Satisfiability Checking and Conjunctive Query Answering in Description Logics with Global and Local Cardinality Constraints

Franz Baader[1], Bartosz Bednarczyk[12], and Sebastian Rudolph[1]

[1] Faculty of Computer Science, TU Dresden, Germany
firstname.lastname@tu-dresden.de
[2] Institute of Computer Science, University of Wrocław, Poland
bartosz.bednarczyk@cs.uni.wroc.pl

Abstract. We consider an expressive description logic (DL) in which the global and local cardinality constraints introduced in previous papers can be mixed. On the one hand, we show that this does not increase the complexity of satisfiability checking and other standard inference problems. On the other hand, conjunctive query entailment in this DL turns out to be undecidable. We prove that decidability of querying can be regained if global and local constraints are not mixed and the global constraints are appropriately restricted.

1 Introduction

DLs that can express both local cardinality constraints (i.e., constraints concerning the role successors of specific individuals) and global cardinality constraints (i.e., constraints on the overall cardinality of concepts) 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. Such statistical information may, of course, also influence the answers to queries. If we know that the car company VMW uses only tires from Betteryear or Badmonth, but the statistical information allows us to conclude that another car company has actually bought all the tires sold by Betteryear, then we know that the cars sold by VMW all have tires produced by Badmonth. This motivates investigating DLs with expressive cardinality constraints, and to consider not just standard inferences such as satisfiability checking for these DLs, but also query answering.

In two previous publications we have, on the one hand, extended the DL $\mathcal{ALCQ}$ by more expressive number restrictions using cardinality and set constraints expressed in the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) [10]. In the resulting DL $\mathcal{ALCSCC}$, which was introduced and investigated in [1], cardinality and set constraints are applied locally, i.e., they refer to the role successors of an individual under consideration. It was shown in [1] that pure concept satisfiability in $\mathcal{ALCSCC}$ is a PSPACE-complete problem, and concept satisfiability w.r.t. a general TBox is EXPTIME-complete. This shows that the more expressive number restrictions do not increase the complexity of reasoning since reasoning in $\mathcal{ALCQ}$ has the same complexity [17,19].

On the other hand, we have extended the terminological formalism of the well-known description logic $\mathcal{ALC}$ from general TBoxes to more general cardinality constraints expressed in QFBAPA [4], which we called extended cardinality constraints (ECBoxes). These constraints are global since they refer to all individuals in the interpretation domain. It was shown in [4] that reasoning w.r.t. ECBoxes is in NEXPTIME even if the numbers occurring in the constraints are encoded in binary. A NEXPTIME lower bound follows from a result of Tobies [18] for a restricted form of cardinality constraints, where the cardinality of a concept can only be compared with a fixed number. This complexity can be lowered to EXPTIME if a restricted form of cardinality constraints (RCBoxes) is used. Such RCBoxes are still powerful enough to express statistical knowledge bases [13].

An obvious way to generalize these two approaches is to combine the two extensions, i.e., to consider extended cardinality constraints, but now on $\mathcal{ALCSCC}$ concepts rather than just $\mathcal{ALC}$ concepts. This combination was investigated in [2], where a NEXPTIME upper bound was established for reasoning in $\mathcal{ALCSCC}$ w.r.t. ECBoxes. It is also shown in [2] that reasoning w.r.t. RCBoxes stays in ExpTime also for $\mathcal{ALCSCC}$.

Here we go one step further by allowing for a tighter integration of global and local constraints. The resulting logic, which we call $\mathcal{ALCSCC}^{++}$, allows, for example, to relate the number of role successors of a given individual with the overall number of elements of a certain concept. We show that, from a worst-case complexity point of view, this extended expressivity comes for free if we consider classical reasoning problems. Concept satisfiability in $\mathcal{ALCSCC}^{++}$ has the same complexity as in $\mathcal{ALC}$ and $\mathcal{ALCSCC}$ with global cardinality constraints: it is NEXPTIME-complete. Yet, for effective conjunctive query answering this logic turns out to be too expressive. We show that conjunctive query entailment w.r.t. $\mathcal{ALCSCC}^{++}$ knowledge bases is, in fact, undecidable. In contrast, we can show that conjunctive query entailment w.r.t. $\mathcal{ALCSCC}$ RCBoxes is decidable.

We assume the reader to be sufficiently familiar with all the standard notions of description logics [3,5,15].

2 The logic $\mathcal{ALCSCC}^{++}$

As in [1,4], we use the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) to express our constraints. 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 *set constraints*, which are equality and inclusion constraints of the form $s = t, s \subseteq t$, where s, t are set terms. *Presburger Arithmetic*

(PA) expressions are built from integer constants and set cardinalities $|s|$ using addition as well as multiplication with an integer constant. They can be used to form *cardinality 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 *QF-BAPA formula* is a Boolean combination of set and cardinality constraints. A *solution* σ of a QFBAPA formula ϕ assigns a finite set $\sigma(\mathcal{U})$ to $\mathcal{U}$ and subsets of $\sigma(\mathcal{U})$ to set variables such that ϕ is satisfied by this assignment (see [1] for more details). 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.

We are now ready to define our new logic, which we call $\mathcal{ALCSCC}^{++}$ to indicate that it is an extension of the logic $\mathcal{ALCSCC}$ introduced in [1]. When defining the semantics of $\mathcal{ALCSCC}^{++}$, we restrict the attention to *finite* interpretations to ensure that cardinalities of concept descriptions are always well-defined non-negative integers.

Definition 1. *Given disjoint* finite *sets N_C and N_R of* concept names *and* role names*, respectively, $\mathcal{ALCSCC}^{++}$* concept descriptions *(short:* concepts*) are Boolean combinations of concept names and* constraint expressions*, where a constraint expression is of the form $sat(c)$ for a set constraint or a cardinality constraint c that uses role names and $\mathcal{ALCSCC}^{++}$ concept descriptions in place of set variables. As usual, we use $\top$ (*top*) and $\bot$ (*bottom*) as abbreviations for $A \sqcup \neg A$ and $A \sqcap \neg A$, respectively.*

A finite interpretation *of N_C and N_R consists of a finite, non-empty 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 $\Delta^{\mathcal{I}}$. For a given element $d \in \Delta^{\mathcal{I}}$ we define $r^{\mathcal{I}}(d) := \{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\}$. The interpretation function $\cdot^{\mathcal{I}}$ is inductively extended to $\mathcal{ALCSCC}^{++}$ concept descriptions by interpreting the Boolean operators as usual, and the constraint expressions as follows:*

$sat(c)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid$ the mapping $\cdot^{\mathcal{I}_d}$ satisfies $c\}$, where $\cdot^{\mathcal{I}_d}$ maps
- *$\emptyset$ to $\emptyset^{\mathcal{I}_d} := \emptyset$ and $\mathcal{U}$ to $\mathcal{U}^{\mathcal{I}_d} := \Delta^{\mathcal{I}}$,*
- *the $\mathcal{ALCSCC}^{++}$ concept descriptions C occurring in c to $C^{\mathcal{I}_d} := C^{\mathcal{I}}$,*
- *and the role names r occurring in c to $r^{\mathcal{I}_d} := r^{\mathcal{I}}(d)$.*

The $\mathcal{ALCSCC}^{++}$ concept description C is satisfiable *if there is a finite interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.*

Note that the interpretation of concepts as set variables in $\mathcal{ALCSCC}^{++}$ is global in the sense that it does not depend on d, i.e., $C^{\mathcal{I}_d} = C^{\mathcal{I}_e}$ for all $d, e \in \Delta^{\mathcal{I}}$. In contrast, the interpretation of role names r as set variables is local since only the r-successors of d are considered by $\cdot^{\mathcal{I}_d}$. In $\mathcal{ALCSCC}$, also the interpretation of concepts as set variables is local since in the semantics of $\mathcal{ALCSCC}$ the mapping $\cdot^{\mathcal{I}_d}$ considers only the elements of $C^{\mathcal{I}}$ that are role successors of d for some role name in N_R. This can clearly be simulated in $\mathcal{ALCSCC}^{++}$ by using $C \cap (\bigcup_{r \in N_R} r)$ instead of C when formulating the constraint. Thus, $\mathcal{ALCSCC}$ concepts can be expressed by $\mathcal{ALCSCC}^{++}$ concepts. In addition, extended cardinality constraints (ECBoxes), as introduced in [4], are expressible within $\mathcal{ALCSCC}^{++}$ concept descriptions, as are nominals, the universal role, and role negation.

Proposition 1. *$\mathcal{ALCSCC}^{++}$ concepts can polynomially express nominals, role conjunctions, and $\mathcal{ALCSCC}$ ECBoxes, and thus also ABoxes, $\mathcal{ALC}$ ECBoxes and $\mathcal{ALCSCC}$ TBoxes. In addition, they have the same expressivity as concepts of $\mathcal{ALCSCC}$ extended with the universal role or with role negation, whereas both of these features are not expressible in plain $\mathcal{ALCSCC}$.*

Proof. ECBoxes correspond to Boolean combinations of concept descriptions of the form $sat(c)$ where c contains only concept descriptions as set variables. Since such concept descriptions are satisfied either by no element of $\Delta^{\mathcal{I}}$ or by all of them, their effect is to enforce the constraint on the whole interpretation domain if they are conjoined to a concept description.

Regarding role negation, for given role names $r, \overline{r}$, the constraint $sat(\top \subseteq sat(r \cap \overline{r} \subseteq \emptyset))$ enforces that, for every individual, the sets of its r and $\overline{r}$ successors are disjoint. In addition, the constraint $sat(\top \subseteq sat(|r| + |\overline{r}| = |\mathcal{U}|))$ says that elements of the domain that are not r successors of a given individual must be $\overline{r}$ successors. Thus, we can express that the role $\overline{r}$ is interpreteted as the complement of r, i.e. $\overline{r}^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus r^{\mathcal{I}}$ for every finite interpretation $\mathcal{I}$.

The other statements in the proposition are also easy to see. □

3 Satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions

In the following we consider an $\mathcal{ALCSCC}^{++}$ concept description E and show how to test E for satisfiability 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 satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions. This bound is optimal since consistency of extended cardinality constraints in $\mathcal{ALC}$, as introduced in [4], is already NExpTime hard, and can be expressed as an $\mathcal{ALCSCC}^{++}$ satisfiability problem by Proposition 1.

Our NExpTime algorithm combines ideas from the satisfiability algorithm for $\mathcal{ALCSCC}$ concept descriptions [1] and the consistency procedure for $\mathcal{ALC}$ ECBoxes [4]. In particular, we use the notion of a type, as introduced in [4]. This notion is also similar to the Venn regions employed in [1]. 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 description E as well as the *negations of these subdescriptions.*

Definition 2. *A subset t of $\mathcal{M}$ is a* type *for E if it satisfies the following three 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 $D \in t$; (3) for every concept description $C \sqcup D \in \mathcal{M}$, we have that $C \sqcup D \in t$ iff $C \in t$ or $D \in t$.*

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

Due to Condition (1) in the definition of types, concept descriptions $C_t, C_{t'}$ induced by different types $t \neq t'$ 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, i.e., we have $C^{\mathcal{I}} = \bigcup_{t\, type\, with\, C \in t} C_t^{\mathcal{I}}$ for all $C \in \mathcal{M}$ and finite interpretations $\mathcal{I}$. In particular, the following holds for all finite interpretations $\mathcal{I}$:

$$|C^{\mathcal{I}}| = \sum_{t\, type\, with\, C \in t} |C_t^{\mathcal{I}}| \quad \text{and} \quad |C_t^{\mathcal{I}}| = |\bigcap_{C \in t} C^{\mathcal{I}}|,$$

where the latter identity is an immediate consequence of the definition of C_t as the conjunction of all the elements of t.

Given a type t, the constraints occurring in the top-level Boolean structure of t induce a QFBAPA formula ψ_t, in which the concepts C and roles r occurring in these constraints are replaced by set variables X_C and X_r^t, respectively. Note that set variables corresponding to concepts are independent of the type t, i.e., they are shared by all types, whereas the set variables corresponding to roles are different for different types. This corresponds to the fact that roles are evaluated locally, but concepts are evaluated globally in the semantics of $\mathcal{ALCSCC}^{++}$. In order to ensure that the Boolean structure of concepts is respected by the set variables, we introduce the formula $\beta :=$

$$\bigwedge_{C \sqcap D \in \mathcal{M}} X_{C \sqcap D} = X_C \cap X_D \wedge \bigwedge_{C \sqcup D \in \mathcal{M}} X_{C \sqcup D} = X_C \cup X_D \wedge \bigwedge_{\neg C \in \mathcal{M}} X_{\neg C} = (X_C)^c.$$

Overall, we translate the $\mathcal{ALCSCC}^{++}$ concept E into the QFBAPA formula

$$\delta_E := (|X_E| \geq 1) \wedge \beta \wedge \bigwedge_{t \in \text{types}(E)} (|\bigcap_{C \in t} X_C| = 0) \vee \psi_t.$$

Intuitively, to satisfy E, we need to have at least one element in it, which explains the first conjunct. The third conjunct together with β ensures that, for any type that is realized (i.e., has elements), the constraints of this type are satisfied. The next lemma states that there is a 1–1-relationship between solvability of δ_E and satisfiability of E.

Lemma 1. *The QFBAPA formula δ_E is of size at most exponential in the size of E, and it is satisfiable iff the $\mathcal{ALCSCC}^{++}$ concept description E is satisfiable.*

Since satisfiability of QFBAPA formulae can be decided within NP even for binary coding of numbers [10], this lemma shows that satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions can be decided within NExpTime. Together with the known NExpTime lower bound for consistency of $\mathcal{ALC}$ ECBoxes in [4], this yields:

Theorem 1. *Satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions with numbers encoded in binary is* NExpTime*-complete.*

Thanks to Proposition 1, this carries over to satisfiability of $\mathcal{ALCSCC}^{++}$ knowledge bases which may feature an ABox, a TBox and an ECBox.

4 Query entailment in $\mathcal{ALCSCC}^{++}$

The final result of this section is the undecidability of conjunctive query entailment for $\mathcal{ALCSCC}^{++}$. To this end, we first briefly recap the notion of (Boolean) conjunctive queries and define query entailment.

In queries, we use *variables* from a countably infinite set V. A Boolean *conjunctive query* (CQ) q is a finite set of atoms of the form $r(x, y)$ or $C(z)$, where r is a role, C is concept, and $x, y, z \in V$. A CQ q is *satisfied* by $\mathcal{I}$ (written: $\mathcal{I} \models q$) if there is a *variable assignment* $\pi : V \to \Delta^{\mathcal{I}}$ (called *match*) such that $(\pi(x), \pi(y)) \in r^{\mathcal{I}}$ for every $r(x, y) \in q$ and $\pi(z) \in C^{\mathcal{I}}$ for every $C(z) \in q$. A CQ q is *(finitely) entailed* from a knowledge base $\mathcal{K}$ (written: $\mathcal{K} \models q$) if every (finite) model $\mathcal{I}$ of $\mathcal{K}$ satisfies q.

We actually show undecidability of CQ entailment for a much weaker logic, thereby providing a very restricted fragment of constant-free and equality-free two-variable first-order logic for which finite CQ entailment is already undecidable, significantly strengthening and solidifying earlier results along those lines [14]. Our proof makes use of deterministic Turing machines (DTMs). For our purposes, it is sufficient to consider only computations starting with an empty tape. For space reasons, we assume the reader to be familiar with standard notions and constructions concerning DTMs. We call a DTM *looping* if its run starting contains repeating configurations,i.e., there are two different (and hence – due to determinism – infinitely many) points in time, where the machine's tape content, head position, and state are the same. It is easy to see that the problem of determining if a given TM is looping is undecidable.

We show our undecidability result for the DL $\mathcal{ALC}^{\mathrm{cov}}$, a slight extension of $\mathcal{ALC}$ by *role cover axioms* of the form $\mathrm{cov}(r, s)$ for role names r and s. An interpretation $\mathcal{I}$ satisfies $\mathrm{cov}(r, s)$ if $r^{\mathcal{I}} \cup s^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Role cover axioms can be expressed in $\mathcal{ALCSCC}^{++}$ via $sat\big(\top \subseteq sat(|r \cup s| = |\mathcal{U}|)\big)$, hence $\mathcal{ALC}^{\mathrm{cov}}$ is subsumed by $\mathcal{ALCSCC}^{++}$.

In what follows, assume that a DTM $\mathcal{M}$ is given. We now describe an $\mathcal{ALC}^{\mathrm{cov}}$ TBox $\mathcal{T}$ and conjunctive query q such that $\mathcal{T} \models q$ exactly if $\mathcal{M}$ is not looping. We provide q and $\mathcal{T}$ together with the underlying intuitions of our construction. The goal of our construction is that a countermodel (i.e., an interpretation satisfying $\mathcal{T}$ but not q) corresponds to a looping configuration sequence of $\mathcal{M}$. Thereby, the domain elements represent tape cells at certain computation steps of $\mathcal{M}$. The role h connects consecutive tape cells of the same configuration, whereas the role v connects a configuration's tape cell with the same tape cell of the successor configuration.

We start by providing the query. Intuitively, the query is meant to catch the unwanted situation that two corresponding tape cells of consecutive configurations are v-connected, but the cells to their right aren't.

$$q = \exists x, y, x', y'.v(x, y) \wedge h(x, x') \wedge h(y, y') \wedge \overline{v}(x', y') \tag{1}$$

Table 1. TBox axioms for DTM implementation

$$\top \sqsubseteq \exists aux.(\mathit{TapeStart} \sqcap \mathit{InitConf} \sqcap \mathit{State}_{\mathsf{q}_{\mathrm{ini}}}) \tag{3}$$

$$\top \sqsubseteq \exists h.\top \sqcap \exists v.\top \tag{4}$$

$$\mathit{TapeStart} \sqsubseteq \forall v.\mathit{TapeStart} \tag{5}$$

$$\mathit{InitConf} \sqsubseteq \forall h.\mathit{InitConf} \qquad \mathit{InitConf} \sqsubseteq \mathit{Symbol}_{\square} \tag{6}$$

$$\mathit{State}_{\mathsf{q}} \sqsubseteq \forall h.\mathit{NoHeadR} \qquad \mathit{NoHeadR} \sqsubseteq \forall h.\mathit{NoHeadR} \qquad \mathit{NoHeadR} \sqsubseteq \mathit{NoHead} \tag{7}$$

$$\exists h.\mathit{State}_{\mathsf{q}} \sqsubseteq \mathit{NoHeadL} \qquad \exists h.\mathit{NoHeadL} \sqsubseteq \mathit{NoHeadL} \qquad \mathit{NoHeadL} \sqsubseteq \mathit{NoHead} \tag{8}$$

$$\mathit{State}_{\mathsf{q}} \sqcap \mathit{NoHead} \sqsubseteq \bot \tag{9}$$

$$\mathit{Symbol}_{\sigma} \sqcap \mathit{Symbol}_{\sigma'} \sqsubseteq \bot \qquad \mathit{State}_{\mathsf{q}} \sqcap \mathit{State}_{\mathsf{q}'} \sqsubseteq \bot \tag{10}$$

$$\mathit{NoHead} \sqcap \mathit{Symbol}_{\sigma} \sqsubseteq \forall v.\mathit{Symbol}_{\sigma} \tag{11}$$

$$\mathit{State}_{\mathsf{q}} \sqcap \mathit{Symbol}_{\sigma} \sqsubseteq \forall v.(\mathit{Symbol}_{\sigma'} \sqcap \forall h.\mathit{State}_{\mathsf{q}'}) \tag{12}$$

$$\exists h.(\mathit{State}_{\mathsf{q}} \sqcap \mathit{Symbol}_{\sigma}) \sqsubseteq \forall v.(\mathit{State}_{\mathsf{q}'} \sqcap \forall h.\mathit{Symbol}_{\sigma'}) \tag{13}$$

$$\mathit{TapeStart} \sqcap \mathit{State}_{\mathsf{q}} \sqcap \mathit{Symbol}_{\sigma} \sqsubseteq \forall v.(\mathit{State}_{\mathsf{q}'} \sqcap \mathit{Symbol}_{\sigma'}) \tag{14}$$

We proceed by giving the axioms of $\mathcal{T}$. The following covering axiom ensures that, whenever two elements are not v-connected, they must be $\overline{v}$-connected. This is needed to enable the above query to catch the described problem.

$$\mathrm{cov}(v, \overline{v}) \tag{2}$$

The remaining TBox axioms can be found in Table 1. Axiom 3 ensures (by means of an auxiliary role aux which serves no further purpose) that there is a first tape cell of the first (initial) configuration where the head of the TM is positioned in the initial state. Axiom 4 enforces that for every cell of every configuration there is both a tape cell to its right and a corresponding tape cell in the successor configuration. Axiom 5 makes sure that, for every cell that is the first on its tape, the corresponding successor configuration's tape cell is also the first. Axioms 6 propagates the information that a cell belongs to the initial configuration along the tape, and fills the tape with blanks. Axioms 7–9 (instantiated for every state q) make sure that in every configuration there can only be one cell where the head is positioned. Every cell can only carry one symbol and the head can be in only one state, as ensured by Axioms 10 (for distinct symbols σ, σ' and distinct states q, q'). Thanks to Axiom 11, symbols on head-free cells carry over to the next configuration. As specified by the DTM's transition function, the head reads a symbol σ, writes a symbol σ', changes its state from q to q' and moves right (Axiom 12) or left (Axiom 13) or stays in its place whenever it is supposed to move left but is already at the leftmost tape cell (Axiom 14). This finishes the description of the TBox $\mathcal{T}$, allowing us to establish the claimed property and consequenty the undecidability result.

Proposition 2. *$\mathcal{M}$ is looping iff there is a finite model $\mathcal{I}$ of $\mathcal{T}$ with $\mathcal{I} \not\models Q$.*

Theorem 2. *Finite CQ entailment over $\mathcal{ALC}^{\text{cov}}$ TBoxes is undecidable.*

Proof. According to Proposition 2, the TM looping problem can be reduced to the problem if for a given $\mathcal{ALC}^{\text{cov}}$ TBox $\mathcal{T}$ and conjunctive query q, there is a finite interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{T}$ with $\mathcal{I} \not\models q$. Note that the latter is the case exactly if $\mathcal{T}$ does not finitely entail q.

Finally, taking into account that $\mathcal{ALCSCC}^{++}$subsumes $\mathcal{ALC}^{\text{cov}}$ and only allows for finite models, we obtain the wanted result.

Corollary 1. *Conjunctive query entailment for $\mathcal{ALCSCC}^{++}$is undecidable.*

5 Query entailment from $\mathcal{ALCSCC}$ RCBoxes

As the last result of the paper, we show that decidability of CQ entailment can be regained by moving to a less expressive logic, namely $\mathcal{ALCSCC}$ RCBoxes, i.e., finite sets of restricted cardinality constraints of the form

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

where C_i are $\mathcal{ALCSCC}$ concept descriptions and N_i are integer constants for $1 \leq i \leq k+\ell$, with the obvious semantics. RCBoxes can express general concept inclusions (GCIs) $C \sqsubseteq D$ via $|C \sqcap \neg D| \leq |\bot|$, so we use GCIs when appropriate.

We first sketch our approach. Let $\mathcal{R}$ be an input RCBox and let q be an input CQ. Assume that q is not entailed by $\mathcal{R}$. Hence there is a *counter-model* $\mathcal{I}$, satisfying $\mathcal{R}$ but not q. Our goal is to develop an algorithm to produce another RCBox $\mathcal{R}'$ consisting of $\mathcal{R}$ and some additional knowledge, in a way that $\mathcal{R}'$ is satisfiable if and only if there is a counter-model of $\mathcal{R}$ for q. This can be done by a careful analysis of query matches. Note that if a query q is entailed by $\mathcal{R}$, every model $\mathcal{I}$ falls into one of the following two categories: (i) either there is an acyclic (also called *tree-shaped*) query match or (ii) every query match contains some cyclic substructure of the model.

To deal with these two cases we proceed as follows. For the first case, in order to rule out such models having tree-shaped query matches, we enrich our RCBox $\mathcal{R}$ with additional knowledge forbidding these matches. This can be done by using the so-called rolling-up technique of transforming query matches into concepts, as, e.g., in [7,11,8]. For the second case, we show that it actually cannot happen: we argue that if there is a model $\mathcal{I}$ with only cyclic query matches, one can employ an appropriate model transformation, called pumping, to turn $\mathcal{I}$ into a proper counter-model. This technique is sometimes called big-cycle method [12] or Rosati covers and was successfully used in the context of finite query entailment, e.g, in [6,14,9]. Concluding, it suffices to enrich an $\mathcal{ALCSCC}$ RCBox with additional statements ruling out all models with acyclic query matches, to obtain an RCBox whose satisfiability coincides with the existence of a counter-model. Checking (un)satisfiability of $\mathcal{ALCSCC}$ RCBoxes is decidable [2], so we can conclude following theorem:

Theorem 3. *CQ entailment from $\mathcal{ALCSCC}$ RCBoxes is decidable.*

The rest of this section is devoted to a sketch of the proof of Theorem 3 as outlined above. In Section 5.1 we show how to forbid tree-shaped query matches, in Section 5.2 we handle the case of cyclic matches and construct counter-models.

5.1 Forbidding acyclic query matches

In this section, we provide a method for forbidding tree-shaped query matches. We strongly rely on previous results on query entailment for $\mathcal{ALCHQ}$ knowledge bases [11]. For the sake of simplicity we assume that all queries under consideration are connected (in the graph-theoretic sense).

We first recall some standard definitions. Let q be a conjunctive query and let V_q be the set of variables appearing in q. We can see a query q as a directed graph $G_q = (V_q, E_q)$, where the nodes are variables from q and any two nodes x, y are connected if some atom $r(x, y)$ appears in q. A query is *tree-shaped*, if the underlying graph does not contain any (undirected) cycles.

We introduce the notion of *treeification*, which for a given query q essentially describes the set of all its ways to match in a tree-shaped way.

Definition 3. *Let q be a CQ. We say that a tree-shaped query q' is a* treeification *of q, if q' can be obtained from q by (possibly multiple times) selecting some atoms $r(x, z)$ and $s(y, z)$ and replacing all variables x in the query by y. By $\mathsf{trees}(q)$ we denote the set of all treeifications of q.*

Note that $\mathsf{trees}(q)$ is finite.

Next we describe the so-called rolling-up technique, which transforms a tree-shaped query match into a single concept [16,11]. Let us fix a conjunctive query q and some treeification q' of it. For each variable $x \in V_q$ we construct a concept $C_{q',x}$, with the supposed meaning that $d \in C^{\mathcal{I}}_{q',x}$ if variable x from q' can be mapped to d in a query match represented by q'. We define these concepts as follows: Picking one arbitrary variable $x^r_{q'} \in V_{q'}$, we let $(V_{q'}, \prec)$ be the tree obtained from $G_{q'}$ by orienting all edges away from $x^r_{q'}$. Now, we define $C_{q',x}$ for every $x \in V_{q'}$ in a bottom-up manner as follows: $C_{q',x}$ equals $\bigsqcap_{C(x)\in q'} C$ if x is a leaf (i.e. $\prec$-minimal), otherwise:

$$\bigsqcap_{C(x)\in q'} C \sqcap \bigsqcap_{\substack{(x,y)\in E_{q'}\\ y\prec x}} \Big(\exists \bigcap_{s(x,y)\in q'} s.C_{q',y}\Big) \sqcap \bigsqcap_{\substack{(y,x)\in E_{q'}\\ y\prec x}} \Big(\exists \bigcap_{s(y,x)\in q'} s^-.C_{q',y}\Big)$$

Concepts of the form $\exists\bigcap_{s(y,x)\in q'} s^-.C_{q',y}$ in the above definition are clearly not in $\mathcal{ALC}$ (due to the presence of inverse roles), but this can be remedied as follows: We replace any $\exists r^-.C_{q',y}$ with a (conjunction of) inverted role name(s) r^- by a newly introduced concept $C_{\exists r^-.C_{q',y}}$ for which we also specify $C_{q',y} \sqsubseteq \forall r.C_{r^-.C_{q',y}}$. Like this, any of the concepts $C_{q',y}$ is free of inverses.

For a given CQ q, we enumerate all of its treeifications and roll them up into a concept. Let $\mathcal{R}^{Match}_q$ be an RCBox defining that there exists a tree-shaped query match in a model:

$$\bigsqcup_{q' \in \mathsf{trees}(q)} C_{q',x^r_{q'}} \sqsubseteq Match_q \tag{15}$$

The two coming lemmas give the precise meaning to the defined concepts.

Lemma 2. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ RCBox and let q be a conjunctive query. Let $\mathcal{R}_q^{Match}$ be as defined above. Assume that $\mathcal{R} \cup \mathcal{R}_q^{Match}$ has a model $\mathcal{I}$ such that $Match_q^{\mathcal{I}}$ is empty. Then $\mathcal{I}$ does not have any tree-shaped query matches.*

Lemma 3. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ RCBox and let q be a conjunctive query and $\mathcal{R}_q^{Match}$ as defined above. If there is a model $\mathcal{I}$ of $\mathcal{R}$ without any tree-shaped query matches, then $\mathcal{R}^* = \mathcal{R} \cup \mathcal{R}_q^{Match} \cup \{\top \sqsubseteq \neg Match_q\}$ is satisfiable.*

Satisfiability checking of $\mathcal{R}^*$ can be performed with an algorithm from [2]. Together with Lemma 2 and Lemma 3, we establish:

Theorem 4. *It is decidable whether for a given CQ q and a given $\mathcal{ALCSCC}$ RCBox $\mathcal{R}$, there is a model of $\mathcal{R}$ without any tree-shaped query matches of q.*

5.2 Pumping models to enforce high girth

Now we take a closer look at the previously announced pumping method for eliminating non-tree-shaped query matches.

Definition 4. *Let $\mathcal{I}$ be an interpretation, $\Delta^{\mathcal{I}} = \{d_1, d_2, \ldots, d_n\}$ be the set of domain elements and E the set of "edges", i.e., $e = (d, d')$ occur in E if there is a role r, s.t. $(d, d') \in r^{\mathcal{I}}$. Additionally, let F be the set of all functions f of type $f : E \to \{0, 1\}$. We define a pumping of $\mathcal{I}$, denoted with $\mathsf{pump}(\mathcal{I}) = \mathcal{I}'$, in the following way: (i) we set $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}} \times F$, (ii) for a concept name A, an element $d \in \Delta^{\mathcal{I}}$ and any function f we set $(d, f) \in A^{\mathcal{I}'}$ iff $d \in A^{\mathcal{I}}$ (iii) for any role name r and a pair $((d, f), (d', f'))$ we set $((d, f), (d', f')) \in r^{\mathcal{I}'}$ iff $(d, d') \in r^{\mathcal{I}}$ and functions f, f' are equal on all arguments except for the argument $e = (d, d')$ for which $f'(e) = 1 - f(e)$.*

The *girth* of $\mathcal{I}$ is the length of a shortest (undirected) proper cycle contained in $\Delta^{\mathcal{I}}$ (proper means that no element from E is used more than once). It is not difficult to see that the girth of $\mathsf{pump}(\mathcal{I})$ is at least twice the girth of $\mathcal{I}$: due to the fact that we need to flip the e value in f every time we cross an edge e.

Lemma 4. *Let $\mathcal{I}$ be an interpretation with girth k. Then the girth of $\mathsf{pump}(\mathcal{I})$ is at least $2k$.*

Correctness of the pumping method is guaranteed by the following lemma. Its proof relies on two observations (i) degree of each node and "types" of its successors are preserved during pumping, and (ii) all global constraints from the RCBox are still satisfied since all cardinalities from inequalities were multiplied by the same number (i.e. $|F|$). Formally, the proof goes via an induction over the depth of $\mathcal{ALCSCC}$ concepts.

Lemma 5. *Let $\mathcal{R}$ be an RCBox with a model $\mathcal{I}$. Then* $\mathsf{pump}(\mathcal{I})$ *is a model of $\mathcal{R}$.*

Now we explain how to use the pumping method from the Definition 4 to erase non-tree query matches and obtain a counter-model.

Lemma 6. *Let q be a CQ and let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ RCBox. Assume $\mathcal{R}$ has a model $\mathcal{I}$ that does not have any acyclic query matches of q. Then there exists a counter-model $\mathcal{I}'$ for q and $\mathcal{R}$.*

Proof. We define $\mathcal{I}'$ as the $|q|$-fold application of pump to $\mathcal{I}$. Since each iteration of pumping doubles the girth of the input structure, the girth of $\mathcal{I}'$ is strictly greater than $|q|$. Moreover, during the pumping process, we haven't introduced any new query matches. Hence, since any cyclic match of q would require girth $\leq |q|$, the query q cannot match into $\mathcal{I}'$ anymore and $\mathcal{I}'$ is a counter-model.

By combining (i) a method of pumping a structure from this Section, (ii) a method of enriching a knowledge-base with a knowledge to forbid all acyclic query matches from the previous section and (iii) an algorithm for testing (un)satisfiability for $\mathcal{ALCSCC}$ concepts w.r.t RCBoxes, we conclude Theorem 3.

A rough complexity analysis give us a 2ExpTime upper bound. We believe that this bound can be improved to ExpTime by adapting techniques from [11].

6 Conclusion

In our work, we have significantly pushed the boundaries of quantitative extensions to description logics. We showed that $\mathcal{ALC}$ can not only be extended by both global (extension-based) and local (neighbourhood-based) arithmetic constraints but even by hybrid constraints mixing the two, yielding a significant increase in expressivity without negative impact on the complexity of satisfiability testing. On the downside, we had to find out that this extension leads to undecidability of conjunctive query answering. However, we were able to regain decidability and even a favourable complexity under appropriate restrictions.

Without any doubt, the ability to deal with factual data, i.e. ABoxes, is of utmost importance in the context of statistical considerations and querying. Therefore, our next step will be to extend our investigations to full-fledged $\mathcal{ALCSCC}$ knowledge bases including ABoxes. In fact, we have already established that ABoxes can be added to $\mathcal{ALCSCC}$ RCBoxes without impacting the ExpTime complexity of satisfiability checking and are confident that standard query partitioning and a slight modification of model pumping will allow us to show ExpTime completeness of CQ answering.

Acknowledgements

Bartosz Bednarczyk was supported by the Polish Ministry of Science and Higher Education program “Diamentowy Grant” no. DI2017 006447. Sebastian Rudolph was supported by the European Research Council (ERC) through the Consolidator Grant 771779 (DeciGUT). Franz Baader was partially supported by the German Research Foundation (DFG) within the Research Unit 1513 Hybris.

References

1. Franz Baader. A new description logic with set constraints and cardinality constraints on role successors. In Clare Dixon and Marcelo Finger, editors, *Proceedings of the 11th International 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. Expressive cardinality constraints on $\mathcal{ALCSCC}$ concepts. In *Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC'19)*. ACM, 2019.
3. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, second edition, 2007.
4. Franz Baader and Andreas Ecke. Extending the description logic alc with more expressive cardinality constraints on concepts. In *GCAI 2017. 3rd Global Conference on Artificial Intelligence*, volume 50 of *EPiC Series in Computing*, pages 6–19. EasyChair, 2017.
5. Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
6. Vince Bárány, Georg Gottlob, and Martin Otto. Querying the guarded fragment. *Logical Methods in Computer Science*, 10(2), 2014.
7. Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.
8. Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic SHIQ. *J. of Artificial Intelligence Research*, 31:157–204, 2008.
9. Yazmin Angélica Ibáñez-García, Carsten Lutz, and Thomas Schneider. Finite model reasoning in horn description logics. In *Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 288–297, 2014.
10. 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.
11. Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2008)*, Lecture Notes in Artificial Intelligence, pages 179–193. Springer-Verlag, 2008.
12. Martin Otto. Highly acyclic groups, hypergraph covers, and the guarded fragment. *J. ACM*, 59(1):5:1–5:40, 2012.
13. Rafael Peñaloza and Nico Potyka. Towards statistical reasoning in description logics over finite domains. In Serafin Moral and Olivier Pivert, editors, *Proc. of the 11th Int. Conf. on Scalable Uncertainty Management (SUM 2017)*, volume 10564 of *Lecture Notes in Computer Science*. Springer-Verlag, 2017.
14. Ian Pratt-Hartmann. Data-complexity of the two-variable fragment with counting quantifiers. *Inf. Comput.*, 207(8):867–888, 2009.
15. Sebastian Rudolph. Foundations of description logics. In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski,

and Peter F. Patel-Schneider, editors, *Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011*, volume 6848 of *LNCS*, pages 76–136. Springer, 2011.

16. Sergio Tessaris. Querying expressive dls. In *Proc. of the 2001 Description Logic Workshop (DL 2001)*, volume 49 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2001.
17. 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.
18. Stephan Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artificial Intelligence Research*, 12:199–217, 2000.
19. Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.