Addition and multiplication in cardinality terms are interpreted as the usual addition and multiplication operations on integers, and the same is is the case for the comparison operators $=$, $<$, and divisibility dvd. This yields the semantics of extended $\mathcal{ALC}$ cardinality constraints and thus also of their Boolean combinations. The finite interpretation $\mathcal{I}$ is a *model* of an ECBox $\mathcal{E}$ if it satisfies the Boolean formula $\mathcal{E}$ according to the above semantics. The

ECBox $\mathcal{E}$ is *consistent* if it has a model. Satisfiability and subsumption w.r.t. an ECBox are defined in the obvious way. In the presence of ECBoxes, satisfiability and subsumption can actually be expressed by adding constraints to the given ECBox. In fact, let $\mathcal{E}$ be an ECBox and C, D be $\mathcal{ALC}$ concept descriptions. Then we have the following:

$$\begin{aligned} C \text{ is satisfiable w.r.t. } \mathcal{E} \quad &\text{iff} \quad \mathcal{E} \wedge |C| > 0 \text{ is consistent;} \\ C \sqsubseteq_{\mathcal{E}} D \quad &\text{iff} \quad \mathcal{E} \wedge |C \sqcap \neg D| > 0 \text{ is inconsistent.} \end{aligned}$$

For this reason, we will concentrate on the consistency problem in the following.

Obviously, cardinality restrictions can be expressed by extended cardinality constraints. The restriction $(\geqslant n\, C)$ has the same models as the constraint $|C| \geq n$, which can be seen as an abbreviation for $|C| > n \vee |C| = n$. In the same way, $(\leqslant n\, C)$ can be expressed as $|C| \leq n$. This shows that the complexity lower bound of NExpTime for the consistency problem of CBoxes with binary coding of numbers transfers to ECBoxes.[3] Regarding unary coding of numbers, note that QFBAPA is powerful enough to express any number n using a formula of size $log(n)$ by repeated addition of 1 and multiplication with 2. Consequently, binary coding of numbers can be simulated also in the unary case. This shows that the NExpTime lower bound in [17] for CBoxes with binary coding of numbers also transfers for $\mathcal{ALC}$ ECBoxes with *unary* coding, in spite of the fact that consistency of CBoxes is in ExpTime for unary coding of numbers.

Proposition 1. *Consistency of ECBoxes is NExpTime-hard in $\mathcal{ALC}$ both for unary and binary encoding of numbers.*

We will prove in the next section that this problem is actually also in NExpTime. But first, we show that statistical $\mathcal{ALC}$ KBs can be expressed using ECBoxes.

Lemma 2. *Let $\ell = \ell_1/\ell_2$ and $u = u_1/u_2$ for integers ℓ_1, ℓ_2, u_1, u_2. Then $(C \mid D)[\ell, u]$ has the same models as the ECBox*

$$\ell_2 \cdot |C \sqcap D| \geq \ell_1 \cdot |D| \wedge u_2 \cdot |C \sqcap D| \leq u_1 \cdot |D|.$$

Note that the special case $|D^{\mathcal{I}}| = 0$ in the definition of the semantics of PCs is also covered by this lemma since then both the PC $(C \mid D)[\ell, u]$ and the cardinality constraints stated in the lemma are satisfied. Lemma 2 shows that an upper bound for the complexity of consistency of ECBoxes also yields an upper bound for the consistency of statistical KBs. Similarly, implication of PCs by statistical KBs can be reduced to inconsistency of ECBoxes. We will see later, however, that we can reduce these problems to (in)consistency w.r.t. restricted forms of ECBoxes, which provides us with better complexity upper bounds.

4 Consistency of $\mathcal{ALC}$ ECBoxes

In the following we consider an ECBox $\mathcal{E}$ and show how to test $\mathcal{E}$ for consistency by reducing this problem to the problem of testing satisfiability of QFBAPA formulae. Since the reduction is exponential and satisfiability in QFBAPA is in NP, this yields an NExpTime upper bound for consistency of $\mathcal{ALC}$ ECBoxes. To simplify the presentation of our algorithm, we assume that all the concept descriptions occurring in $\mathcal{E}$ are built using the constructors conjunction, negation,

[3] Actually, NExpTime-hardness of consistency was shown for $\mathcal{ALC}$ CBoxes without a restriction to finite interpretations. However, the reduction used in [17] actually produces a CBox that has only finite models, and thus this reduction also works for the case of finite models.

and existential restriction only. This is without loss of generality due to the equivalences $\forall r.C \equiv \neg\exists r.\neg C$ and $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$.

A well-known approach for proving complexity upper bounds in Description Logics and Modal Logics is based on the notion of types [12, 14].[4] Basically, given a set of concept descriptions $\mathcal{M}$, the *type* of an individual in an interpretation consists of the elements of $\mathcal{M}$ to which the individual belongs. Such a type t can also be seen as a concept description C_t, which is the conjunction of all the elements of t. We assume in the following, that $\mathcal{M}$ contains *all subdescriptions* of the concept descriptions occurring in $\mathcal{E}$ as well as the *negations of these subdescriptions*.

Definition 3. *A subset t of $\mathcal{M}$ is a* type *for $\mathcal{E}$ if it satisfies the following properties:*

1. *for every concept description $\neg C \in \mathcal{M}$, either C or $\neg C$ belongs to t;*
2. *for every concept description $C \sqcap D \in \mathcal{M}$, we have that $C \sqcap D \in t$ iff $C \in t$ and $D \in t$.*

We denote the set of all types for $\mathcal{E}$ with types($\mathcal{E}$). Given an interpretation $\mathcal{I}$ and an individual $d \in \Delta^{\mathcal{I}}$, the type of d *is the set*

$$t_{\mathcal{I}}(d) := \{C \in \mathcal{M} \mid d \in C^{\mathcal{I}}\}.$$

It is easy to show that the type of an individual really satisfies the conditions stated in Definition 3. Due to Condition 1 in the definition of types, concept descriptions induced by different types are disjoint, and all concept descriptions in $\mathcal{M}$ can be obtained as the disjoint union of the concept descriptions induced by the types containing them.

Lemma 4. *If $t \neq t'$ are two different types, then C_t and $C_{t'}$ are* disjoint*, i.e., $C_t^{\mathcal{I}} \cap C_{t'}^{\mathcal{I}} = \emptyset$ for all interpretations $\mathcal{I}$. In addition, all concept descriptions C in $\mathcal{M}$ can be expressed as disjunctions of such disjoint type concepts:*

$$C \equiv \bigsqcup_{t \text{ type with } C \in t} C_t \quad \text{and} \quad |C^{\mathcal{I}}| = \sum_{t \text{ type with } C \in t} |C_t^{\mathcal{I}}| \quad \text{for all finite interpretations } \mathcal{I}.$$

We transform the ECBox $\mathcal{E}$ into a QFBAPA formula $\phi_{\mathcal{E}}$ by introducing an integer variable v_t for every type t, stating that these variables have a non-negative value, and then replacing every concept cardinality $|C|$ in $\mathcal{E}$ by the sum of the corresponding type variables, i.e.,

$$|C| \text{ is replaced by } \sum_{t \text{ type with } C \in t} v_t.$$

A model of $\mathcal{E}$ yields a solution of $\phi_{\mathcal{E}}$ as follows:

Lemma 5. *Assume that $\mathcal{I}$ is a finite interpretation that is a model of $\mathcal{E}$. If we define $\sigma(v_t) := |C_t^{\mathcal{I}}|$ for all types t, then σ is a solution of $\phi_{\mathcal{E}}$.*

Proof. This is an immediate consequence of Lemma 4 and the fact that the cardinalities of type concepts are obviously non-negative integers. □

However, not every solution of $\phi_{\mathcal{E}}$ is induced by a model of $\mathcal{E}$ in this way. For example, let

$$\mathcal{E} := |\exists r.A| > 0 \wedge |A| = 0.$$

[4] Note that types are also closely related to the Venn regions used in [10] to show that satisfiability of QFBAPA formulae is in NP.

In this case, the set $\mathcal{M}$ consists of the concept descriptions $A, \neg A, \exists r.A, \neg\exists r.A$, and there are four types:

$$t_1 := \{A, \exists r.A\}, \quad t_2 := \{\neg A, \exists r.A\}, \quad t_3 := \{A, \neg\exists r.A\}, \quad t_4 := \{\neg A, \neg\exists r.A\}.$$

The ECBox $\mathcal{E}$ is now translated into the QFBAPA formula $\phi_\mathcal{E}$ by replacing $|\exists r.A|$ with $v_{t_1} + v_{t_2}$ and $|A|$ with $v_{t_1} + v_{t_3}$ and adding the information that $v_{t_1}, v_{t_2}, v_{t_3}, v_{t_4}$ have values ≥ 0, i.e.,

$$\phi_\mathcal{E} = v_{t_1} \geq 0 \wedge v_{t_2} \geq 0 \wedge v_{t_3} \geq 0 \wedge v_{t_4} \geq 0 \wedge v_{t_1} + v_{t_2} > 0 \wedge v_{t_1} + v_{t_3} = 0.$$

If we set $\sigma(v_{t_1}) = \sigma(v_{t_3}) = \sigma(v_{t_4}) = 0$ and $\sigma(v_{t_2}) = 1$, then σ is a solution of $\phi_\mathcal{E}$. However, $\mathcal{E}$ does not have a model. In fact, $\mathcal{E}$ requires that there is an element belonging to the concept $\exists r.A$, which implies that there also must be an element belonging to A, which is however prohibited by $\mathcal{E}$.

This example shows that we must take elements required by existential restrictions into account. Basically, if a type t is *realized* in the sense that the variable v_t receives a value > 0, then we must ensure that also types required by the existential restrictions in t are realized.

Definition 6. *Let t be a type, $\exists r.C$ an existential restriction contained in t, and $\neg\exists r.C_1, \ldots, \neg\exists r.C_\ell$ all the negated existential restrictions for the role r contained in t. A type t'* satisfies *$\exists r.C$ in t if it has $C, \neg C_1, \ldots, \neg C_\ell$ as elements.*

If a type is realized by a solution of $\phi_\mathcal{E}$, then for every existential restriction $\exists r.C$ contained in t a type that satisfies this existential restriction in t must be realized as well. Thus, we must conjoin to $\phi_\mathcal{E}$ the following formulae: for every type t and every existential restriction $\exists r.C$ contained in t the formula

$$v_t = 0 \vee \sum_{t' \text{ satisfies } \exists r.C \text{ in } t} v_{t'} > 0. \tag{1}$$

Let us call the resulting formula $\psi_\mathcal{E}$.

Lemma 7. *The QFBAPA formula $\psi_\mathcal{E}$ is of size at most exponential in the size of $\mathcal{E}$, and it is satisfiable iff $\mathcal{E}$ is consistent.*

Proof. Regarding the size of $\psi_\mathcal{E}$, first note that there are at most exponentially many types t and thus at most exponentially many variables v_t since the cardinality of $\mathcal{M}$ is linear in the size of $\mathcal{E}$. Consequently, concept cardinalities $|C|$ in $\mathcal{E}$ are replaced by sums of at most exponentially many variables, and $\phi_\mathcal{E}$ contains at most exponentially many inequalities of the form $v_t \geq 0$. This shows that he size of $\phi_\mathcal{E}$ is exponentially bounded. Finally, $\psi_\mathcal{E}$ contains exponentially many conjuncts of the form (1), and each such conjunct has at most exponential size.

Next, assume that the finite interpretation $\mathcal{I}$ is a model of $\mathcal{E}$. If we define σ as in Lemma 5, then by this lemma we know that σ solves $\phi_\mathcal{E}$. Now, consider a conjunct of the form (1) in $\psi_\mathcal{E}$ for the existential restriction $\exists r.C$. If $\sigma(v_t) = 0$, then this conjunct is obviously satisfied. Otherwise, there is an individual $d \in \Delta^\mathcal{I}$ such that $d \in C_t^\mathcal{I}$. In particular, if $\neg\exists r.C_1, \ldots, \neg\exists r.C_\ell$ are all the negated existential restrictions for the role r contained in t, then

$$d \in (\exists r.C \sqcap \neg\exists r.C_1 \sqcap \ldots \sqcap \neg\exists r.C_\ell)^\mathcal{I}.$$

Consequently, there is an individual $d' \in \Delta^\mathcal{I}$ such that $d' \in (C \sqcap \neg C_1 \sqcap \ldots \sqcap \neg C_\ell)^\mathcal{I}$. Let $t' := t_\mathcal{I}(d')$. Then t' satisfies $\exists r.C$ in t, and $\sigma(v_{t'}) > 0$ since $d' \in C_{t'}^\mathcal{I}$. This shows that σ solves all the conjuncts of the form (1), and thus that it solves $\psi_\mathcal{E}$.

Conversely, assume that there is a solution σ of $\psi_{\mathcal{E}}$. Let

$$T_\sigma := \{t \mid t \text{ type with } \sigma(v_t) \neq 0\}$$

be the types that are realized by σ. We now define a finite interpretation $\mathcal{I}$ and show that it is a model of $\mathcal{E}$. The interpretation domain consists of copies of the realized types, where the number of copies is determined by σ:

$$\Delta^{\mathcal{I}} := \{(t, j) \mid t \in T_\sigma \text{ and } 1 \leq j \leq \sigma(v_t)\}.$$

For concept names A we define $A^{\mathcal{I}} := \{(t, j) \in \Delta^{\mathcal{I}} \mid A \in t\}$, and for role names r we set

$$r^{\mathcal{I}} := \{((t, j), (t', j')) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid t' \text{ satisfies an existential restriction for } r \text{ in } t\}.$$

We want to show that the following holds for all types $t \in T_\sigma$ and $j, 1 \leq j \leq \sigma(v_t)$:

$$(t, j) \in C_t^{\mathcal{I}}. \tag{2}$$

Note that, due to the disjointness of the type concepts, this implies that (t, j) cannot be an element of $C_{t'}^{\mathcal{I}}$ for any type $t' \neq t$. As an easy consequence we obtain that $|C_t^{\mathcal{I}}| = \sigma(v_t)$ for all types t. Thus, the fact that σ solves $\psi_{\mathcal{E}}$ implies that $\mathcal{I}$ is a model of $\mathcal{E}$.

It remains to show (2). For this it is sufficient to show the following: for all concept descriptions $C \in \mathcal{M}$, all types $t \in T_\sigma$ and all $j, 1 \leq j \leq \sigma(v_t)$ we have

$$(t, j) \in C^{\mathcal{I}} \text{ iff } C \in t. \tag{3}$$

We show (3) by *induction on the structure of* C:

- Let $C = A$ for a concept name A. Then (3) is an immediate consequence of the definition of $A^{\mathcal{I}}$.
- Let $C = \neg D$. Then induction yields $(t, j) \in D^{\mathcal{I}}$ iff $D \in t$. By contraposition, this is the same as $(t, j) \notin D^{\mathcal{I}}$ iff $D \notin t$. By Condition 1 in the definition of types and the semantics of negation, this is in turn equivalent to $(t, j) \in (\neg D)^{\mathcal{I}}$ iff $\neg D \in t$.
- Let $C = D \sqcap E$. Then induction yields $(t, j) \in D^{\mathcal{I}}$ iff $D \in t$ and $(t, j) \in E^{\mathcal{I}}$ iff $E \in t$. From this, we obtain $(t, j) \in (D \sqcap E)^{\mathcal{I}}$ iff $D \sqcap E \in t$ using Condition 2 in the definition of types and the semantics of conjunction.
- Let $C = \exists r.D$. First, assume that $(t, j) \in C^{\mathcal{I}}$, i.e., there is $(t', j') \in \Delta^{\mathcal{I}}$ such that $((t, j), (t', j')) \in r^{\mathcal{I}}$ and $(t', j') \in D^{\mathcal{I}}$. The former implies that there is an existential restriction $\exists r.D'$ in t that is satisfied by t'. The latter implies by induction that $D \in t'$. Now, assume that $C = \exists r.D \notin t$. By Condition 1 in the definition of types this implies $\neg\exists r.D \in t$. But then we would need to have $\neg D \in t'$ since t' satisfies $\exists r.D'$ in t. This contradicts the fact that $D \in t'$. Consequently, we have $C \in t$.

  Second, assume that $C = \exists r.D \in t$. Since $(t, j) \in \Delta^{\mathcal{I}}$, we have $\sigma(v_t) > 0$, and thus there is a type t' satisfying $\exists r.D$ in t such that $\sigma(v_{t'}) > 0$. Consequently $(t', 1) \in \Delta^{\mathcal{I}}$ and $((t, j), (t', 1)) \in r^{\mathcal{I}}$. Since t' satisfies $\exists r.D$ in t, we have $D \in t'$, and induction yields $(t', 1) \in D^{\mathcal{I}}$. This shows $(t, j) \in (\exists r.D)^{\mathcal{I}}$.

This completes the proof of (3) and thus the proof of the lemma. □

Since satisfiability of QFBAPA formulae can be decided within NP even for binary coding of numbers [10], this lemma shows that consistency of $\mathcal{ALC}$ ECBoxes can be decided within NExpTime. Together with the NExpTime lower bound stated in Proposition 1, this yields:

Theorem 8. *Consistency of ECBoxes is NExpTime-complete in $\mathcal{ALC}$ both for unary and binary encoding of numbers.*

5 Restricted Cardinality Constraints on Concepts

The extended cardinality constraints introduced above are expressive enough to formulate cardinality restrictions on concepts, and thus increase the complexity of reasoning in $\mathcal{ALC}$ from ExpTime to NExpTime. In this section we introduce a restricted version of our constraints, which still retains some of the extended expressiveness, but is restricted enough to leave the complexity of reasoning within ExpTime. Basically, the high complexity of ECBoxes comes from two sources. First, the NExpTime-hardness proof for $\mathcal{ALC}$ with cardinality restrictions given in [17] makes use of constraints of the form $|C| \leq n$, where n is a number with exponentially large value, but polynomially large binary representation. We will disallow such direct comparisons to integer constants, by restricting the constraints to comparisons between (linear combinations of) concept cardinalities. Second, the use of arbitrary Boolean combinations of constraints in ECBoxes causes NP-complexity of testing solvability of the constraint formula independently of which kind of constraints are actually considered. *Restricted cardinality constraints on $\mathcal{ALC}$ concepts* avoid these two sources of complexity:

- *restricted $\mathcal{ALC}$ cardinality constraints* are of the form

$$N_1|C_1| + N_2|C_2| + \cdots + N_k|C_k| \leq N_{k+1}|C_{k+1}| + \cdots + N_{k+\ell}|C_{k+\ell}|,$$

  where C_i are $\mathcal{ALC}$ concept descriptions and N_i are integer constants for $1 \leq i \leq k + \ell$;
- a *restricted $\mathcal{ALC}$ cardinality box (RCBox)* is a conjunction of restricted $\mathcal{ALC}$ cardinality constraints.

Restricted cardinality constraints are still expressive enough to formulate probabilistic conditionals. In fact, the ECBox in Lemma 2 is actually an RCBox. Restricted cardinality constraints can be seen as generalizations of probabilistic conditionals where one can compare the ratio of linear combinations of cardinalities. Indeed, any restricted cardinality constraint is equivalent to an inequality of the form:

$$\frac{N_1|C_1| + N_2|C_2| + \cdots + N_k|C_k|}{N_{k+1}|C_{k+1}| + \cdots + N_{k+\ell}|C_{k+\ell}|} \leq q,$$

for $q \in \mathbb{Q}$, and vice versa.

Lemma 2 implies that an upper bound on the complexity of $\mathcal{ALC}$ with restricted cardinality constraints also holds for statistical $\mathcal{ALC}$. Conversely, the ExpTime-hardness of $\mathcal{ALC}$ with CIs also holds for statistical $\mathcal{ALC}$, and thus for $\mathcal{ALC}$ with restricted cardinality constraints.

Proposition 9. *Consistency of RCBoxes is ExpTime-hard in $\mathcal{ALC}$, independently of whether numbers are encoded in unary or binary.*

6 Consistency of $\mathcal{ALC}$ RCBoxes

In this section, we show that consistency of $\mathcal{ALC}$ RCBoxes is in ExpTime even for binary coding of numbers. As in Section 4 we assume that all the concept descriptions occurring in RCBoxes are built using the constructors conjunction, negation, and existential restriction only. We again replace each cardinality $|C|$ occurring in the RCBox by the sum of variables $\sum_{t|C\in t} v_t$ for all types t that contain C. Additionally, for each type t, we add an inequality $v_t \geq 0$. Due to the restriction that RCBoxes may only compare cardinalities with each other, but not with constants, this system of linear inequalities can be transformed into the form $A \cdot \boldsymbol{v} \geq \mathbf{0}$

with $\boldsymbol{v} \geq \mathbf{0}$. As in the previous section, however, not every solution to this system of linear inequalities corresponds to a model of the RCBox since existential restrictions in realized types are not yet taken into account. For this purpose, we also need to require some of the variables v_t to have a value ≥ 1. Overall, this results in a system of linear inequalities of exponential size, which satisfies the properties stated in the next lemma.

Lemma 10. *Let ϕ be a system of linear inequalities consisting of $A \cdot \boldsymbol{v} \geq \mathbf{0}$, $\boldsymbol{v} \geq \mathbf{0}$, and $B \cdot \boldsymbol{v} \geq \mathbf{1}$, where A, B are matrices of integer coefficients and $\boldsymbol{v}$ is the variable vector.*

1. *Deciding whether ϕ has a non-negative integer solution can be done in polynomial time.*
2. *The solutions of ϕ are closed under addition, and in particular we have the following: if T is a set of types such that, for all $t \in T$, ϕ has a solution σ_t in which v_t has a non-zero value, then there is a non-negative integer solution σ such that $\sigma(v_t) \geq 1$ for all $t \in T$.*

Proof. Regarding the first statement, note that solvability of ϕ in the *reals* can be checked in polynomial time using linear programming [9]. Since all coefficients occurring in ϕ are rational, if a real solution exists, then there is also a rational solution of polynomial size [16]. Since solutions of $A \cdot \boldsymbol{v} \geq \mathbf{0}$, $\boldsymbol{v} \geq \mathbf{0}$, and $B \cdot \boldsymbol{v} \geq \mathbf{1}$ are obviously closed under multiplication with positive integer constants, multiplication with the least common multiple of the denominators then shows that systems solvable in the reals also have an integer solution. Summing up, we know that solvability of such a system in the *non-negative integers* can be checked in polynomial time in the size of the system. Note that this polynomial upper bound holds for numbers in the matrices A, B encoded in binary.

Regarding the second statement, note that, for solutions σ_1 and σ_2 of ϕ, σ with $\sigma(v_t) = \sigma_1(v_t) + \sigma_2(v_t)$ is a solution as well, i.e., the solutions of ϕ are closed under addition. But then, if T is a set of types such that, for all $t \in T$, the system has a solution σ_t in which v_t has a non-zero value, then there is a non-negative integer solution σ consisting of the sum of all σ_t in which $\sigma(v_t) \geq 1$ for all $t \in T$. □

Basically, our algorithm for deciding the consistency of RCBoxes described in Figure 1 is a classical type elimination algorithm [12, 14], but instead of only removing types whose existential restrictions can no longer be satisfied, we also remove types whose variables are forced to have value 0 by the current system of inequalities. We need to show that the algorithm is sound and complete and runs in ExpTime.

Lemma 11. *Given an $\mathcal{ALC}$ RCBox $\mathcal{R}$, the call* consistent$(\mathcal{R})$ *runs in time exponential in the size of $\mathcal{R}$.*

Proof. For any RCBox $\mathcal{R}$ the number of types of $\mathcal{R}$ is at most exponential. Type elimination removes one type during each iteration of its while loop, and the same is true for the repeat loop of the procedure consistent.

The system of linear inequalities constructed during the main loop of consistent$(\mathcal{R})$ has at most exponential size, since each cardinality $|C|$ is replaced by a sum of at most exponentially many types, and it satisfies the preconditions of Lemma 10. Thus, it can be solved in ExpTime. All together, this provides us with an ExpTime upper bound on the runtime of consistent$(\mathcal{R})$. □

Lemma 12. *The algorithm* consistent *is sound and complete.*

Procedure: type-elimination(T)
Input: T: a set of types
Output: the largest subset of T in which every existential restrictions occurring in a type in this set is satisfied in this set

while there is a type $t \in T$ with unsatisfied existential restrictions in T **do**
  $T \leftarrow T \setminus \{t\}$
end while
return T

Procedure: consistent($\mathcal{R}$)
Input: $\mathcal{R}$: $\mathcal{ALC}$ RCBox
Output: `true`, if $\mathcal{R}$ is consistent; otherwise `false`

$T \leftarrow$ type-elimination(types($\mathcal{R}$))
repeat
  $T' \leftarrow T$
  $\phi_T \leftarrow$ replace each $|C|$ in $\mathcal{R}$ with $\sum_{t \in T \text{ s.t. } C \in t} v_t$ and add $v_t \geq 0$ for each $t \in T$
  if there is a type $t \in T$ such that $\phi_T \wedge v_t \geq 1$ has no solution **then**
    $T \leftarrow$ type-elimination($T \setminus \{t\}$)
  end if
until $T = T'$
return $T \neq \emptyset$

Figure 1: The $\mathcal{ALC}$ RCBox consistency algorithm

Proof. Let $\mathcal{R}$ be an $\mathcal{ALC}$ RCBox. In order to show *soundness*, assume that consistent($\mathcal{R}$) return `true`. This means that it computes a non-empty set T of types such that no type in T has unsatisfied existential restrictions in T (otherwise it would have been eliminated during the last call of type-elimination), and for each type $t \in T$ there is an solution σ_t to ϕ_T with $\sigma(v_t) \geq 1$. By Lemma 10, this implies that ϕ_T has a non-negative integer solution σ in which for all $t \in T$ we have $\sigma(v_t) \geq 1$. We define an interpretation $\mathcal{I}$ as follows:

$$\Delta^{\mathcal{I}} := \{(t, j) \mid t \in T \text{ and } 1 \leq j \leq \sigma(v_t)\},$$
$$A^{\mathcal{I}} := \{(t, j) \in \Delta^{\mathcal{I}} \mid A \in t\} \text{ for } A \in N_C, \text{ and}$$
$$r^{\mathcal{I}} := \{((t, j), (t', j')) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid t' \text{ satisfies an existential restriction for } r \text{ in } t\}.$$

Analogously to the proof of Lemma 7, we can show that $(t, j) \in C^{\mathcal{I}}$ iff $C \in t$ for all types $t \in T$ and all $j, 1 \leq j \leq \sigma(v_t)$ by induction on the structure of C. There is one difference in the argumentation compared to Lemma 7: For the case $C = \exists r.D$, if we assume that $\exists r.D \in t$, we know that (since there are no unsatisfied existential restrictions in any of the types $t \in T$) there must be a type $t' \in T$ that satisfies $\exists r.D$ in t. As σ assigns a value ≥ 1 to all type variables in T, we have $\sigma(v_{t'}) \geq 1$ and thus $(t', 1) \in \Delta^{\mathcal{I}}$ and $((t, j), (t', 1)) \in r^{\mathcal{I}}$ by definition of $r^{\mathcal{I}}$. Since t' satisfies $\exists r.D$ in t, this implies $D \in t'$ and hence $(t', 1) \in D^{\mathcal{I}}$ by induction, which finally yields $(t, j) \in (\exists r.D)^{\mathcal{I}}$.

The equivalence we have just shown also implies $(t, j) \in C_t^{\mathcal{I}}$ and thus $|C_t^{\mathcal{I}}| = \sigma(v_t)$ for all types t. Since σ solves ϕ_T, this implies that $\mathcal{I}$ is a model of $\mathcal{R}$.

To show *completeness*, assume that there exists a model $\mathcal{I}$ of $\mathcal{R}$ and define $T_{\mathcal{I}} = \{t \in \text{types}(\mathcal{R}) \mid |C_t^{\mathcal{I}}| > 0\}$. Similarly to Lemma 5, we formulate the following claim:

Claim 13. *Assume that $\mathcal{I}$ is a finite interpretation that is a model of $\mathcal{R}$. Then for all $T \supseteq T_{\mathcal{I}}$, if we define $\sigma(v_t) := |C_t^{\mathcal{I}}|$ for all types $t \in T$, then σ is a solution of ϕ_T.*

This claim immediately follows from Lemma 5 and the fact that all the types in types$(\mathcal{R}) \setminus T$ satisfy $|C_t^{\mathcal{I}}| = 0$. However, it does not immediately imply that consistent$(\mathcal{R})$ returns `true`. However, we can show that none of the types $t \in T_{\mathcal{I}}$ will ever get removed from the set T in the algorithm. We show this by induction on the number k of iterations of the repeat-until loop.

In case $k = 0$, i.e., before entering the loop the first time, we know that $T_{\mathcal{I}} \subseteq$ types$(\mathcal{R})$ and no type $t \in T_{\mathcal{I}}$ has unsatisfied existential restrictions, since $\mathcal{I}$ is a model of $\mathcal{R}$ and thus all existential restrictions of any $t \in T_{\mathcal{I}}$ are satisfied even within $T_{\mathcal{I}}$. Consequently, none of these types is removed in the call of type-elimination.

For the case $k > 0$, we know that there is a solution σ of ϕ_T due to Claim 13, and in particular, this solution has $\sigma(v_t) \geq 1$ for all $t \in T_{\mathcal{I}}$. Thus none of the $t \in T_{\mathcal{I}}$ can satisfy the if-statement in the loop. For the same reason as before, type-elimination will not remove types $t \in T_{\mathcal{I}}$.

Thus, the algorithm will eventually terminate due to Lemma 11 with a set $T \supseteq T_{\mathcal{I}}$. Since the model $\mathcal{I}$ has a non-empty domain, $T_{\mathcal{I}} \neq \emptyset$, and thus the procedure returns `true`. □

This completes the proof that consistency for $\mathcal{ALC}$ RCBoxes is in ExpTime for binary coding of numbers, which implies the same upper bound also for unary coding. Since $\mathcal{ALC}$ TBoxes without cardinality constraints already have an ExpTime-hard consistency problem, this finally implies ExpTime-completeness of consistency for RCBoxes

Theorem 14. *Consistency of RCBoxes is ExpTime-complete in $\mathcal{ALC}$ both for unary and binary encoding of numbers.*

Since RCBoxes can express statistical KBs, this also implies ExpTime-completeness for the consistency problem of statistical $\mathcal{ALC}$ KBs. Furthermore, it is not hard to show that the ExpTime complexity also extends to entailment of PCs from statistical KBs (called l-entailment in [11]).

Corollary 15. *Consistency of statistical $\mathcal{ALC}$ KBs, and entailment of a PC $(C|D)[\ell, u]$ from a statistical $\mathcal{ALC}$ KB $\mathcal{K}$ are ExpTime-complete both for unary and binary encoding of numbers.*

Proof. The proof for consistency is trivial. Regarding entailment, recall that the PC $(C|D)[\ell, u]$ is entailed by a statistical $\mathcal{ALC}$ KB $\mathcal{K}$, if all models $\mathcal{I}$ of $\mathcal{K}$ also satisfy

$$|D|^{\mathcal{I}} = 0 \text{ or } \ell \leq \frac{|C \sqcap D|^{\mathcal{I}}}{|D|^{\mathcal{I}}} \leq u.$$

Note that models with $|D|^{\mathcal{I}} = 0$ always satisfy the PC $(C|D)[\ell, u]$, so we only have to check all models $\mathcal{I}$ with $|D|^{\mathcal{I}} \geq 1$. Let $\mathcal{R}$ be the equivalent RCBox for $\mathcal{K}$ and let $\ell = \ell_1/\ell_2$ and $u = u_1/u_2$ for $\ell_1, \ell_2, u_1, u_2 \in \mathbb{N}$. Then, we know that $\mathcal{K}$ entails $(C|D)[\ell, u]$ iff

$$\mathcal{R} \wedge |D| \geq 1 \wedge \ell_2 \cdot |C \sqcap D| < \ell_1 \cdot |D| \quad \text{and} \quad \mathcal{R} \wedge |D| \geq 1 \wedge u_2 \cdot |C \sqcap D| > u_1 \cdot |D|$$

are both inconsistent. In particular, since we are only dealing with integers here, we can replace the strict inequalities by non-strict ones by simply adding a variable v_ϵ. Thus, the entailment holds iff $\mathcal{R} \wedge |D| \geq 1 \wedge \ell_2 \cdot |C \sqcap D| + v_\epsilon \leq \ell_1 \cdot |D| \wedge v_\epsilon \geq 1$ and $\mathcal{R} \wedge |D| \geq 1 \wedge u_2 \cdot |C \sqcap D| \geq u_1 \cdot |D| + v_\epsilon \wedge v_\epsilon \geq 1$ are both inconsistent. This way, we obtain RCBoxes extended by a single positive integer variable v_ϵ, for which $v_\epsilon \geq 1$ must hold. While these are not RCBoxes in the sense introduced in Section 5, the only effect this additional inequality has is that, in the

systems of inequalities considered in a run of the algorithm consistent, there is an additional inequality $v_\epsilon \geq 1$, which does not destroy applicability of Lemma 10. Thus, entailment of PCs by a statistical $\mathcal{ALC}$ KB can be decided in ExpTime. This upper bound holds for binary coding of numbers, and thus also for unary coding. The ExpTime lower bound for unary (and thus also binary) coding follows from the fact that subsumption w.r.t. $\mathcal{ALC}$ TBoxes is a special case of entailment of PCs from a statistical $\mathcal{ALC}$ KB. □

7 Conclusion

We have used the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) as an inspiration for introducing very expressive cardinality constraints on concepts in the DL $\mathcal{ALC}$. Although considerably more expressive than the traditional cardinality restrictions on concepts, the complexity of the consistency problem in this new formalism does not increase compared to the case of cardinality restrictions. Nevertheless, the NExpTime complexity achieved this way is quite high. For this reason, we have introduced a restricted formalism, for which the complexity is ExpTime. Since this restricted formalism can express the statistical knowledge bases of [11], this also settles the open problem of what is the exact complexity of reasoning w.r.t. statistical knowledge bases in $\mathcal{ALC}$. Interestingly, our complexity results do not depend on whether unary or binary coding of numbers is assumed.

Regarding related work, we have already mentioned in the introduction that the NExpTime upper bound for the general case actually also follows from the NExpTime upper bound in [18] for a more expressive logic with n-ary relations and function symbols, called QFBAPA-Rel. In fact, our ECBoxes correspond to the restriction of QFBAPA-Rel to the case where no function symbols are used and the only expressions involving relations are of the form $r^{-1}[B]$ for binary relations r (see Fig. 3 in [18]), which correspond to existential restrictions. Interestingly, the NExpTime lower bound for reasoning in QFBAPA-Rel is shown in [18] for the restricted case without any relation symbols and only one unary function symbol. Thus, expressibility of ECBoxes in QFBAPA-Rel yields NExpTime-hardness of a quite different fragment of this logic. We have give our own proof of the NExpTime upper bound for the case of extended cardinality constraints on concepts since it is quite simple and sets the stage for our ExpTime result in the restricted case. This result does not follow from any of the results in [18], and thus is the main new result of this paper.

Regarding future work, it would be interesting to combine the two formalisms for cardinality constraints on concepts introduced in the present paper with the QFBAPA-based more expressive number restrictions investigated in [1]. Extending our algorithms to the combined formalisms is probably a non-trivial task. In fact, for the formalisms considered in the present paper, there is only a weak interaction between the concept descriptions and the cardinality constraints: existential restrictions in non-empty types may require certain other types to be non-empty as well. With very expressive number restrictions, more sophisticated interactions can occur. In fact, both the cardinality restrictions and the number restrictions induce systems of inequations, which must be appropriately combined.

Regarding implementation, procedures based on type-elimination such as the one employed to show our ExpTime upper bound for the restricted formalism are usually not useful in practice since they are not only worst-case but also best-case exponential. We intend to investigate whether a tableau-like procedure combined with a solver for linear inequations can be used to obtain a more practical procedure.

References

[1] Franz Baader. A new description logic with set constraints and cardinality constraints on role successors. In *Proc. of the 11th Int. Symp. on Frontiers of Combining Systems (FroCoS'17)*, volume 10483 of *Lecture Notes in Computer Science*, Brasília, Brazil, 2017. Springer-Verlag. To appear.

[2] Franz Baader, Martin Buchheit, and Bernhard Hollunder. Cardinality restrictions on concepts. *Artificial Intelligence*, 88(1–2):195–213, 1996.

[3] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[4] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[5] Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori Alperin Resnick. CLASSIC: A structural data model for objects. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 59–67, 1989.

[6] Robert Hoehndorf, Paul N. Schofield, and Georgios V. Gkoutos. The role of ontologies in biological and biomedical research: A functional perspective. *Brief. Bioinform.*, 16(6):1069–1080, 2015.

[7] Bernhard Hollunder and Franz Baader. Qualifying number restrictions in concept languages. In *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 335–346, 1991.

[8] Bernhard Hollunder, Werner Nutt, and Manfred Schmidt-Schauß. Subsumption algorithms for concept description languages. In *Proc. of the 9th Eur. Conf. on Artificial Intelligence (ECAI'90)*, pages 348–353, London (United Kingdom), 1990. Pitman.

[9] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proc. of the 16th Annual ACM Symp. on Theory of Computing (STOC'84)*, pages 302–311, 1984. ACM.

[10] Viktor Kuncak and Martin C. Rinard. Towards efficient satisfiability checking for Boolean algebra with Presburger arithmetic. In *Proc. of the 21st Int. Conf. on Automated Deduction (CADE-07)*, volume 4603 of *Lecture Notes in Computer Science*, pages 215–230. Springer-Verlag, 2007.

[11] Rafael Peñaloza and Nico Potyka. Towards statistical reasoning in description logics over finite domains. In *Proc. of the 11th Int. Conf. on Scalable Uncertainty Management (SUM 2017)*, volume 10564 of *Lecture Notes in Computer Science*, Granada, Spain, 2017. Springer-Verlag. To appear.

[12] V. R. Pratt. Models of program logic. In *Proc. of the 20th Annual Symp. on the Foundations of Computer Science (FOCS'79)*, pages 115–122, 1979.

[13] Ian Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *J. of Logic, Language and Information*, 14(3):369–395, 2005.

[14] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Type-elimination-based reasoning for the description logic $\mathcal{SHIQ}\mathrm{b}_s$ using decision diagrams and disjunctive datalog. *Logical Methods in Computer Science*, 8(1), 2012.

[15] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 466–471, 1991.

[16] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[17] Stephan Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artificial Intelligence Research*, 12:199–217, 2000.

[18] Kuat Yessenov, Ruzica Piskac, and Viktor Kuncak. Collections, cardinalities, and relations. In *Proc. of the 11th Int. Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI'10)*, volume 5944 of *Lecture Notes in Computer Science*, pages 380–395. Springer-Verlag, 2010.