Expressive Cardinality Restrictions on Concepts in a Description Logic with Expressive Number Restrictions

Franz Baader
Theoretical Computer Science, TU Dresden
Nöthnitzer Straße 46
01062 Dresden, Germany
franz.baader@tu-dresden.de

ABSTRACT

In two previous publications we have, on the one hand, extended the description logic (DL) $\mathcal{ALCQ}$ by more expressive number restrictions using numerical and set constraints expressed in the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA). The resulting DL was called $\mathcal{ALCSCC}$. On the other hand, we have extended the terminological formalism of the well-known description logic $\mathcal{ALC}$ from concept inclusions (CIs) to more general cardinality constraints expressed in QFBAPA, which we called extended cardinality constraints. Here, we combine the two extensions, i.e., we consider extended cardinality constraints on $\mathcal{ALCSCC}$ concepts. We show that this does not increase the complexity of reasoning, which is NExpTime-complete both for extended cardinality constraints in the DL $\mathcal{ALC}$ and in its extension $\mathcal{ALCSCC}$. The same is true for a restricted version of such cardinality constraints, where the complexity of reasoning decreases to ExpTime, not just for $\mathcal{ALC}$, but also for $\mathcal{ALCSCC}$.

CCS Concepts

•Theory of computation → Description logics;

Keywords

Description Logic, Number Restrictions, Cardinality Restrictions, QFBAPA, Complexity

1. INTRODUCTION

Description Logics (DLs) [4] 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 [11]. 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). Using an example from [6], 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 used early on in DLs [8, 13, 12]. For example, using number restrictions, motorcycles can be constrained to being motor vehicles with exactly two wheels:

$$\begin{aligned}\mathit{Motorcycle} \sqsubseteq\ & \mathit{Motor\text{-}vehicle}\ \sqcap \\ & (\leqslant 2\, \mathit{part}.\mathit{Wheel}) \sqcap (\geqslant 2\, \mathit{part}.\mathit{Wheel}).\end{aligned}$$

The exact complexity of reasoning in $\mathcal{ALCQ}$, the DL that has all Boolean operations and number restrictions of the form $(\leqslant n\, r.C)$ and $(\geqslant n\, r.C)$ as concept constructors, was determined by Stephan Tobies [18, 20]: it is PSpace-complete without CIs and ExpTime-complete w.r.t. CIs, independently of whether the numbers occurring in the number restrictions are encoded in unary or binary. Note that, using unary coding of numbers, the number n is assumed to contribute n to the size of the input, whereas with binary coding the size of the number n is $\log n$. Thus, for large numbers, using binary coding is more realistic.

The classical number restrictions available in $\mathcal{ALCQ}$ can only be used to compare the number of role successors of an individual with a fixed natural number. They cannot relate numbers of different kinds of role successors to each other. This would, e.g., be required to state that the number of cylinders of a motor coincides with the number of spark plugs in this motor, without fixing what this number actually is. To overcome this deficit, we have extended $\mathcal{ALCQ}$ by allowing the statement of constraints on role successors that are more general than the number restrictions of $\mathcal{ALCQ}$ [1]. To formulate these constraints, we have used the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) [15], in which one can express Boolean combinations of set constraints and numerical constraints on the cardinalities of sets. In the resulting logic $\mathcal{ALCSCC}$, the above constraint regarding cylinders and spark plugs can be expressed using a cardinality constraint on the role *succ*essors:

$$Motor \sqsubseteq succ(|part \cap Cylinder| = |part \cap SparkPlug|).$$

In general, such a *succ*-expression considers the set of all role successors of a given individual, and requires certain subsets to satisfy the stated QFBAPA constraints. In our example, the cardinality of the set of *part*-successors that belong to the concept *Cylinder* must be the same as of the set of *part*-successors that belong to the concept *SparkPlug*.

Such cardinality constraints strictly extend the expressive power of $\mathcal{ALCQ}$. In [1] it shown that the constraint $succ(|r| = |s|)$, which describes individuals that have the same number of r-successors as s-successors, cannot be expressed in $\mathcal{ALCQ}$. In [5], the constraint $succ(|r \cap A| = |r \cap \neg A|)$ describing individuals such that the number of r-successors belonging to A is the same as the number of r-successors not belonging to A was shown to be not even expressible in first-order logic. Intuitively, both kinds of constraints can, e.g., be used to describe people that have the same number of sons and daughters, where in the first constraint one uses roles *son* and *daughter*, whereas in the second one uses the role *child* and the concept *Male*.

In spite of this considerable increase of the expressive power, we were able to show in [1] that this does not increase the complexity of reasoning: like for $\mathcal{ALCQ}$, the complexity of the satisfiability problem in $\mathcal{ALCSCC}$ is PSpace-complete without CIs and ExpTime-complete w.r.t. CIs. While the PSpace result also follows from previous work on modal logics with Presburger constraints [9], the ExpTime result was new.

Whereas number restrictions are local in the sense that they consider role successors of an individual under consideration (e.g. the wheels that are part of a particular motor vehicle), cardinality restrictions on concepts (CRs) [3, 19] are global, i.e., they consider all individuals in an interpretation. For example, the cardinality restriction

$$(\leqslant 45000000\,(Car \sqcap \exists registered\text{-}in.German\text{-}district))$$

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 fact, it is well known that models of CIs in $\mathcal{ALC}$ and $\mathcal{ALCQ}$ are closed under disjoint union (see [7], Theorem 3.8, for a proof in the case of $\mathcal{ALC}$), but models of a CR like $(\leqslant 1\,A)$ are clearly not.

In addition, CRs increase the complexity of reasoning: for the DLs $\mathcal{ALC}$ and $\mathcal{ALCQ}$, consistency w.r.t. CIs is ExpTime-complete [16, 20], but consistency w.r.t. CRs is NExpTime-complete if the numbers occurring in the CRs are assumed to be encoded in binary [19]. With unary coding of numbers, consistency stays ExpTime-complete even w.r.t. CRs [19], but the above example considering 45 million cars clearly shows that unary coding is not appropriate if numbers with large values are employed.

Just like classical number restrictions, CRs can only relate the cardinality of a concept to a fixed number. In [6], we have introduced and investigate more general constraints on the cardinalities of concepts, which we called extended cardinality constraints. The main idea was again to use QFBAPA to formulate and combine these constraints. An example of a constraint expressible this way, but not expressible using CRs is

$$\begin{aligned}&2 \cdot |Car \sqcap \exists registered\text{-}in.German\text{-}district \sqcap \exists fuel.Diesel|\\ &\le |Car \sqcap \exists registered\text{-}in.German\text{-}district \sqcap \exists fuel.Petrol|,\end{aligned}$$

which states that, in Germany, cars running on petrol outnumber cars running on diesel by a factor of at least two.

In [6] it is shown that, in the DL $\mathcal{ALC}$, the complexity of reasoning w.r.t. extended cardinality constraints (NExpTime for binary coding of numbers) is the same as for reasoning w.r.t. CRs. In addition, the paper introduces a restricted version of this formalism, which can express CIs, but not CRs, and shows that this way the complexity can be lowered to ExpTime. The NExpTime upper bound for the general case actually also follows from the NExpTime upper bound in [21] for a more expressive logic with n-ary relations and function symbols, but the ExpTime result for the restricted case was new. Regarding expressive power, we will show in this paper that, in contrast to CRs, extended cardinality constraints cannot be expressed in first-order logic.

The results on the complexity of reasoning with extended cardinality constraints in [6] were restricted to concepts of the DL $\mathcal{ALC}$. It was not even clear whether the complexity upper bounds also hold for $\mathcal{ALCQ}$, let alone the considerably more expressive DL $\mathcal{ALCSCC}$. In the present paper, we combine the work in [1] and [6] by considering extended cardinality constraints in $\mathcal{ALCSCC}$. This turned out to be non-trivial since the local cardinality constraints of $\mathcal{ALCSCC}$ may interact with the global ones in the extended cardinality constraints. Nevertheless, we are able to show that the complexity results (NExpTime-complete in general, and ExpTime-complete in the restricted case) hold not only for $\mathcal{ALC}$, but also for $\mathcal{ALCSCC}$.

2. PRELIMINARIES

Before we define $\mathcal{ALCSCC}$ and extended cardinality constraints, we must introduce QFBAPA, on which both are based. More details on this logic can be found in [15].

2.1 The logic QFBAPA

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. For example, if *Car*, *Motorcycle*, *Motor-vehicle* are set variables, then the set constraints

$$Car \cap Motorcycle = \emptyset, \quad Car \cup Motorcycle \subseteq Motor\text{-}vehicle$$

say that cars are not motorcycles, and that both are motor vehicles.

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. For example, the numerical constraint

$$|Car| > 10\cdot(|Motorcycle| + |EBike|)$$

says that there are more than ten times as many cars as there are motorcycles and E-bikes together. A *QFBAPA formula* is a Boolean combination of set and numerical constraints. For example, the QFBAPA formula

$$|GoodThing| > 0 \Rightarrow |GoodThing| \geq 3$$

expresses the saying that all good things come in threes.

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 the cardinality of the union of the sets $\sigma(s)$ and $\sigma(t)$ is the same as the sum of the cardinalities of these sets. Note that this is the case iff $\sigma(s)$ and $\sigma(t)$ are disjoint, which we could also have expressed using the set constraint $s \cap t \subseteq \emptyset$, as in the above example. A QFBAPA formula ϕ is *satisfiable* if it has a solution. As shown in [15], satisfiability of QFBAPA formulae is an NP-complete problem.

The main tool used in [15] to show that satisfiability in QFBAPA is in NP is a "sparse solution" lemma (see Lemma 1 below), which will also turn out to be useful for showing one of our complexity upper bounds. Assume that ϕ is a QFBAPA formula containing the set variables $X_1, \ldots, X_k$. A *Venn region* is of the form

$$X_1^{c_1} \cap \ldots \cap X_k^{c_k},$$

where c_i is either empty or c for $i = 1, \ldots, k$. It is shown in [15] that, given ϕ, one can easily compute a number N whose value is polynomial in the size of ϕ such that the following holds: ϕ is satisfiable iff it has a solution in which $\leq N$ Venn regions are interpreted by non-empty sets. Taking a closer look at how this result is proved in [15], one can actually strengthen it (see [1] for a proof).

Lemma 1. *For every QFBAPA formula ϕ, one can compute in polynomial time a number N whose value is polynomial in the size of ϕ such that the following holds for every solution σ of ϕ: there is a solution σ' of ϕ such that*

- $|\{v \mid v$ *Venn region and* $\sigma'(v) \neq \emptyset\}| \leq N$, *and*
- $\{v \mid v$ *Venn region and* $\sigma'(v) \neq \emptyset\} \subseteq \{v \mid v$ *Venn region and* $\sigma(v) \neq \emptyset\}$.

2.2 Extended number restrictions

In the following, we recall syntax and semantics of $\mathcal{ALCSCC}$. In this logic, which was first introduced in [1], numerical and set constraints of QFBAPA can be used to formulate more expressive number restrictions. Basically, $\mathcal{ALCSCC}$ provides us with Boolean operations on concepts and constraints on role successors, which are expressed in QFBAPA. In these constraints, role names and concept descriptions can be used as set variables, and there are no PA variables allowed.

Definition 2 (Syntax of $\mathcal{ALCSCC}$). *Given finite, disjoint sets N_C of concept names and N_R of role names, the set of $\mathcal{ALCSCC}$ concept descriptions over the signature (N_C, N_R) is inductively defined as follows:*

- *every concept name in N_C is an $\mathcal{ALCSCC}$ concept description over (N_C, N_R);*
- *if C, D are $\mathcal{ALCSCC}$ concept descriptions over the signature (N_C, N_R), then so are $C \sqcap D$, $C \sqcup D$, and $\neg C$;*
- *if Con is a set or numerical constraint of QFBAPA using role names and already defined $\mathcal{ALCSCC}$ concept descriptions over the signature (N_C, N_R) as set variables, then succ(Con) is an $\mathcal{ALCSCC}$ concept description over (N_C, N_R).*

An $\mathcal{ALCSCC}$ TBox over (N_C, N_R) is a finite set of concept inclusions of the form $C \sqsubseteq D$, where C, D are $\mathcal{ALCSCC}$ concept descriptions over (N_C, N_R).

For example, the $\mathcal{ALCSCC}$ concept description

$$Female \sqcap succ(|child \cap Female| = |child \cap Male|)$$

describes all female individuals that have exactly as many sons as daughters. Of course, successor constraints can also be nested, as in the $\mathcal{ALCSCC}$ concept description

$$succ(|child \cap succ(child \subseteq Female)| = |child \cap succ(child \subseteq Male)|),$$

which describes all individuals having as many children that have only daughters as they have children having only sons. For the sake of simplicity, we will sometimes use "concept" in place of "concept description," and often dispense with explicitly mentioning the signature.

As usual in DL, the semantics of $\mathcal{ALCSCC}$ is defined using the notion of an interpretation. A more detailed definition of the semantics can be found in [1].

Definition 3 (Semantics of $\mathcal{ALCSCC}$). *Given finite, disjoint sets N_C and N_R of concept and role names, respectively, an* interpretation *of N_C and N_R consists of a non-empty and* finite *set $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$ that maps every concept name $A \in N_C$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and every role name $r \in N_R$ to a binary relation $r^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$. Given an individual $d \in \Delta^{\mathcal{I}}$ and a role name $r \in N_R$, we define $r^{\mathcal{I}}(d) := \{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\}$ (r-successors) and $ars^{\mathcal{I}}(d) := \bigcup_{r \in N_R} r^{\mathcal{I}}(d)$ (all role successors).*
The function $\cdot^{\mathcal{I}}$ is inductively extended to $\mathcal{ALCSCC}$ concept descriptions over (N_C, N_R) by interpreting $\sqcap$, $\sqcup$, and $\neg$ respectively as intersection, union and complement. Successor constraints are evaluated according to the semantics of QFBAPA: to determine whether $d \in succ(Con)^{\mathcal{I}}$ or not,

- *$\mathcal{U}$ is evaluated as $ars^{\mathcal{I}}(d)$ (i.e., the set of all role successors of d),*
- *$\emptyset$ as the empty set,*
- *roles r occurring in Con as $r^{\mathcal{I}}(d)$ (i.e., the set of r-successors of d),*
- *and concept descriptions D as $D^{\mathcal{I}} \cap ars^{\mathcal{I}}(d)$ (i.e., the set of role successors of d that belong to D). Note that, by induction, the sets $D^{\mathcal{I}}$ are well-defined.*

Then $d \in succ(Con)^{\mathcal{I}}$ iff the substitution obtained this way is a solution of the QFBAPA formula Con.
The interpretation $\mathcal{I}$ is a model *of the TBox $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all $C \sqsubseteq D \in \mathcal{T}$. The $\mathcal{ALCSCC}$ concept description C is* satisfiable *if there is an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$, and it is* satisfiable w.r.t. the TBox $\mathcal{T}$ *if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. The $\mathcal{ALCSCC}$ concept descriptions C, D are* equivalent *(written $C \equiv D$) if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$.*

Note that, in the above definition, interpretations are required to have a finite domain to ensure that all the sets considered when evaluating QFBAPA constraints are finite. In [1] a weaker restriction is used, where only the number of role successors of all individuals must be finite. Here we use the stronger restrictions that $\Delta^{\mathcal{I}}$ is finite since we need this restriction in the presence of extended cardinality constraints [6] (see below). By using a variant of QFBAPA that can also deal with infinite sets [14, 5], one can actually define a variant of $\mathcal{ALCSCC}$ in which the restriction to finite domains is not needed [5]. However, in the following we stick with the version originally introduced in [1, 6] and defined above.

Note that $\mathcal{ALCSCC}$ contains the well-known DLs $\mathcal{ALC}$ and $\mathcal{ALCQ}$ as sub-logics. In fact, in $\mathcal{ALC}$ only successors constraints of the form $succ(r \subseteq C)$ and $succ(|r \cap C| \geq 1)$ are allowed, which are usually written as $\forall r.C$ and $\exists r.C$, respectively. Similarly, in $\mathcal{ALCQ}$ only successors constraints of the form $succ(|r \cap C| \leq n)$ and $succ(|r \cap C| \geq n)$ are allowed, which are usually written as $\leqslant n\, r.C$ and $\geqslant n\, r.C$, respectively.

As shown in [1], satisfiability of $\mathcal{ALCSCC}$ concept descriptions is PSpace-complete without TBox and ExpTime-complete w.r.t. TBoxes. Thus, these problems have the same complexity as in $\mathcal{ALC}$ and $\mathcal{ALCQ}$.

2.3 Extended cardinality constraints

Basically, extended cardinality constraints are QFBAPA formulae where $\mathcal{ALCSCC}$ concept descriptions are used in place of set variables. However, since inclusion constraints $s \subseteq t$ (and thus equality constraints) 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 $\mathcal{ALCSCC}$ has all Boolean operations on concepts, we do not need Boolean operations on set terms. Consequently, we define *extended cardinality constraints on $\mathcal{ALCSCC}$ concepts* as follows:

Definition 4. *Let N_C and N_R be finite, disjoint sets concept names and of role names, respectively.*

- *$\mathcal{ALCSCC}$* cardinality terms *over the signature (N_C, N_R) are built from integer constants and concept cardinalities $|C|$ for $\mathcal{ALCSCC}$ concepts C over (N_C, N_R) using addition and multiplication with integer constants.*
- Extended $\mathcal{ALCSCC}$ cardinality constraints *over the signature (N_C, N_R) are of the form $k = \ell, k < \ell, N \,\mathsf{dvd}\, \ell$, where N is an integer constant and k, ℓ are $\mathcal{ALCSCC}$ cardinality terms over (N_C, N_R).*
- Extended $\mathcal{ALCSCC}$ cardinality boxes *over the signature (N_C, N_R) are Boolean combinations of extended $\mathcal{ALCSCC}$ cardinality constraints over (N_C, N_R). We will call such a Boolean combination an* ECBox.

When defining the *semantics of ECBoxes*, we again restrict the attention to *finite* interpretations to ensure that cardinalities of concepts 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 the case for the comparison operators $=$, $<$, and divisibility dvd. This yields the semantics of extended $\mathcal{ALCSCC}$ cardinality constraints and thus also of their Boolean combinations.

As shown in [5], $\mathcal{ALCSCC}$ concepts are in general nor expressible in first-order logic. We will show now that the same is true for extended cardinality constraints, even if the concepts used within these constraints are first-order expressible.

Proposition 5. *The extended cardinality constraint*

$$2 \,\mathsf{dvd}\, |\top|$$

cannot be expressed in first-order logic, i.e., there is no first-order sentence φ such that φ and $2 \,\mathsf{dvd}\, |\top|$ have the same finite models.

Proof. Clearly, a given interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies the extended cardinality constraint $2 \,\mathsf{dvd}\, |\top|$ iff the cardinality of $\Delta^{\mathcal{I}}$ is even. However, even cardinality of the domain cannot be expressed in first-order logic since this would contradict the so-called 0–1-law. To be more precise, the 0–1-law says the following. For a first-order sentence φ and a positive integer N, let p_N be the percentage of interpretations with domain $\{1, \ldots, N\}$ that satisfy φ among all interpretations with this domain. Then $\lim_{N \to \infty} p_N$ exists and is either 0 or 1 [10]. Obviously, if φ were a first-order sentence expressing even cardinality, then we would have $\ell_N = 0$ if N is odd and $\ell_N = 1$ if N is even, and thus the limit would not exist.

Regarding the complexity of reasoning with extended cardinality constraints, note that extended cardinality constraints on $\mathcal{ALC}$ concepts are clearly a special case of extended cardinality constraints on $\mathcal{ALCSCC}$ concepts. In [6] it was shown that consistency of $\mathcal{ALC}$ ECBoxes is a NExpTime-complete problem independently of whether the numbers occurring in the ECBoxes are encoded in unary or binary. The hardness result clearly transfers to $\mathcal{ALCSCC}$.

Proposition 6. *Consistency of ECBoxes is NExpTime-hard in $\mathcal{ALCSCC}$ both for unary and binary coding of numbers.*

In the next section, we show that the NExpTime upper bound holds not only for $\mathcal{ALC}$, but also for $\mathcal{ALCSCC}$.

3. CONSISTENCY OF ECBOXES

In the following we consider an $\mathcal{ALCSCC}$ 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 a NExpTime upper bound for consistency of $\mathcal{ALCSCC}$ ECBoxes. Such an ExpTime-reduction to satisfiability in QFBAPA has already been used in the consistency procedure for $\mathcal{ALC}$ ECBoxes in [6]. However, for $\mathcal{ALCSCC}$ it needs to be extended considerably. In addition to encoding the global cardinality constraints in $\mathcal{E}$ using the notion of a type, we also need to take care of the local constraints expressed by successor constraints within $\mathcal{ALCSCC}$, and to link both kinds of constraints. This makes both the reduction and the proof of its correctness considerably more complicated.

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 $\mathcal{M}$ consists of *all subdescriptions* of the concept descriptions occurring in $\mathcal{E}$ as well as the *negations of these subdescriptions.*

Definition 7. *A subset t of $\mathcal{M}$ is a* type *for $\mathcal{E}$ if it satisfies the following properties for all concept descriptions C, D:*

1. *if $\neg C \in \mathcal{M}$, then either C or $\neg C$ belongs to t;*
2. *if $C \sqcap D \in \mathcal{M}$, then $C \sqcap D \in t$ iff $C \in t$ and $D \in t$;*
3. *if $C \sqcup D \in \mathcal{M}$, then $C \sqcup D \in t$ iff $C \in t$ or $D \in t$.*

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 the definition of a type. 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. In particular, we have for all finite interpretations $\mathcal{I}$:

$$|C^{\mathcal{I}}| = \sum_{t\ type\ with\ C \in t} |C_t^{\mathcal{I}}|.$$

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.,

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

A model $\mathcal{I}$ of $\mathcal{E}$ then yields a solution of $\phi_{\mathcal{E}}$ as follows: if we define $\sigma(v_t) := |C_t^{\mathcal{I}}|$ for all types t, then σ is a solution of the QFBAPA formula $\phi_{\mathcal{E}}$.

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

$$\mathcal{E} := |\operatorname{succ}(|A \cap r| \geq 5)| \geq 1 \wedge |A| \leq 3.$$

In this case, $\mathcal{M}$ consists of the concept descriptions

$$A, \neg A, \operatorname{succ}(|A \cap r| \geq 5), \neg \operatorname{succ}(|A \cap r| \geq 5),$$

and there are four types:

$$\begin{aligned} t_1 &:= \{A, \operatorname{succ}(|A \cap r| \geq 5)\}, \\ t_2 &:= \{\neg A, \operatorname{succ}(|A \cap r| \geq 5)\}, \\ t_3 &:= \{A, \neg \operatorname{succ}(|A \cap r| \geq 5)\}, \\ t_4 &:= \{\neg A, \neg \operatorname{succ}(|A \cap r| \geq 5)\}. \end{aligned}$$

The ECBox $\mathcal{E}$ is now translated into the QFBAPA formula $\phi_{\mathcal{E}}$ by replacing $|\operatorname{succ}(|A \cap r| \geq 5)|$ 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., $\phi_{\mathcal{E}}$ is the following formula:

$$\begin{aligned} \phi_{\mathcal{E}} \quad = \quad & v_{t_1} \geq 0 \wedge v_{t_2} \geq 0 \wedge v_{t_3} \geq 0 \wedge v_{t_4} \geq 0 \wedge \\ & v_{t_1} + v_{t_2} \geq 1 \wedge v_{t_1} + v_{t_3} \leq 3. \end{aligned}$$

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 $\operatorname{succ}(|A \cap r| \geq 5)$, which implies that this element must have at least 5 distinct r-successors belonging to A. However, this is prohibited by $\mathcal{E}$ since the second conjunct states that globally (i.e., in the whole model) there are at most 3 elements belonging to A.

This example shows that we must take elements required by successor constraints 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 successor constraints in t are realized with the right multiplicity. In our example, the types t_1 and t_2 require that an element belonging to them has at least 5 distinct r-successors belonging to A. As shown in [1], we can express such constraints again by QFBAPA formulae. Given a type t, the (possibly negated) successor constraints occurring in t induce a QFBAPA formula ψ_t, in which the concepts C and roles r occurring in these constraints induce set variables X_C^t and X_r^t. In addition, to take into account the semantics of $\mathcal{ALCSCC}$, we conjoin

$$\alpha_t := (\mathcal{U}^t = \bigcup_{r \in N_R} X_r^t) \wedge \bigwedge_{C \in \mathcal{M}} X_C^t \subseteq \mathcal{U}^t$$

to the formula obtained from the successor constraints and replace $\mathcal{U}$ in these constraints by $\mathcal{U}^t$. In our example, we have

$$\begin{aligned} \psi_{t_1} &:= |X_A^{t_1} \cap X_r^{t_1}| \geq 5 \wedge \alpha_{t_1}, \\ \psi_{t_2} &:= |X_A^{t_2} \cap X_r^{t_2}| \geq 5 \wedge \alpha_{t_2}, \\ \psi_{t_3} &:= \neg(|X_A^{t_3} \cap X_r^{t_3}| \geq 5) \wedge \alpha_{t_3}, \\ \psi_{t_4} &:= \neg(|X_A^{t_4} \cap X_r^{t_4}| \geq 5) \wedge \alpha_{t_4}. \end{aligned}$$

In principle, the formula ψ_t constrains the "local" role successors of an individual of type t. In order to ensure that the Boolean structure of concepts is respected by the set variables, we introduce

$$\begin{aligned} \beta_t \quad := \quad & \bigwedge_{C \sqcap D \in \mathcal{M}} X_{C \sqcap D}^t = X_C^t \cap X_D^t \wedge \\ & \bigwedge_{C \sqcup D \in \mathcal{M}} X_{C \sqcup D}^t = X_C^t \cup X_D^t \wedge \bigwedge_{\neg C \in \mathcal{M}} X_{\neg C} = (X_C^t)^c. \end{aligned}$$

Lemma 8. *If σ solves β_t, then for every $e \in \sigma(\mathcal{U}^t)$ there is a unique type t_e such that $e \in \bigcap_{C \in t_e} \sigma(X_C^t)$.*

PROOF. Given $e \in \sigma(\mathcal{U}^t)$, we define

$$t_e := \{C \in \mathcal{M} \mid e \in \sigma(X^t_C)\}.$$

By definition, t_e satisfies $e \in \bigcap_{C \in t_e} \sigma(X^t_C)$. The set t_e is a type since σ satisfies β_t. Finally, assume that t' is a type such that $e \in \bigcap_{C \in t'} \sigma(X^t_C)$. To show that $t = t'$, first assume that $C \in t'$. Then $e \in \sigma(X^t_C)$, which implies $C \in t_e$ by the definition of t_e. Conversely, if $C \notin t'$, then $\neg C \in t'$ and thus $\neg C \in t_e$, which yields $C \notin t_e$.

It remains to link the local constraints ψ_t and β_t with the global ones in case type t is populated. Basically, we need to ensure that our solutions of the local constraints do not assume that a concept is populated by more individuals than the solution of the global constraints allows. This can be expressed by the following QFBAPA formula:

$$\gamma_t := \bigwedge_{t' \in \mathrm{types}(\mathcal{E})} |\bigcap_{C \in t'} X^t_C| \leq v_{t'}. \qquad (1)$$

In our example, the formulae γ_{t_i} (for $1 \leq i \leq 4$) in particular contain the conjuncts

$$|X^{t_i}_A \cap X^{t_i}_C| \leq v_{t_1} \text{ and } |X^{t_i}_A \cap X^{t_i}_{\neg C}| \leq v_{t_3},$$

where $C = \mathrm{succ}(|A \cap r| \geq 5)$. Overall, we translate the $\mathcal{ALCSCC}$ ECBox $\mathcal{E}$ into the QFBAPA formula

$$\delta_{\mathcal{E}} := \phi_{\mathcal{E}} \wedge \bigwedge_{t \in \mathrm{types}(\mathcal{E})} v_t = 0 \vee (\psi_t \wedge \beta_t \wedge \gamma_t).$$

Assume that σ is a solution of $\delta_{\mathcal{E}}$ for the ECBox $\mathcal{E}$ of our example. Then $\sigma(v_{t_1}) + \sigma(v_{t_2}) \geq 1$, which implies that $\sigma(v_{t_1}) > 0$ or $\sigma(v_{t_2}) > 0$. If $\sigma(v_{t_1}) > 0$, then σ must satisfy ψ_{t_1}, β_{t_1}, and γ_{t_1}. Now, γ_{t_1} and β_{t_1} yield that $|\sigma(X^{t_1}_A)| = |\sigma(X^{t_1}_A \cap X^{t_1}_C)| + |\sigma(X^{t_1}_A \cap X^{t_1}_{\neg C})| \leq \sigma(v_{t_1}) + \sigma(v_{t_3}) \leq 3$, where the latter inequality holds because σ satisfies the conjunct $v_{t_1} + v_{t_3} \leq 3$ of $\phi_{\mathcal{E}}$. This yields a contradiction to $|\sigma(X^{t_1}_A)| \geq 5$, which is obtained from ψ_{t_1}, and thus $\sigma(v_{t_1}) = 0$. However, then we must have $\sigma(v_{t_2}) > 0$, which also leads to a contradiction: $5 \leq |\sigma(X^{t_2}_A)| \leq \sigma(v_{t_1}) + \sigma(v_{t_3}) \leq 3$. This shows that $\delta_{\mathcal{E}}$ does not have a solution in our example, which corresponds to the fact that $\mathcal{E}$ is actually inconsistent.

The next lemma shows that there is indeed a 1–1-relationship between solvability of $\delta_{\mathcal{E}}$ and consistency of $\mathcal{E}$.

LEMMA 9. *The QFBAPA formula $\delta_{\mathcal{E}}$ is of size at most exponential in the size of $\mathcal{E}$, and it is satisfiable iff $\mathcal{E}$ is consistent.*

PROOF. The at most exponential size of $\delta_{\mathcal{E}}$ is an easy consequence of the fact 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 $\mathcal{E}$. This implies that the size of $\phi_{\mathcal{E}}$ is at most exponential. For every type t, the size of $\psi_t \wedge \beta_t$ is polynomial in the size of $\mathcal{E}$ since the cardinality of $\mathcal{M}$ is linear in the size of $\mathcal{E}$ and the constraints in ψ_t not contained in α_t are obtained from successor constraints occurring in the concepts in $\mathcal{M}$. Finally, the size of γ_t is at most exponential in the size of $\mathcal{E}$ since there are at most $|\mathcal{M}|$ concepts C and at most exponentially many types t'.

Now, assume that the finite interpretation $\mathcal{I}$ is a model of $\mathcal{E}$. If we define σ as $\sigma(v_t) := |C^{\mathcal{I}}_t|$ for all types t, then we know that σ solves $\phi_{\mathcal{E}}$. Let t by a type such that $\sigma(v_t) \neq 0$. Then there is an individual $d \in \Delta^{\mathcal{I}}$ such that $d \in C^{\mathcal{I}}_t$. The semantics of $\mathcal{ALCSCC}$ then implies that we can extend σ to a solution of ψ_t by interpreting the set variables with superscript t using the role successors of d:

$$\sigma(X^t_r) := \{e \mid (d,e) \in r^{\mathcal{I}}\}, \quad \sigma(\mathcal{U}^t) := \bigcup_{r \in N_R} \sigma(X^t_r), \quad \text{and}$$
$$\sigma(X^t_C) := C^{\mathcal{I}} \cap \sigma(\mathcal{U}^t).$$

This obviously satisfies β_t, and it also yields a solution of γ_t since the following holds for all types t': $\bigcap_{C \in t'} \sigma(X^t_C) \subseteq \bigcap_{C \in t'} C^{\mathcal{I}} = C^{\mathcal{I}}_{t'}$ and thus $|\bigcap_{C \in t'} \sigma(X^t_C)| \leq |C^{\mathcal{I}}_{t'}| = \sigma(v_{t'})$.

If t is a type such that $\sigma(v_t) = 0$, then it is not necessary for σ to satisfy ψ_t, β_t, and γ_t. We can thus extend σ to the set variables with superscript t in an arbitrary way, for instance by interpreting all of them as the empty set. Overall, this show that we can use a model of $\mathcal{E}$ to define a solution σ of the QFBAPA formula $\delta_{\mathcal{E}}$.

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

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

Defining the interpretation of a role name in $\mathcal{I}$ is a bit more involved. Consider a type $t \in T_\sigma$. Then σ satisfies ψ_t, β_t, and γ_t. The solution of ψ_t yields a finite set $\sigma(\mathcal{U}^t)$ and interprets the set variables X^t_r and X^t_C as subsets of $\sigma(\mathcal{U}^t)$ such that ψ_t is satisfied. By Lemma 8, for every element e of $\sigma(\mathcal{U}^t)$ there is a unique type t_e such that $e \in \bigcap_{C \in t_e} \sigma(X^t_C)$. In addition, the fact that σ satisfies γ_t implies that, for all types t', we have $|\{e \in \sigma(\mathcal{U}^t) \mid t_e = t'\}| \leq \sigma(v_{t'})$. This show that there exists an injective mapping π_t of $\sigma(\mathcal{U}^t)$ into $\Delta^{\mathcal{I}}$ such that $\pi_t(e) = (t',j)$ implies that $t' = t_e$. Given a role name r, we now define

$$r^{\mathcal{I}} := \{((t,j),(t',j')) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists e \in \sigma(\mathcal{U}^t) : \pi_t(e) = (t',j') \wedge e \in \sigma(X^t_r)\}.$$

Note that π_t is a bijection between $\sigma(\mathcal{U}^t)$ and the role successors of $(t,j) \in \Delta^{\mathcal{I}}$. To show this it is enough to prove that, for all $e \in \sigma(\mathcal{U}^t)$, there is a role $r \in N_R$ such that $e \in \sigma(X^t_r)$. This is an immediate consequence of the fact that σ satisfies α_t.

We want to show that the following holds for all types $t \in T_\sigma$ and $j, 1 \leq j \leq \sigma(v_t)$: $(t,j) \in C^{\mathcal{I}}_t$.

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

It remains to show $(t, j) \in C_t^{\mathcal{I}}$. For this it is sufficient to show the following by *induction on the structure of* C: 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

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

- Let $C = A$ for a concept name A. Then (2) 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. The case where $C = D \sqcup E$ can be handled similarly.

- $C = \text{succ}(c)$ be a successor restriction. First, assume that $C \in t$. Then the translation c' of c using set variables with superscript t is a conjunct in ψ_t. Consequently, σ satisfies this translation c'. We know that the mapping π_t is a bijection between $\sigma(\mathcal{U}^t)$ and the set $ars^{\mathcal{I}}(t, j)$ of role successors of (t, j) in $\mathcal{I}$. Because of our definition of the interpretation of roles in $\mathcal{I}$, we know more precisely that π_t is also a bijection between $\sigma(X_r^t)$ and $r^{\mathcal{I}}(t, j)$.

  It is thus sufficient to show that π_t is a bijection between $\sigma(X_D^t)$ and $D^{\mathcal{I}} \cap ars^{\mathcal{I}}(t, j)$ for all concept descriptions D occurring in the constraint c. In fact, then π_t is an "isomorphism" between $\sigma(\mathcal{U}^t)$ and $ars^{\mathcal{I}}(t, j)$, and thus the fact that σ satisfies c' implies that $(t, j) \in \text{succ}(c)^{\mathcal{I}}$.

  By induction, we know that the equivalence (2) holds for the concept descriptions D occurring in c. Let $e \in \sigma(X_D^t)$ and $\pi_t(e) = (t', i')$. By the definiton of π_t, we have $t' = t_e$ where $t_e = \{E \in \mathcal{M} \mid e \in \sigma(X_E^t)\}$, which yields $D \in t'$. By induction, we obtain $(t', i') \in D^{\mathcal{I}}$, and we already know that $(t', i') \in ars^{\mathcal{I}}(t, j)$. This shows that π_t is an injective mapping from $\sigma(X_D^t)$ into $D^{\mathcal{I}} \cap ars^{\mathcal{I}}(t, j)$.

  To show surjectivity, assume that we have an element $(t', i') \in D^{\mathcal{I}} \cap ars^{\mathcal{I}}(t, j)$. By induction $(t', i') \in D^{\mathcal{I}}$ yields $D \in t'$. In addition, $(t', i') \in ars^{\mathcal{I}}(t, j)$ implies that there is an $e \in \sigma(\mathcal{U}^t)$ such that $\pi_t(e) = (t', i')$. Consequently, $t' = t_e$, which together with $D \in t'$ implies that $e \in \sigma(X_D^t)$, and thus establishes the desired surjectivity result.

  Second, assume that $C \notin t$. Then $\neg\,\text{succ}(c) \in t$, and thus the translation $\neg c'$ of $\neg c$ using set variables with superscript t is a conjunct in ψ_t. We can now proceed as in the first case, but with $\neg c$ and $\neg c'$ in place of c and c'.

This completes the proof of (2) and thus the proof of the lemma.

Since satisfiability of QFBAPA formulae can be decided within NP even for binary coding of numbers [15], this lemma shows that consistency of $\mathcal{ALCSCC}$ ECBoxes can be decided within NExpTime. Together with the known NExpTime lower bound for consistency of $\mathcal{ALC}$ ECBoxes [6], this yields:

Theorem 10. *Consistency of ECBoxes with numbers encoded in unary or binary is NExpTime-complete in* $\mathcal{ALCSCC}$.

4. RESTRICTING THE CONSTRAINTS

For $\mathcal{ALC}$, a restricted notion of ECBoxes was introduced in [6], and it was show that this restriction lowers the complexity of the consistency problem from NExpTime to ExpTime. We will show below that the same is true for $\mathcal{ALCSCC}$. *Restricted cardinality constraints on* $\mathcal{ALCSCC}$ *concepts* are defined as follows:

Definition 11. *Let* N_C *and* N_R *be finite, disjoint sets concept names and of role names, respectively.*

- Restricted $\mathcal{ALCSCC}$ cardinality constraints *over the signature* (N_C, N_R) *are of the form*

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

  where $C_1, \ldots, C_{k+\ell}$ *are* $\mathcal{ALCSCC}$ *concept descriptions over* (N_C, N_R) *and* $N_1, \ldots, N_{k+\ell}$ *are integer constants.*

- *A* restricted $\mathcal{ALCSCC}$ cardinality box (RCBox) *over* (N_C, N_R) *is a conjunction of restricted* $\mathcal{ALCSCC}$ *cardinality constraints over* (N_C, N_R).

Restricted cardinality constraints cannot express cardinality restrictions on concepts (CRs), as considered in [3, 19]. In fact, it is easy to see that models of RCBoxes are closed under disjoint union, and thus always have models of arbitrarily large finite cardinality, whereas the CR $(\leq 5\top)$ only has models of cardinality at most 5.

Proposition 12. *Restricted cardinality constraints cannot express cardinality restrictions on concepts.*

Restricted cardinality constraints can, however, express CIs since the constraint $|C \sqcap D| = |C|$ is only satisfied in interpretations $\mathcal{I}$ in which $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds. Consequently, as already stated in [6], ExpTime hardness of consistency w.r.t. TBoxes in $\mathcal{ALC}$ implies that consistency of restricted $\mathcal{ALC}$ cardinality boxes is ExpTime hard. Since no large number are needed when expressing CIs using restricted cardinality constraints, this hardness result also holds for unary coding of numbers. Using the fact that $\mathcal{ALC}$ is a sub-logic of $\mathcal{ALCSCC}$, we thus obtain the following complexity lower bound for $\mathcal{ALCSCC}$ RCBoxes.

Proposition 13. *Consistency of* $\mathcal{ALCSCC}$ *RCBoxes is ExpTime-hard, independently of whether numbers are encoded in unary or binary.*

In the next section we show that ExpTime is indeed the optimal worst-case complexity for the consistency problem of restricted $\mathcal{ALCSCC}$ cardinality constraints.

5. CONSISTENCY OF RCBOXES

We show the ExpTime upper bound for numbers encoded in binary using type elimination, where the notion of augmented type from [1] is used, and a second step for removing types is added to take care of the RCBox, similar to what is done in the type elimination procedure for $\mathcal{ALC}$ RCBoxes in [6].

The ExpTime upper bound for our procedure depends on the following lemma, which applies in our setting due to the special form of RCBoxes.

Lemma 14. *Let ϕ be a system of linear inequalities consisting of $A \cdot \boldsymbol{v} \geq \boldsymbol{0}$, $\boldsymbol{v} \geq \boldsymbol{0}$, and $B \cdot \boldsymbol{v} \geq \boldsymbol{1}$, 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 even if the numbers occurring in the system are encoded in binary.*
2. *The solutions of ϕ are closed under addition, and in particular we have the following: if $\{v_1, \ldots, v_k\}$ is a set of variables such that, for all v_i $(1 \leq i \leq k)$, ϕ has a solution σ_i in which v_i has a non-zero value, then there is a non-negative integer solution σ of ϕ such that $\sigma(v_i) \geq 1$ for all $i, 1 \leq i \leq k$.*

A formal proof of this lemma can be found in [6]. Intuitively, closure under addition is easily seen to hold due to the special form of the inequalities considered in the lemma. On the one hand, this yields that the existence of a rational solution (which can be tested in polynomial time [17]) implies the existence of an integer solution. On the other hand, it also implies the existence of a solution as required in the second part of the lemma.

Another important ingredient of our ExpTime procedure are augmented types, which have been introduced in [1] to show that satisfiability in $\mathcal{ALCSCC}$ w.r.t. concept inclusions is in ExpTime.

Augmented types consider not just the concepts to which a single individual belongs, but also the Venn regions to which its role successors belong. Given a type t for $\mathcal{R}$, we consider the corresponding QFBAPA formula ϕ_t, which is induced by the (possibly negated) successor constraints occurring in t. We conjoin to this formula the set constraint $X_{r_1} \cup \ldots \cup X_{r_n} = \mathcal{U}$, where $N_R = \{r_1, \ldots, r_n\}$ and we assume without loss of generality that N_R contains only the role names occurring in $\mathcal{R}$. For the resulting formula ϕ'_t, we compute the number N_t that bounds the number of Venn regions that need to be non-empty in a solution of ϕ'_t (see Lemma 1).

Definition 15. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ RCBox. An* augmented type (t, V) *for $\mathcal{R}$ consists of a type t for $\mathcal{R}$ together with a set of Venn region V such that $|V| \leq N_t$ and the formula ϕ'_t has a solution in which exactly the Venn regions in V are non-empty.*

The existence of a solution of ϕ'_t in which exactly the Venn regions in V are non-empty can obviously be checked (within NP) by adding to ϕ'_t conjuncts that state non-emptiness of the Venn regions in V and the fact that the union of these Venn regions is the universal set (see the description of the PSpace algorithm in the proof of Theorem 1 in [1]). Another easy to show observation is that there are only exponentially many augmented types (see the accompanying technical report of [1] for a proof of the following lemma).

Lemma 16. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ RCBox. The set of augmented types for $\mathcal{R}$ contains at most exponentially many elements in the size of $\mathcal{R}$ and it can be computed in exponential time.*

Basically, type elimination starts with the set of all augmented types, and then successively eliminates augmented types (i) whose Venn regions are not realized by the currently available augmented types, or (ii) whose first component is forced to be empty by the constraints in $\mathcal{R}$. To make the first reason for elimination more precise, assume that $\mathcal{A}$ is a set of augmented types and that v is a Venn region. The Venn region v yields a set of concept descriptions S_v that contains, for every set variable X_D occurring in v, the element D in case v contains X_D and the element $\neg D$ in case v contains X_D^c. It is easy to see that S_v is actually a subset of $\mathcal{M}$ (modulo removal of double negation). We say that v is *realized by $\mathcal{A}$* if there is an augmented type $(t, V) \in \mathcal{A}$ such that $S_v \subseteq t$.

Algorithm 17. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ RCBox. The following steps decide consistency of $\mathcal{R}$:*

1. *Compute the set $\mathcal{M}$ consisting of all subdescriptions of $\mathcal{R}$ as well as the negations of these subdescriptions, and continue with the next step.*
2. *Based on $\mathcal{M}$, compute the set $\mathcal{A}$ of all augmented types for $\mathcal{R}$, and continue with the next step.*
3. *If the current set $\mathcal{A}$ of augmented types is empty, then the algorithm* fails. *Otherwise, check whether $\mathcal{A}$ contains an element (t, V) such that not all the Venn regions in V are realized by $\mathcal{A}$. If there is no such element (t, V) in $\mathcal{A}$, then continue with the next step. Otherwise, let (t, V) be such an element, and set $\mathcal{A} := \mathcal{A} \setminus \{(t, V)\}$. Continue with this step, but now using the new current set of augmented types.*
4. *Let $T_\mathcal{A} := \{t \mid$ there is V such that $(t, V) \in \mathcal{A}\}$, and let $\phi_{T_\mathcal{A}}$ be obtained from $\mathcal{R}$ by replacing each $|C|$ in $\mathcal{R}$ with $\sum_{t \in T_\mathcal{A} \text{ s.t. } C \in t} v_t$ and adding $v_t \geq 0$ for each $t \in T_\mathcal{A}$. Check whether $T_\mathcal{A}$ contains an element t such that $\phi_{T_\mathcal{A}} \wedge v_t \geq 1$ has no solution. If this is the case for t, then remove all augmented types of the form $(t, \cdot)$ from $\mathcal{A}$, and continue with the previous step. If no type t is removed in this step, then the algorithm* succeeds.

First, we show soundness of this algorithm.

Lemma 18. *If Algorithm 17 succeeds on input $\mathcal{R}$, then the $\mathcal{ALCSCC}$ RCBox $\mathcal{R}$ is consistent.*

Proof. Assume that the algorithm succeeds on input $\mathcal{R}$, and let $\mathcal{A}$ be the final set of augmented types when the algorithm stops successfully. We show how $\mathcal{A}$ can be used to construct a model $\mathcal{I}$ of $\mathcal{R}$.

For this construction, we first consider the formula $\phi_{T_\mathcal{A}}$, which is obtained from $\mathcal{R}$ by replacing each $|C|$ in $\mathcal{R}$ with $\sum_{t\in T_\mathcal{A} \text{ s.t. } C\in t} v_t$ and adding $v_t \geq 0$ for each $t \in T_\mathcal{A}$. Note that, due to the special form of RCBoxes, we know that this yields a system of linear inequalities of the form $A \cdot \boldsymbol{v} \geq \mathbf{0}$, $\boldsymbol{v} \geq \mathbf{0}$. Since the algorithm has terminated successfully, we know for all $t \in T_\mathcal{A}$ that the formula $\phi_{T_\mathcal{A}} \wedge v_t \geq 1$ has a solution. By Lemma 14(2) this implies that $\phi_{T_\mathcal{A}}$ has a solution in which all variables v_t for $t \in T_\mathcal{A}$ have a value ≥ 1 and all variables v_t with $t \notin T_\mathcal{A}$ have value 0. In addition, given an arbitrary number $N \geq 1$, we know that there is a solution σ_N of $\phi_{T_\mathcal{A}}$ such that $\sigma_N(v_t) \geq 1$ and $N|\sigma_N(v_t)$ holds for all $t \in T_\mathcal{A}$. To see this, note that we can just multiply with N a given solution satisfying the properties mentioned in the previous sentence.

We use the augmented types in $\mathcal{A}$ to determine the right N:

- For each augmented type (t, V), we know that the formula ϕ'_t has a solution where exactly the Venn regions in V are non-empty (see Definition 15). Assume that this solution assigns a set of cardinality $k_{(t,V)}$ to the universal set.
- For each $t \in T_\mathcal{A}$, let n_t be the cardinality of the set $\{V \mid (t, V) \in \mathcal{A}\}$, i.e., the number of augmented types in $\mathcal{A}$ that have t as their first component.

We now define N as

$$N := (\max\{k_{(t,V)} \mid (t, V) \in \mathcal{A}\}) \cdot \prod_{t\in T_\mathcal{A}} n_t,$$

and use the solution σ_N of $\phi_{T_\mathcal{A}}$ to construct a finite interpretation $\mathcal{I}$ as follows. The domain of $\mathcal{I}$ is defined as

$$\Delta^\mathcal{I} := \{(t, V)^i \mid (t, V) \in \mathcal{A} \text{ and } 1 \leq i \leq \sigma_N(v_t)/n_t\}.$$

Note that $\sigma_N(v_t)/n_t$ is a natural number since $N|\sigma_N(v_t)$ implies $n_t|\sigma_N(v_t)$. In addition, $\Delta^\mathcal{I} \neq \emptyset$ because $\mathcal{A} \neq \emptyset$ and $\sigma_N(v_t)/n_t \geq 1$ since $\sigma_N(v_t) \geq 1$. Moreover, for each type $t \in T_\mathcal{A}$, the set $\{(t, V)^i \mid (t, V)^i \in \Delta^\mathcal{I}\}$ has cardinality $\sigma_N(v_t)$.

The interpretation of the concept names A is based on the occurrence of these names in the first component of an augmented type, i.e., $A^\mathcal{I} := \{(t, V)^i \in \Delta^\mathcal{I} \mid A \in t\}$.

Defining the interpretation of the role names is again more tricky. Obviously, it is sufficient to define, for each role name $r \in N_R$ and each $d \in \Delta^\mathcal{I}$, the set $r^\mathcal{I}(d)$. Thus, consider an element $(t, V)^i \in \Delta^\mathcal{I}$. Since (t, V) is an augmented type in $\mathcal{A}$, the formula ϕ'_t has a solution σ in which exactly the Venn regions in V are non-empty, and which assigns a set of cardinality $m := k_{(t,V)}$ to the universal set. In addition, each Venn region $w \in V$ is realized by an augmented type $(t^w, V^w) \in \mathcal{A}$. Assume that the solution σ assigns the finite set $\{d_1, \ldots, d_m\}$ to the set term $\mathcal{U}$. We consider an injective mapping π of $\{d_1, \ldots, d_m\}$ into $\Delta^\mathcal{I}$ such that the following holds for each element d_j of $\{d_1, \ldots, d_m\}$: if d_j belongs to the Venn region $w \in V$, then $\pi(d_j) = (t^w, V^w)^\ell$ for some $1 \leq \ell \leq \sigma_N(v_{t^w})/n_{t^w}$. Such a bijection exists since $\sigma_N(v_{t^w})/n_{t^w} \geq \max\{k_{(t',V')} \mid (t', V') \in \mathcal{A}\} \geq k_{(t,V)} = m$. We now define

$$r^\mathcal{I}((t, V)^i) := \{\pi(d_j) \mid d_j \in \sigma(X_r)\}.$$

Soundness of Algorithm 17 is now an easy consequence of the following claim:

Claim: *For all $C \in \mathcal{M}$, $(t, V) \in \mathcal{A}$, and $i, 1 \leq i \leq \sigma_N(v_t)/n_t$ we have $C \in t$ iff $(t, V)^i \in C^\mathcal{I}$.*

We prove the claim by induction on the size of C:

- The cases $C = A$, $C = \neg D$, $C = D_1 \sqcap D_2$, and $C = D_1 \sqcup D_2$ can be handled as in the proof of (2) in the proof of Lemma 9.
- Now assume that $C = \mathit{succ}(c)$ for a set or cardinality constraint c.
  - If $C \in t$, then this constraint is part of the QFBAPA formula ϕ'_t obtained from t, and thus satisfied by the solution σ of ϕ'_t used to define the role successors of $(t, V)^i$. According to this definition, there is a 1–1 correspondence between the elements of $\sigma(\mathcal{U})$ and the role successors of $(t, V)^i$. This bijection π also respects the assignment of subsets of $\sigma(\mathcal{U})$ to set variables of the form X_r (for $r \in N_R$) and X_D (for concept descriptions D) occurring in ϕ'_t, i.e.,

    $(*)$ $\quad d_j \in \sigma(X_r)$ iff $\pi(d_j) \in r^\mathcal{I}((t, V)^i)$ and
    $\quad d_j \in \sigma(X_D)$ iff $\pi(d_j) \in D^\mathcal{I}$.

    Once $(*)$ is shown it is clear that $(t, V)^i \in \mathit{succ}(c)^\mathcal{I}$. In fact, the translation ϕ_c of c, where r is replaced by X_r and D by X_D, is a conjunct in ϕ'_t and thus σ satisfies ϕ_c. Now $(*)$ shows that (modulo the application of the bijection π), when checking whether $(t, V)^i \in \mathit{succ}(c)^\mathcal{I}$, roles r and concepts D in ϕ_c are interpreted in the same way as the set variables X_r and X_D in c, respectively. Thus the fact that σ satisfies ϕ_c implies that the role successors of $(t, V)^i$ satisfy c, i.e., $(t, V)^i \in \mathit{succ}(c)^\mathcal{I}$ holds.

    For role names r, property $(*)$ is immediate by the definition of $r^\mathcal{I}((t, V)^i)$.

    Now consider a concept description D such that X_D occurs in ϕ'_t. Then D occurs in c, and is thus smaller than C, which means that we can apply induction to it.

    If $d_j \in \sigma(X_D)$, then the Venn region w to which d_j belongs contains X_D positively. Consequently, S_w contains D, and the augmented type (t^w, V^w) realizing w satisfies $D \in t^w$. By induction, we obtain $\pi(d_j) = (t^w, V^w)^\ell \in D^\mathcal{I}$.

    Conversely, assume that $\pi(d_j) = (t^w, V^w)^\ell \in D^\mathcal{I}$, where w is the Venn region to which d_j belongs w.r.t. σ. By induction, we obtain $D \in t^w$, and thus the Venn region w contains X_D positively. Since d_j belongs to this Venn region, we obtain $d_j \in \sigma(X_D)$.
  - The case where $C \notin t$ can be treated similarly. In fact, in this case the constraint $\neg c$ is part of the QFBAPA formula ϕ'_t obtained from t, and we can employ the same argument as above, just using $\neg c$ instead of c.

This finishes the proof of the claim. As an easy consequence of this claim we have for all C occurring in $\mathcal{R}$ that

$$C^{\mathcal{I}} = \{(t,V)^i \mid C \in t, (t,V) \in \mathcal{A}, \text{ and } 1 \leq i \leq \sigma_N(v_t)/n_t\}.$$

Consequently,

$$|C^{\mathcal{I}}| = \sum_{t \in T_{\mathcal{A}} \text{ s.t. } C \in t} \sigma_N(v_t),$$

which shows that $\mathcal{I}$ satisfies $\mathcal{R}$ since σ_N solves $\phi_{T_{\mathcal{A}}}$.

Next, we show completeness of the algorithm.

Lemma 19. *If the $\mathcal{ALCSCC}$ RCBox $\mathcal{R}$ is consistent, then Algorithm 17 succeeds on input $\mathcal{R}$.*

Proof. Assume that $\mathcal{I}$ is a model of $\mathcal{R}$. Consider the set of all types of elements of $\mathcal{I}$, i.e.,

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

As mentioned below Definition 7, the elements of $T_{\mathcal{I}}$ are indeed types according to Definition 7. In addition, since $\Delta^{\mathcal{I}} \neq \emptyset$, we also know that $T_{\mathcal{I}} \neq \emptyset$.

First, note that no element of $T_{\mathcal{I}}$ can be removed in Step 4 of our algorithm. This is an easy consequence of the following observation. Let T be a set of types such that $T_{\mathcal{I}} \subseteq T$, and let ϕ_T be obtained from $\mathcal{R}$ by replacing each $|C|$ in $\mathcal{R}$ with $\sum_{t \in T \text{ s.t. } C \in t} v_t$ and adding $v_t \geq 0$ for each $t \in T$. Since $\mathcal{I}$ is a model of $\mathcal{R}$, it is easy to see that ϕ_T has a solution that also satisfies $v_t \geq 1$ for all $t \in T_{\mathcal{I}}$.

Regarding Step 3 of our algorithm, we want to show that for every type $t \in T_{\mathcal{I}}$ there is at least one set of Venn regions V such that the augmented type (t,V) is not removed in this step. To this purpose, let us now extend the types in $T_{\mathcal{I}}$ by adding appropriate Venn regions as second components. Consider $t := t_{\mathcal{I}}(d)$ for an element $d \in \Delta^{\mathcal{I}}$. Then the QFBAPA formula ϕ'_t corresponding to t has a solution σ in which the universal set $\mathcal{U}$ consists of all the role successors of d, and the other set variables are assigned sets according to the interpretations of roles and concept descriptions in the model $\mathcal{I}$. Let $\{d_1, \ldots, d_m\} = \sigma(\mathcal{U})$ be the set of all role successors of d, and w_i the Venn region to which d_i belongs w.r.t. σ. By Lemma 1, there is a solution σ' of ϕ'_t such that the set V of non-empty Venn regions w.r.t. σ' has cardinality $\leq N_t$ and each of these non-empty Venn regions in V is one of the Venn regions w_i, i.e., $V \subseteq \{w_1 \ldots, w_m\}$. By construction, (t,V) is an augmented type. Let $\mathcal{A}_{\mathcal{I}}$ denote the set of augmented types obtained by extending the types in $T_{\mathcal{I}}$ in this way for every $d \in \Delta^{\mathcal{I}}$. By construction, for every $t \in T_{\mathcal{I}}$ there is a set of Venn regions V such that $(t,V) \in \mathcal{A}_{\mathcal{I}}$. In particular, this yields $\mathcal{A}_{\mathcal{I}} \neq \emptyset$ and $T_{\mathcal{A}_{\mathcal{I}}} = T_{\mathcal{I}}$.

Next, we show that the Venn regions occurring in some augmented type in $\mathcal{A}_{\mathcal{I}}$ are realized by $\mathcal{A}_{\mathcal{I}}$. Thus, let (t,V) be an augmented type constructed from a type $t = t_{\mathcal{I}}(d)$ as described above, and let $w \in V$ be a Venn region occurring in this augmented type. Then there is a role successor d_i of d such that d_i belongs to the Venn region $w = w_i$ w.r.t. the solution σ of ϕ'_t induced by $\mathcal{I}$. We know that $d_i \in D^{\mathcal{I}}$ for all $D \in S_w$, and thus $S_w \subseteq t_{\mathcal{I}}(d_i)$. Since $\mathcal{A}_{\mathcal{I}}$ contains an augmented type with first component $t_{\mathcal{I}}(d_i)$, this shows that w is realized by $\mathcal{A}_{\mathcal{I}}$.

We claim that, during the run of Algorithm 17, we always have $\mathcal{A}_{\mathcal{I}} \subseteq \mathcal{A}$ and $T_{\mathcal{I}} \subseteq T_{\mathcal{A}}$. Obviously, this is true after Step 2 of our algorithm. In addition, in Step 3 of our algorithm, no element of $\mathcal{A}_{\mathcal{I}}$ can be removed since we have seen that the Venn regions occurring in some augmented type in $\mathcal{A}_{\mathcal{I}}$ are realized by $\mathcal{A}_{\mathcal{I}}$. Finally, we have also seen above that in Step 4 of our algorithm, no element of $T_{\mathcal{I}} = T_{\mathcal{A}_{\mathcal{I}}}$ can be removed.

Since $\mathcal{A}_{\mathcal{I}}$ is non-empty, this shows that the algorithm cannot fail on input $\mathcal{R}$, and thus must succeed. This completes the proof of completeness.

Theorem 20. *The consistency problem for RCBoxes in $\mathcal{ALCSCC}$ is ExpTime-complete both for unary and binary coding of numbers.*

Proof. Since ExpTime lower bounds for both cases were already stated in Proposition 13, it remains to show the ExpTime upper bound for the case of binary coding of numbers. In addition, since we have already shown soundness and completeness of Algorithm 17, it is sufficient to prove that this algorithm indeed runs in exponential time.

To see this, first note that, according to Lemma 16, there are only exponentially many augmented types, and they can be computed in exponential time. Thus, the first two steps of the algorithm take only exponential time. In addition, the iteration between Steps 3 and 4 can happen only exponentially often since in each iteration at least one augmented type is removed.

A single Step 3 takes only exponential time since for each of the exponentially many augmented types (t,V), only exponentially many other augmented types need to be considered.

Finally, a single Step 4 takes only exponential time. In fact, we need to consider exponentially many systems of linear inequalities $\phi_{T_{\mathcal{A}}} \wedge v_t \geq 1$. Each of these systems may be of exponential size, but its solvability can be tested in time that is polynomial in this size (according to Lemma 14), and thus exponential in the size of the input.

6. CONCLUSION

In two previous papers we have shown how to use QFBAPA constraints to extend the expressive power of number restrictions, on the hand, and of cardinality restrictions on concepts, on the other hand. In the present paper, we have demonstrated that these two extensions can be combined without increasing the complexity of reasoning. In fact, reasoning w.r.t. ECBoxes (RCBoxes) in $\mathcal{ALCSCC}$, the extension of $\mathcal{ALC}$ with more expressive number restrictions, has the same complexity as in $\mathcal{ALC}$: NExpTime-complete for ECBoxes and ExpTime-complete for RCBoxes. This combination is non-trivial since role successors required by local constraints might actually be prevented to exist by global constraints. For ECBoxes, we have shown that this can be taken care of by inequalities that basically relate cardinalities of global types with cardinalities of local Venn regions (see the definition of the formulae γ_t in (1)). For RCBoxes, stating such inequalities explicitly is not necessary since in this restricted case the cardinalities of global types can be made as large as is needed to satisfy the local constraints (see the definition of the solution σ_N in the proof of Lemma 18). Technically, in both cases the fact that the interaction between global and local constraints is handled appropriately shows up in the definition of the interpretation of roles in the soundness proofs, where we see that sufficiently many copies of the (augmented) types belong to the domain of the interpretation.

The combined logic can, for instance, be used to check the correctness of statistical statements. For example, if a German car company claims that they have produced more than N cars in a certain year, and P% of the tires used for their cars were produced by Betteryear, this may be contradictory to a statement of Betteryear that they have sold less than M tires in Germany. This information can be expressed using $\mathcal{ALCSCC}$ ECBoxes, and thus the contradiction can be found using our consistency algorithm. It would not have been possible to express this using $\mathcal{ALC}$ ECBoxes since stating how many tires a car has requires local number restrictions.

In [2], we have considered a setting that is even more expressive than $\mathcal{ALCSCC}$ ECBoxes. To be more precise, in the DL introduced in [2], local and global constraints can interact directly. For example, the concept description

$$succ(|likes \cap Cat| = |Cat|)$$

describes cat lovers, i.e., individuals d that like *all* cats. In contrast to the semantics of $\mathcal{ALCSCC}$ introduced in the present paper, *Cat* is here interpreted by all cats in the interpretation domain, and not just by the cats that are linked to the individual d via a role. A local interpretation as in $\mathcal{ALCSCC}$ can be achieved by intersection with a role, as in *likes* $\cap$ *Cat*. It is shown in [2] that satisfiability of concepts in this logic has the same complexity as the consistency problem for $\mathcal{ALCSCC}$ ECBoxes. However, conjunctive query entailment in this setting is undecidable. In contrast, conjunctive query entailment w.r.t. $\mathcal{ALCSCC}$ RCBoxes is shown to be decidable (in ExpTime).

7. ACKNOWLEDGEMENTS

This work was partially supported by the German Research Foundation (DFG) within the Research Unit 1513 Hybris.

8. REFERENCES

[1] Franz Baader. A new description logic with set constraints and cardinality constraints on role successors. In Clare Dixon and Marcelo Finger, editors, *Proc. of the 11th Int. Symposium on Frontiers of Combining Systems (FroCoS'17)*, volume 10483 of *Lecture Notes in Computer Science*, pages 43–59, Brasília, Brazil, 2017. Springer-Verlag.

[2] Franz Baader, Bartosz Bednarczyk, and Sebastian Rudolph. Satisfiability checking and conjunctive query answering in description logics with global and local cardinality constraints. In Mantas Simkus and Grant E. Weddell, editors, *Proc. of the 32nd Int. Workshop on Description Logics (DL'19)*, volume 2373 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

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

[4] 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.

[5] Franz Baader and Filipo de Bortoli. On the expressive power of description logics with cardinality constraints on finite and infinite sets. In Andreas Herzig and Andrei Popescu, editors, *Proc. of the 12th Int. Symposium on Frontiers of Combining Systems (FroCoS'19)*, volume 11715 of *Lecture Notes in Computer Science*. Springer-Verlag, 2019.

[6] Franz Baader and Andreas Ecke. Extending the description logic $\mathcal{ALC}$ with more expressive cardinality constraints on concepts. In *Proc. of the 3rd Global Conf. on Artificial Intelligence (GCAI'17)*, volume 50 of *EPiC Series in Computing*, pages 6–19. EasyChair, 2017.

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

[8] 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.

[9] Stéphane Demri and Denis Lugiez. Complexity of modal logics with Presburger constraints. *J. Applied Logic*, 8(3):233–252, 2010.

[10] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, second edition edition, 1999.

[11] 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.

[12] 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.

[13] 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.

[14] Viktor Kuncak, Ruzica Piskac, and Philippe Suter. Ordered sets in the calculus of data structures. In Anuj Dawar and Helmut Veith, editors, *Proc. of the 19th Annual Conf. of the EACSL (CSL-10)*, volume 6247 of *Lecture Notes in Computer Science*, pages 34–48. Springer, 2007.

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

[16] 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.

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

[18] Stephan Tobies. A PSPACE algorithm for graded modal logic. In Harald Ganzinger, editor, *Proc. of the 16th Int. Conf. on Automated Deduction (CADE'99)*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 52–66. Springer-Verlag, 1999.

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

[20] Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation.* PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

[21] Kuat Yessenov, Ruzica Piskac, and Viktor Kuncak. Collections, cardinalities, and relations. In Gilles Barthe and Manuel V. Hermenegildo, editors, *Proc. of the 11th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'10)*, volume 5944 of *Lecture Notes in Computer Science*, pages 380–395. Springer-Verlag, 2010.