Regular Path Queries in Expressive Description Logics with Nominals[*]

Diego Calvanese
KRDB Research Centre
Free University of Bozen-Bolzano
Piazza Domenicani 3, Bolzano, Italy

calvanese@inf.unibz.it

Thomas Eiter and **Magdalena Ortiz**
Institute of Information Systems
Vienna University of Technology
Favoritenstraße 9-11, Vienna, Austria

(eiter|ortiz)@kr.tuwien.ac.at

Abstract

Reasoning over complex queries in the DLs underlying OWL 2 is of importance in several application domains. We provide decidability and (tight) upper bounds for the problem of checking entailment and containment of positive regular path queries under various combinations of constructs used in such expressive DLs; specifically: regular expressions and (safe) Booleans over roles, and allowing for the combination of any two constructs among inverse roles, qualified number restrictions, and nominals. Our results carry over also to the DLs of the $\mathcal{SR}$ family, and thus have a direct impact on OWL 2.

1 Introduction

OWL 2, the upcoming W3C Web Ontology Language [Cuenca Grau *et al.*, 2008], is based on the expressive *Description Logic* (DL) $\mathcal{SROIQ}$ [Horrocks *et al.*, 2006] and features several constructs considered important in ontology-based applications. A crucial challenge is to access OWL 2 ontologies via expressive, database inspired, query languages, such as *(unions of) conjunctive queries, (U)CQs*, or variants of *regular path queries, RPQs* (allowing for binary query atoms that are regular expressions), used in semi-structured data.

Reasoning over complex queries had yet to be addressed for expressive DLs that support expressive concept and role constructs plus different combinations of qualified number restrictions ($\mathcal{Q}$), inverse roles ($\mathcal{I}$), and nominals ($\mathcal{O}$). In particular, this applies to $\mathcal{SROQ}$, $\mathcal{SROI}$, and $\mathcal{SRIQ}$, three mutually incomparable sublogics of $\mathcal{SROIQ}$. Indeed, the only algorithms for query entailment in expressive DLs with $\mathcal{O}$ are for UCQs in $\mathcal{SHOQ}$ [Glimm *et al.*, 2008], and for UCQs without transitive roles in $\mathcal{SHOI}$ [Ortiz *et al.*, 2008]; none of them supports complex role assertions as in the $\mathcal{SR}$ family. Calvanese *et al.* [2007] consider the DL $\mathcal{ALCQIb}_{reg}$ which lacks nominals but supports $\mathcal{I}$, $\mathcal{Q}$, and regular expressions and (safe) Booleans over roles, capturing a large fragment of $\mathcal{SRIQ}$. Their algorithm answers *positive 2-way reg-*

ular path queries (P2RPQs), which capture all the aforementioned query languages and are, to our knowledge, the most expressive query language considered so far.

In this paper, we address KB satisfiability, as well as entailment and containment of P2RPQs in the sublogics of $\mathcal{SROIQ}$ that allow for any two among $\mathcal{Q}$, $\mathcal{I}$, and $\mathcal{O}$. Specifically, we show that KB satisfiability in $\mathcal{SRIQ}$, $\mathcal{SROI}$, and $\mathcal{SROQ}$ can be solved in 2EXPTIME. For $\mathcal{SROQ}$ and $\mathcal{SROI}$, these are, to our knowledge, the first such bounds, and they all hold even when the numbers in the number restrictions are coded in binary. We also show that P2RPQ entailment is decidable, and so is containment $q_1 \subseteq q_2$ if the DL has $\mathcal{O}$ or q_1 has no regular expressions. These are the first decidability results for reasoning on queries in the DLs of the $\mathcal{SR}$ family, and hence in significant sublogics of OWL 2.

Our results are based on automata theoretic techniques, and are achieved indirectly by reducing (with an unavoidable exponential blowup) $\mathcal{SROIQ}$ to the novel DL $\mathcal{ZOIQ}$, and tackling KB satisfiability and P2RPQ entailment and containment in its sublogics that allow for any two among $\mathcal{Q}$, $\mathcal{I}$, and $\mathcal{O}$. $\mathcal{ZOIQ}$ is the DL that extends $\mathcal{ALCQIb}_{reg}$ with $\mathcal{O}$ and concepts $\exists S.\mathsf{Self}$. Specifically, we exploit the recently introduced *fully enriched automata (FEA)* [Bonatti *et al.*, 2008] and reduce KB satisfiability in $\mathcal{ZOIQ}$ to their emptiness. Our construction yields a decision procedure for KB satisfiability in all sublogics of $\mathcal{ZOIQ}$ that enjoy the *quasi-forest model property* (see Section 3.1), and in particular for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$. This is, to our knowledge, the first automata procedure that simultaneously handles $\mathcal{Q}$, $\mathcal{I}$, and $\mathcal{O}$; it additionally considers Boolean role expressions and Self concepts. Relying on the results in [Bonatti *et al.*, 2008], we obtain a tight EXPTIME upper bound for these logics, even with binary coding of numbers.

We then build on the techniques in [Calvanese *et al.*, 2007] and reduce entailment of P2RPQs in the sublogics of $\mathcal{ZOIQ}$ to automata emptiness. This requires us to show how FEAs can be reduced to a simpler automata model. Further, we show that in DLs with $\mathcal{O}$ (and regular role expressions), containment of P2RPQs can be reduced to entailment, and hence obtain the first decidability and complexity results for containment of recursive queries in DLs. Specifically, we obtain (with unary coding of numbers), an optimal 2EXPTIME upper bound for P2RPQ entailment in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$, and for P2RPQ containment in $\mathcal{ZOQ}$ and $\mathcal{ZOI}$. For $\mathcal{ZIQ}$,

the same bound holds for containment of CQs in P2RPQs.

2 Preliminaries

Description Logics (DLs). The DL $\mathcal{ALCOIQb}^{\mathsf{Self}}_{reg}$ extends the basic DL $\mathcal{ALC}$ with nominals ($\mathcal{O}$), inverse roles ($\mathcal{I}$), qualified number restrictions ($\mathcal{Q}$), regular expressions over roles (reg), safe Boolean role expressions (b) and inclusion axioms, and concepts of the form $\exists S.\mathsf{Self}$ (Self) as in $\mathcal{SROIQ}$ [Horrocks *et al.*, 2006]. In the following, we use $\mathcal{Z}$ as an abbreviation for $\mathcal{ALCb}^{\mathsf{Self}}_{reg}$; thus we will deal with the logic $\mathcal{ZOIQ}$ and its sublogics $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$

Syntax. Let **C**, **R**, and **I** be fixed, countably infinite sets of *concept*, *role*, and *individual names*, respectively. We assume that **C** contains $\top$ and $\bot$, denoting respectively the universal and the empty concept, and that **R** contains T and B, denoting respectively the universal and the empty role. *Atomic concepts B, concepts C, atomic roles R, simple roles S, and roles T*, obey the following EBNF grammar, where $a \in \mathbf{I}$, $A \in \mathbf{C}$, $P \in \mathbf{R}$, and $P \neq \mathsf{T}$:

$$B ::= A \mid \{a\}$$
$$C ::= B \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall T.C \mid \exists T.C \mid$$
$$\geqslant n\,S.C \mid \leqslant n\,S.C \mid \exists S.\mathsf{Self}$$
$$R ::= P \mid P^{-}$$
$$S ::= R \mid S \cap S \mid S \cup S \mid S \setminus S$$
$$T ::= \mathsf{T} \mid S \mid T \cup T \mid T \circ T \mid T^{*} \mid id(C)$$

An *expression* is a concept or a role. *Subconcepts, subroles,* and *subexpressions* are defined in the natural way.

An *assertion* has the form $C(a)$, $S(a, b)$, or $a \neq b$, where C and S are as above and $a, b \in \mathbf{I}$. A *concept inclusion axiom (CIA)* has the form $C \sqsubseteq C'$, where C and C' are concepts, and a *Boolean role inclusion axiom (BRIA)* has the form $S \sqsubseteq S'$, where S and S' are simple roles. A $\mathcal{ZOIQ}$ *knowledge base* (KB) is a pair $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{A}$ is a finite set of assertions (called *ABox*), and $\mathcal{T}$ is a finite set of CIAs and BRIAs (called *TBox*). W.l.o.g. we assume that $\mathcal{A}$ is non-empty.

We also consider three sublogics of $\mathcal{ZOIQ}$ that result by disallowing different constructors in concepts and roles:
– $\mathcal{ZIQ}$ disallows $\{a\}$ (*nominal concepts*);
– $\mathcal{ZOQ}$ disallows P^{-} (*inverse roles*);
– $\mathcal{ZOI}$ disallows $\geqslant n\,S.C$, $\leqslant n\,S.C$ (*number restrictions*).

Semantics. We rely on the usual notion of *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, consisting of a *domain* $\Delta^{\mathcal{I}} \neq \emptyset$ and a *valuation function* $\cdot^{\mathcal{I}}$ [Baader *et al.*, 2003]. The semantics of concept and role constructs (including the constructs for regular expressions over roles) is the standard one. We just note that $(\exists S.\mathsf{Self})^{\mathcal{I}} = \{x \mid (x, x) \in S^{\mathcal{I}}\}$ and $(id(C))^{\mathcal{I}} = \{(x, x) \mid x \in C^{\mathcal{I}}\}$.

$\mathcal{I}$ is a *model* of a concept C, denoted $\mathcal{I} \models C$, if $C^{\mathcal{I}} \neq \emptyset$. Satisfaction of an assertion (resp., CIA, BRIA) γ by $\mathcal{I}$, denoted $\mathcal{I} \models \gamma$, is defined as usual. $\mathcal{I}$ is a *model* of an ABox (resp., TBox) Γ, denoted $\mathcal{I} \models \Gamma$, if $\mathcal{I} \models \gamma$ for each γ in Γ. $\mathcal{I}$ is a *model* of a KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ if $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \models \mathcal{T}$. The *knowledge base satisfiability problem* consists on deciding whether a given KB $\mathcal{K}$ has a model.

Query Entailment and Containment. A *positive 2-way regular path query* (P2RPQ) is a formula $q = \exists \vec{x}.\varphi(\vec{x})$, where $\varphi(\vec{x})$ is built using $\wedge$ and $\vee$ from atoms $C(z)$ and $T(z, z')$, where z, z' are variables from $\vec{x}$ or individuals, C is a concept and T a role.[1] If all atomic concepts and roles in φ occur in a KB $\mathcal{K}$, then q is a *query over $\mathcal{K}$*. Note that P2RPQs generalize conjunctive RPQs [Calvanese *et al.*, 2000], in which the formula $\varphi(\vec{x})$ is built using only conjunction, and ordinary *conjunctive queries* (CQs), where in addition only atomic concepts and roles may be used in atoms.[2]

Given an interpretation $\mathcal{I}$, a *match π for $\mathcal{I}$ and q* is an assignment of an element $\pi(x) \in \Delta^{\mathcal{I}}$ to each variable x in $\vec{x}$ that makes φ true in the usual sense. $\mathcal{I}$ *satisfies* q, denoted $\mathcal{I} \models q$, if there is a match π for $\mathcal{I}$ and q.

– *Query entailment* is the problem of deciding, given a KB $\mathcal{K}$ and a query q over $\mathcal{K}$, whether $\mathcal{I} \models q$ for each model $\mathcal{I}$ of $\mathcal{K}$, denoted $\mathcal{K} \models q$.

– *Query containment* is the problem of deciding, given a KB $\mathcal{K}$ and two queries q_1 and q_2 over $\mathcal{K}$, whether $\mathcal{I} \models q_1$ implies $\mathcal{I} \models q_2$ for each model $\mathcal{I}$ of $\mathcal{K}$, denoted $\mathcal{K} \models q_1 \subseteq q_2$.

We observe that KB (un)satisfiability trivially reduces to query entailment: $\mathcal{K}$ is unsatisfiable iff $\mathcal{K} \models \exists x.\bot(x)$.

3 Reasoning with automata in the $\mathcal{Z}$ family

In this section, we reduce KB satisfiability to emptiness of automata that run over infinite labeled forests.

Reducing KB to concept satisfiability. We rewrite a $\mathcal{ZOIQ}$ KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ into a *normal* concept $C_{\mathcal{K}}$, which is a concept in negation normal form (NNF) not containing $\top$, $\bot$, T, or B.

To do so, we first eliminate all BRIAs, replacing each $S \sqsubseteq S'$ in $\mathcal{T}$ by a CIA $\exists (S \setminus S').\top \sqsubseteq \bot$ (cf. [Rudolph *et al.*, 2008]). The special symbols $\top$, $\bot$ and B are simulated via fresh concept names $A_{\top}$, $A_{\bot}$ and a fresh role name P_B, by adding $C \sqcup \neg C \sqsubseteq A_{\top}$ and $A_{\top} \sqsubseteq \neg A_{\bot}$ for some concept C, and adding $A_{\top} \sqsubseteq \forall P_B.A_{\bot}$.

To eliminate T, we add an assertion $P_U(a, b)$ for each pair of individuals a, b occurring in $\mathcal{A}$, where P_U is a fresh role name. Then we replace in $\mathcal{K}$ each occurrence of T by the role R_{T}, where $R_{\mathsf{T}} = (P_U \cup \{P \mid P \in \mathbf{R} \text{ occurs in } \mathcal{K}\})^*$ if no inverse roles P^- occur in $\mathcal{K}$, and $R_{\mathsf{T}} = (P_U \cup \{P, P^- \mid P \in \mathbf{R} \text{ occurs in } \mathcal{K}\})^*$ otherwise.

As usual, using nominals, the ABox is *internalized* into the TBox: assertions $C(a)$, $S(a, b)$, and $a \neq b$ become CIAs $\{a\} \sqsubseteq C$, $\{a\} \sqsubseteq \exists S.\{b\}$, and $\{a\} \sqsubseteq \neg\{b\}$, respectively. Finally, using the role R_{T}, the BRIA-free TBox $\mathcal{T}$ is *internalized* into a concept $C_{\mathcal{T}} = \forall R_{\mathsf{T}}.\bigsqcup_{C_1 \sqsubseteq C_2 \in \mathcal{T}}(\neg C_1 \sqcup C_2)$. We thus obtain:

Proposition 3.1. *Given a $\mathcal{ZOIQ}$ KB $\mathcal{K}$, one can construct in linear time a normal concept $C_{\mathcal{K}}$ such that: (i) if $\mathcal{K}$ is in $\mathcal{L}$, then $C_{\mathcal{K}}$ is in $\mathcal{L}$, for $\mathcal{L}$ any of $\mathcal{ZOIQ}$, $\mathcal{ZOI}$, or $\mathcal{ZOQ}$; (ii) if $\mathcal{K}$ is in $\mathcal{ZIQ}$, then $C_{\mathcal{K}}$ is in $\mathcal{ZOIQ}$ and all nominals in it stem from ABox internalization; (iii) for every P2RPQ q, $\mathcal{K} \not\models q$ iff there is some $\mathcal{I}$ such that $\mathcal{I} \models C_{\mathcal{K}}$ and $\mathcal{I} \not\models q$.*

---

[1] Such queries are called *Boolean*; it is well known that queries with answer variables are reducible to Boolean ones.

[2] The restriction in CQs to atomic concept atoms is w.l.o.g., since complex concepts can be defined in the TBox. Instead, roles containing regular expressions cannot be defined in the TBox, and hence conjunctive RPQs and P2RPQs properly extend CQs.

3.1 Quasi-Forest Model Properties

We show now that $\mathcal{ZIQ}$, $\mathcal{ZOI}$, and $\mathcal{ZOQ}$ enjoy the *quasi-forest model property*. This allows us to decide concept satisfiability (and query entailment) by deciding the existence of quasi-forest models (where the query has no match).

A *forest* is a set $F \subseteq \mathbb{N}^*$ such that $x \cdot c \in F$ and $x \in \mathbb{N}^+$ imply $x \in F$; its elements are called *nodes*. For each $x \in F$, $\mathsf{succ}(x) = \{x \cdot c \in F \mid c \in \mathbb{N}\}$ is the set of *successors* of x; x is their *predecessor*. F has *branching degree* k, if $|\mathsf{succ}(x)| \leq k$ for each $x \in F$. By $\mathsf{roots}(F)$ we denote the *roots* of F, i.e., the nodes with no predecessor. F is a *tree* if $|\mathsf{roots}(F)| = 1$. The *tree of F rooted at c* is $T_c = \{c \cdot x \mid x \in \mathbb{N}^*\} \cap F$.

An *(infinite) path* in F is an (infinite) tree $P \subseteq F$ with branching degree 1. By convention, $x \cdot \varepsilon = x$ and $(x \cdot i) \cdot {-}1 = x$.

A *Σ-labeled forest (resp. tree)* is a pair $\langle F, V \rangle$, where F is a forest (resp. tree) and $V : F \to \Sigma$ is a labeling function.

Definition 3.2 (Quasi-forest models). Let C be a $\mathcal{ZOIQ}$ concept. An interpretation $\mathcal{I}$ is a *quasi-forest model of C* if:
- $\Delta^{\mathcal{I}}$ is a forest,
- $a^{\mathcal{I}} \in \mathsf{roots}(\Delta^{\mathcal{I}})$ for each individual a occurring in C,
- $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for some individual a occurring in C, and
- for every $x, y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in R^{\mathcal{I}}$ for some role R, either (i) $\{x, y\} \cap \mathsf{roots}(\Delta^{\mathcal{I}}) \neq \emptyset$, (ii) $x = y$, (iii) $y \in \mathsf{succ}(x)$, or (iv) $x \in \mathsf{succ}(y)$.

Note that $(x, y) \in R^{\mathcal{I}}$ may hold if (*i*) x or y is a root of $\Delta^{\mathcal{I}}$ (due to nominals), (*ii*) y is x itself (due to $\exists S.\mathsf{Self}$ concepts), or, as usual in logics with inverses, (*iii*) y is in $\mathsf{succ}(x)$ or (*iv*) y is the predecessor of x.

The above definition generalizes those of related logics (e.g., in [Bonatti *et al.*, 2008; Calvanese *et al.*, 2007; Sattler & Vardi, 2001]), and accommodates all constructs of $\mathcal{ZOIQ}$.

The following proposition states that to decide query entailment, we only need to consider quasi-forest models.

Proposition 3.3. *Let C be a normal $\mathcal{ZOIQ}$ concept such that (a) C is a $\mathcal{ZOQ}$ or $\mathcal{ZOI}$ concept, or (b) C is obtained from a $\mathcal{ZIQ}$ KB as in Proposition 3.1. Then, for every P2RPQ q, if C has a model $\mathcal{I}$ with $\mathcal{I} \not\models q$, then it has a quasi forest model $\mathcal{I}'$ with $\mathcal{I}' \not\models q$.*

Proof (sketch). Every model $\mathcal{I}$ of C can be used to obtain a quasi forest model $\mathcal{I}'$ of C, such that $\mathcal{I}'$ is a counterexample to query entailment whenever $\mathcal{I}$ is. If C is a $\mathcal{ZOI}$ or $\mathcal{ZOQ}$ concept, we can proceed along the lines of the proofs in [Bonatti *et al.*, 2008; Sattler and Vardi, 2001], respecting Boolean role constructs and Self. When C is a $\mathcal{ZOIQ}$ concept obtained from a rewriting a $\mathcal{ZIQ}$ KB, the impact of its nominals (which all stem from ABox internalization) can be confined to the roots of the quasi-forest model. In this case, we can proceed as in [Bonatti *et al.*, 2008], again respecting Boolean role constructs and Self. $\square$

Note that Proposition 3.3 does not hold for arbitrary $\mathcal{ZOIQ}$ concepts, even in the absence of Boolean roles, regular expressions, and concepts of the form $\exists S.\mathsf{Self}$ [Tobies, 2000].[3]

---

[3] Prop. 3.3 fails also in the presence of arbitrary role negation, as opposed to role difference, even if regular role expressions and all of $\mathcal{O}$, $\mathcal{I}$, and $\mathcal{Q}$ are disallowed.

3.2 Enriched Automata

For deciding KB satisfiability in the sublogics of $\mathcal{ZOIQ}$, we build on the techniques of [Bonatti *et al.*, 2008] that use *fully enriched automata (FEAs)*. FEAs extend two-way alternating parity tree automata by adding graded and root transitions.

For a set W, let $\mathcal{B}(W)$ be the set of Boolean formulas constructible with atoms $W \cup \{\mathbf{t}, \mathbf{f}\}$ and $\wedge$, $\vee$. We say that $V \subseteq W$ *satisfies* $\varphi \in \mathcal{B}(W)$, if assigning $\mathbf{t}$ to all $v \in V$ and $\mathbf{f}$ to all $w \in W \setminus V$ makes φ true. For $b > 0$, let $D_b = \{-1, \varepsilon\} \cup \{\langle 0 \rangle, \ldots, \langle b \rangle\} \cup \{[0], \ldots, [b]\} \cup \{\langle \mathsf{root} \rangle, [\mathsf{root}]\}$.

Definition 3.4. A *fully enriched automaton (FEA) with index n* is a tuple $\mathbf{A} = \langle \Sigma, b, Q, \delta, q_0, F \rangle$, where Σ is a finite input alphabet, $b > 0$ is a counting bound, Q is a finite set of states, $\delta : Q \times \Sigma \to \mathcal{B}(D_b \times Q)$ is a transition function, $q_0 \in Q$ is an initial state, and $F = (G_1, \ldots, G_n)$, with $G_1 \subseteq G_2 \subseteq \cdots \subseteq G_n = Q$, is a *parity acceptance condition*.

Intuitively, a *graded transition* $(\langle i \rangle, q)$ (resp., $([i], q)$), sends off a copy of $\mathbf{A}$ in state q to $i{+}1$ (resp., to all but i) successor nodes, and a *root transition* $\langle \mathsf{root} \rangle$ (resp., $[\mathsf{root}]$), sends off a copy of $\mathbf{A}$ in state q to one (resp., to all) roots.

The acceptance of a forest F by $\mathbf{A}$ can be formalized through the notion of run, which is a tree labeled by elements of $F \times Q$. Intuitively, in a run, a node y labeled by (x, q) describes a copy of $\mathbf{A}$ that is in state q and reads node x of F. The conditions on a run ensure that the labels of adjacent nodes satisfy the transition function of $\mathbf{A}$.

Definition 3.5. A *run* of $\mathbf{A}$ over a labeled forest $\langle F, V \rangle$ is a $F \times Q$-labeled tree $\langle T_r, r \rangle$ such that

(i) $r(\mathsf{root}(T_r)) = (c, q_0)$ for some $c \in \mathsf{roots}(F)$, and

(ii) for every $y \in T_r$ with $r(y) = (x, q)$, some $W \subseteq D_b \times Q$ satisfying $\delta(q, V(x))$ exists such that, for all $(d, s) \in W$:
- if $d \in \{-1, \varepsilon\}$, then $x \cdot d$ is defined and there is some $j \in \mathbb{N}$ such that $x \cdot j \in T_r$ and $r(x \cdot j) = (x \cdot d, s)$;
- if $d = \langle n \rangle$, then there is some $M \subseteq \mathsf{succ}(x)$ with $|M| > n$ such that, for each $z \in M$, there is some $j \in \mathbb{N}$ such that $x \cdot j \in T_r$ and $r(x \cdot j) = (z, s)$;
- if $d = [n]$, then there is some $M \subseteq \mathsf{succ}(x)$ with $|M| \leq n$ such that, for each $z \in \mathsf{succ}(x) \setminus M$, there is some $j \in \mathbb{N}$ such that $x \cdot j \in T_r$ and $r(x \cdot j) = (z, s)$;
- if $d = \langle \mathsf{root} \rangle$, then there is some $c \in \mathsf{roots}(F)$ and $j \in \mathbb{N}$ such that $x \cdot j \in T_r$ and $r(x \cdot j) = (c, s)$;
- if $d = [\mathsf{root}]$, then for each $c \in \mathsf{roots}(F)$ there is some $j \in \mathbb{N}$ such that $x \cdot j \in T_r$ and $r(x \cdot j) = (c, s)$.

The run $\langle T_r, r \rangle$ is *accepting* if, for each infinite path P of T_r, there is an *even* i such that $\mathsf{Inf}(\langle P, r \rangle) \cap G_i \neq \emptyset$ and $\mathsf{Inf}(\langle P, r \rangle) \cap G_{i-1} = \emptyset$, where $\mathsf{Inf}(\langle P, r \rangle)$ is the set of all states $q \in Q$ such that $\{y \in P \mid \exists x. r(y) = (x, q)\}$ is infinite.

A FEA $\mathbf{A}$ *accepts* a labeled forest $\langle F, V \rangle$ if it has some accepting run over $\langle F, V \rangle$. The set of all forests accepted by $\mathbf{A}$ is $\mathscr{L}(\mathbf{A})$. The *non-emptiness problem* is the problem of deciding whether $\mathscr{L}(\mathbf{A}) \neq \emptyset$ for a given FEA $\mathbf{A}$.

Theorem 3.6 ([Bonatti *et al.*, 2008]). *The non-emptiness problem for a FEA $\mathbf{A} = \langle \Sigma, b, Q, \delta, q_0, F \rangle$ with index k can be solved in time $(b + 2)^{\mathcal{O}(|Q|^3 \cdot k^2 \cdot \log k \cdot \log b^2)}$.*

3.3 Reducing Satisfiability to Automata Emptiness

In the rest of this section, C denotes a normal $\mathcal{ZOIQ}$ concept. To represent a quasi-forest model $\mathcal{I}$ of C as a labeled forest, we label each individual with the set of atomic concepts (i.e., concept names and nominals) it satisfies. For atomic roles, we add R to the label of x whenever $(x, x') \in R^{\mathcal{I}}$ and x' is not a root. Arcs leading to roots are handled as in [Bonatti *et al.*, 2008], by adding a special symbol $\uparrow_a^R$ to the label of x whenever $(x, a^{\mathcal{I}}) \in R^{\mathcal{I}}$. Finally, we represent loops $(x, x) \in R^{\mathcal{I}}$ using special labels R_{Self}.

Definition 3.7. We denote by $\mathbf{R}_C$ and $\mathbf{I}_C$ respectively the sets of role and individual names occurring in C, by $\mathbf{CI}_C$ the set of atomic concepts occurring in C, and we define $\overline{\mathbf{R}}_C = \mathbf{R}_C \cup \{P^- \mid P \in \mathbf{R}_C\}$. We also define:

$$\Theta(C) = \mathbf{CI}_C \cup \overline{\mathbf{R}}_C \cup \{R_{\mathsf{Self}} \mid R \in \overline{\mathbf{R}}_C\} \cup \{\uparrow_a^R \mid R \in \overline{\mathbf{R}}_C \text{ and } a \in \mathbf{I}_C\}, \text{ and}$$
$$\Sigma_C = 2^{\Theta(C)}.$$

The *forest encoding* of a quasi-forest model $\mathcal{I}$ of C is the Σ_C-labeled forest $\langle \Delta^{\mathcal{I}}, L^{\mathcal{I}} \rangle$ such that for each $x \in \Delta^{\mathcal{I}}$:

$$L^{\mathcal{I}}(x) = \{B \in \mathbf{CI}_C \mid x \in B^{\mathcal{I}}\} \cup$$
$$\{R_{\mathsf{Self}} \mid R \in \overline{\mathbf{R}}_C \text{ and } (x, x) \in R^{\mathcal{I}}\} \cup$$
$$\{R \in \overline{\mathbf{R}}_C \mid (x', x) \in R^{\mathcal{I}} \text{ and } x \in \mathsf{succ}(x')\} \cup$$
$$\{\uparrow_a^R \mid R \in \overline{\mathbf{R}}_C, (x, a^{\mathcal{I}}) \in R^{\mathcal{I}}, \text{ and } a \in \mathbf{I}_C\}.$$

Now we define a FEA $\mathbf{A}_C$ that accepts a labeled forest $F = \langle \Delta, L \rangle$ if F represents a quasi-forest model $\mathcal{I}$ of C, or if it can be homomorphically embedded into such a forest. Note that $\mathbf{A}_C$ can not ensure that an accepted forest is *nominal unique*, i.e., that each $a \in \mathbf{I}_C$ occurs in exactly one root; instead, it enforces that any two roots sharing a nominal are indistinguishable by its transition function.

The construction, given for a normal $\mathcal{ZOIQ}$ concept, combines in a novel way techniques that had been used separately for different combinations of nominals, inverses, and counting [Bonatti *et al.*, 2008; Sattler and Vardi, 2001], with techniques for Boolean roles [Calvanese *et al.*, 2002; 2007] and $\exists S.\mathsf{Self}$ concepts [Ortiz, 2008]. As Bonatti *et al.*, we employ FEAs; however, while they and Sattler and Vardi build an automaton for a specific *guess* (a partition of the nominals into equivalence classes that are interpreted as the same root, and a set of atomic concepts satisfied by each class), we defer the existence of a guess to the emptiness test of $\mathbf{A}_C$. This is more convenient for query answering, although it requires additional states and more involved transitions to properly handle the connections from each node to the nominals.

In what follows, we extend the syntax to allow for negation of simple roles and of the symbols in $\Theta(C)$. We let $\sim E$ denote the NNF of an expression E, and $\mathsf{Inv}(S)$ denote the role obtained from a simple role S by replacing P by P^- and P^- by P, for each $P \in \mathbf{R}_C$. The *(syntactic) closure* $Cl(C)$ of a $\mathcal{ZOIQ}$ concept C contains all concepts and simple roles that are relevant for deciding the satisfiability of C. It contains C and is closed under subconcepts, subroles, $\sim$, and $\mathsf{Inv}(S)$. Concerning concepts with regular role expressions, it is analogous to the standard Fischer-Ladner closure of PDL. Formally, $Cl(C)$ is as in [Ortiz, 2008] (where

atomic concepts may now be nominal concepts), extended with $\geq 1 S.C$ for each $\exists S.C \in Cl(C)$, and with $\leq 0 S.\sim C$ for each $\forall S.C \in Cl(C)$, with S a simple role.

Definition 3.8. Let b_C denote the maximal number n occurring in a number restriction in C, and let $a_1, \ldots, a_k$ be a fixed, arbitrary enumeration of the elements of $\mathbf{I}_C$. The automaton $\mathbf{A}_C = \langle \Sigma_C, b_C, Q_C, \delta_C, q_C^0, F_C \rangle$ is defined as follows:

- $\Sigma_C = 2^{\Theta(C)}$ is as in Definition 3.7;

- $Q_C = Cl(C) \cup \{q_C^0\} \cup \overline{\Theta}(C) \cup Q_{\mathbf{ICl}} \cup \cdots \cup Q_{\mathsf{bin}}$,

where $\overline{\Theta}(C) = \{s, \neg s \mid s \in \Theta(C)\}$ and $Q_{\mathbf{ICl}}, \ldots, Q_{\mathsf{bin}}$ are explained below, along with the transition function. Below, $\mathbf{S}$ and $\mathbf{C}$ respectively contain all simple roles $S, \sim S$ and concepts $C, \sim C$ such that $\geq n S.C$ or $\leq n S.C$ is in $Cl(C)$, $\overline{\mathbf{I}}$ contains each $\{a\}$ and $\neg\{a\}$ such that $a \in \mathbf{I}_C$, and $\mathbf{CL} = Cl(C) \cup \Theta(C)$.

$$Q_{\mathbf{ICl}} = \{\langle a, \alpha \rangle, \langle a, \sim\alpha \rangle \mid a \in \mathbf{I}_C, \alpha \in \mathbf{CL}\}$$
$$Q_\uparrow = \{\uparrow_a^S \mid S \in Cl(C) \text{ is a simple role and } a \in \mathbf{I}_C\}$$
$$Q_{\mathsf{Self}} = \{S_{\mathsf{Self}} \mid S \in Cl(C) \text{ is a simple role}\}$$
$$Q_{\mathsf{Nom}} = \{\neg a \vee \neg a' \mid a, a' \in \mathbf{I}_C, a \neq a'\}$$
$$Q_{\mathsf{roots}} = \{\langle \langle \mathsf{root} \rangle^i, S, C \rangle \mid 0 \leq i \leq |\mathbf{I}_C|, S \in \mathbf{S}, C \in \mathbf{C}\} \cup \{\langle [\mathsf{root}]^i, S, C \rangle \mid 0 \leq i \leq |\mathbf{I}_C|, S \in \mathbf{S}, C \in \mathbf{C}\}$$
$$Q_{\mathsf{bin}} = \{\langle \circ, \alpha, C \rangle \mid \circ \in \{\wedge, \vee\}, \alpha \in \mathbf{S} \cup \overline{\mathbf{I}}, C \in \mathbf{C}\}$$

- $F_C = (\emptyset, \{\forall R^*.C \mid \forall R^*.C \in Cl(C)\}, Q_C)$ is the acceptance condition [Calvanese *et al.*, 2002; 2007].

- There are transitions for each $\sigma \in \Sigma_C$ as defined below. First, we have

$$\delta_C(q_C^0, \sigma) = (\langle \mathsf{root} \rangle, C) \wedge \bigwedge_{i=1}^k (\langle \mathsf{root} \rangle, \{a_i\}) \wedge \bigwedge_{a \in \mathbf{I}_C, \alpha \in \mathbf{CL}} (([\mathsf{root}], \langle a, \alpha \rangle) \vee ([\mathsf{root}], \langle a, \sim\alpha \rangle)).$$

This initial transition checks that the input forest encodes a quasi-model of C. Its three conjuncts respectively check that (i) some root is in the interpretation of C, (ii) each nominal is interpreted as some root, and (iii) all pairs of roots interpreting the same nominals have identical labels and satisfy the same expressions in the closure. For testing (iii), $\mathbf{A}_C$ moves to the states in $Q_{\mathbf{ICl}}$. For each such state, there are transitions

$$\delta_C(\langle a, \alpha \rangle, \sigma) = (\varepsilon, \neg\{a\}) \vee (\varepsilon, \alpha).$$

Transitions that use the states $Cl(C)$ to inductively decompose simple roles, concepts (except number restrictions and $\exists S.\mathsf{Self}$ concepts), and regular role expressions within concepts are as usual, see e.g., [Calvanese *et al.*, 2002; 2007]. We recall that propagation of $\forall T.C$ (resp., $\exists T.C$) in the case where $T = R^*$ is by $\delta_C(\forall R^*.C, \sigma) = (\varepsilon, C) \wedge (\varepsilon, \forall R.\forall R^*.C)$ (resp., $\delta_C(\exists R^*.C, \sigma) = (\varepsilon, C) \vee (\varepsilon, \exists R.\exists R^*.C)$). For each $\exists S.\mathsf{Self}$ in $Cl(C)$, we have

$$\delta_C(\exists S.\mathsf{Self}, \sigma) = (\varepsilon, S_{\mathsf{Self}}),$$

as in [Ortiz, 2008]. For each $\exists S.C$ and $\forall S.C$ in $Cl(C)$ where S is simple, we respectively have

$$\delta_C(\exists S.C, \sigma) = (\varepsilon, \geq 1 S.C), \quad \text{and}$$
$$\delta_C(\forall S.C, \sigma) = (\varepsilon, \leq 0 S.\sim C).$$

We next give the transitions that ensure satisfaction of the number restrictions. They are novel and differ from all previous approaches. In $\mathcal{ZOIQ}$, to ensure that $\geq n S.C$ or

$\leq n\, S.C$ is satisfied at some node x of a forest F we must take all nodes x' into account for which $(x, x') \in S^{\mathcal{I}}$ may hold in the encoded interpretation $\mathcal{I}$. This x' may be (cf. Def. 3.2): (i) a root of F, (ii) x itself, (iii) a node in $\mathsf{succ}(x)$, or (iv) the predecessor of x in F. Our transitions are more involved than those in [Bonatti *et al.*, 2008] for two reasons. First, we must consider the four cases above, while they consider either just (i) and (iii) or (iii) and (iv). Second, as we are not building an automaton for a *specific* guess, verifying which roots of F take part in the satisfaction of a number restriction is more complicated, and special care is needed to ensure that roots interpreting more than one nominal are not counted more than once. The transitions differ also from those in [Calvanese *et al.*, 2002; 2007; Ortiz, 2008], which use non-graded automata and count the successors of x one-by-one; this requires exponentially many states if n is coded in binary.

For each $\geq n\, S.C$ in $Cl(C)$, we define:

$$\delta_C(\geq n\, S.C, \sigma) = \bigvee_{0 \leq i \leq |\mathbf{I}_C|} ((\varepsilon, \langle\langle\mathsf{root}\rangle^i, S, C\rangle) \wedge \mathsf{NR}^{\wedge}(n - i, S, C))$$

where for $m \geq 0$, $\mathsf{NR}^{\wedge}(m, S, C) = \varphi_1 \vee \varphi_2 \vee \varphi_3 \vee \varphi_4$ and

$$
\begin{aligned}
\varphi_1 &= (\langle m \rangle, \langle \wedge, S, C \rangle), \\
\varphi_2 &= (\varepsilon, S_{\mathsf{Self}}) \wedge (\varepsilon, C) \wedge (\langle m{-}1 \rangle, \langle \wedge, S, C \rangle), \\
\varphi_3 &= (\varepsilon, \mathsf{Inv}(S)) \wedge (-1, C) \wedge (\langle m{-}1 \rangle, \langle \wedge, S, C \rangle), \\
\varphi_4 &= (\varepsilon, S_{\mathsf{Self}}) \wedge (\varepsilon, C) \wedge (\varepsilon, \mathsf{Inv}(S)) \wedge (-1, C) \wedge \\
& \quad\ (\langle m{-}2 \rangle, \langle \wedge, S, C \rangle).
\end{aligned}
$$

To understand these transitions, suppose satisfaction of $\geq n\, S.C$ is verified at node x. Then, among the nodes of type (i) to (iv) above, there must exist distinct nodes $x'_1, \ldots, x'_n$ for which the following holds: (*) x is related to x'_j via S and x'_j satisfies C. These nodes are grouped further into two kinds: the roots, in (i), and nodes that are not roots, in (ii) to (iv). The first transition searches for some i such that i nodes of the first group and $m = n{-}i$ of the second group satisfy (*). The latter check is done via $\mathsf{NR}^{\wedge}(m, S, C)$. Its disjuncts φ_1 to φ_4 correspond to the four possible ways in which these m nodes can be found among the nodes of types (ii) to (iv), viz.:

(φ_1) m successors of x satisfy (*),

(φ_2) x itself and (at least) $m{-}1$ successors of x satisfy (*),

(φ_3) the predecessor of x and (at least) $m{-}1$ successors of x satisfy (*), or

(φ_4) x itself, the predecessor of x, and (at least) $m{-}2$ successors of x satisfy (*).

Finally, the following transitions for each $\langle\langle\mathsf{root}\rangle^i, S, C\rangle$ in Q_{roots} check whether i roots satisfy (*):

$$\delta_C(\langle\langle\mathsf{root}\rangle^i, S, C\rangle, \sigma) = \bigvee_{N \subseteq \mathbf{I}_C, |N|=i} \left(\mathsf{R}^{\wedge}(N, S, C)\right), \text{ where}$$

$$
\begin{aligned}
\mathsf{R}^{\wedge}(N, S, C) = &\bigwedge_{a \in N}((\varepsilon, \uparrow_a^S) \wedge (\langle\mathsf{root}\rangle, \langle\wedge, \{a\}, C\rangle)) \wedge \\
&\bigwedge_{a_i, a_j \in N, i \neq j}([\mathsf{root}], \neg a_i \vee \neg a_j).
\end{aligned}
$$

The transitions for concepts $\leq n\, S.C$ are analogous. They use states $\langle[\mathsf{root}]^i, S, C\rangle$ in Q_{roots}, states $\langle\vee, \alpha, C\rangle$ in Q_{bin}, and states in Q_{Nom}. Note that in the above transitions, the automaton moves to the auxiliary states in Q_{roots} and Q_{bin}. Then, for each $\neg a \vee \neg a'$ in Q_{Nom}, there is a transition

$$\delta_C(\neg a \vee \neg a', \sigma) = (\varepsilon, \neg\{a\}) \vee (\varepsilon, \neg\{a'\})$$

and for each $\langle\circ, \alpha, C\rangle$ in Q_{bin}, there is a transition

$$\delta_C(\langle\circ, \alpha, C\rangle, \sigma) = (\varepsilon, \alpha) \circ (\varepsilon, C).$$

The automaton also moves to the states S_{Self} in Q_{Self} (resp. $\uparrow_a^S$ in $Q_\uparrow$), to verify whether a node x is connected by S to itself (resp. to a root where a holds). Then the simple role S is decomposed using, for each $q \in Q_\uparrow$,

$$
\begin{aligned}
\delta_C(\uparrow_a^{S \cap S'}, \sigma) &= (\varepsilon, \uparrow_a^S) \wedge (\varepsilon, \uparrow_a^{S'}), \\
\delta_C(\uparrow_a^{S \cup S'}, \sigma) &= (\varepsilon, \uparrow_a^S) \vee (\varepsilon, \uparrow_a^{S'}), \\
\delta_C(\uparrow_a^{S \setminus S'}, \sigma) &= (\varepsilon, \uparrow_a^S) \wedge (\varepsilon, \neg\uparrow_a^{S'}),
\end{aligned}
$$

and similar transitions for all $q \in Q_{\mathsf{Self}}$.

Finally, the automaton checks the label of the current node for atomic expressions and special symbols. For each $s \in \Theta(C)$,

$$\delta_C(s, \sigma) = \begin{cases} \mathbf{t} & \text{if } s \in \sigma, \\ \mathbf{f} & \text{if } s \notin \sigma, \end{cases} \qquad \delta_C(\neg s, \sigma) = \begin{cases} \mathbf{f} & \text{if } s \in \sigma, \\ \mathbf{t} & \text{if } s \notin \sigma. \end{cases}$$

The automaton $\mathbf{A}_C$ provides the desired reduction.

Lemma 3.9. *Let C be a normal $\mathcal{ZOIQ}$ concept. If $\mathscr{L}(\mathbf{A}_C) \neq \emptyset$ then C is satisfiable, and if C has a quasi-forest model, then $\mathscr{L}(\mathbf{A}_C) \neq \emptyset$.*

Proof (sketch). If C has a quasi-forest model, it is routine to verify that $\mathbf{A}_C$ accepts its forest encoding. The converse is less direct, as $\mathbf{A}_C$ also accepts forests F that are not nominal unique. However, one can verify that if F has two subtrees whose roots r and r' have the same labels and satisfy the same concepts in the closure, then a run of $\mathbf{A}_C$ on F visiting both can be modified into one visiting only one of them. Hence, if $\mathscr{L}(\mathbf{A}_C) \neq \emptyset$ then $\mathbf{A}_C$ accepts some nominal-unique F. To show that such an F corresponds to a model of C is easy. $\square$

From this and Propositions 3.1 and 3.3, we obtain:

Theorem 3.10. *Let $\mathcal{K}$ be a $\mathcal{ZIQ}$, $\mathcal{ZOI}$, or $\mathcal{ZOQ}$ KB. Then we can construct from $\mathcal{K}$ a concept $C_\mathcal{K}$ such that $\mathscr{L}(\mathbf{A}_{C_\mathcal{K}}) \neq \emptyset$ iff $\mathcal{K}$ is satisfiable.*

For a given KB $\mathcal{K}$ and the concept $C_\mathcal{K}$ obtained from it, one can easily verify that for $\mathbf{A}_{C_\mathcal{K}}$ (i) the number of states is polynomial in the size of $\mathcal{K}$, (ii) the alphabet size and the counting bound are at most single exponential, even when numbers are coded in binary, and (iii) the index is fixed. Hence Theorems 3.6 and 3.10 yield an EXPTIME upper bound for KB satisfiability in all sublogics of $\mathcal{ZOIQ}$ that enjoy the quasi-forest model property. Since a matching lower bound is known for much weaker DLs, we obtain our first main result.

Theorem 3.11. *KB satisfiability in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ is EXPTIME-complete.*

4 Query Entailment and Containment

To decide query entailment, we follow the ideas in [Calvanese *et al.*, 2007], which use complementation, projection, and intersection of automata. How to complement and do projection on FEAs is open. Therefore, we exploit the fact that root transitions can be easily removed from FEAs [Bonatti *et al.*, 2008], and then show how to eliminate also graded transitions, obtaining an automaton for which we know how to perform complementation and projection.

For $b > 0$, let $D'_b = \{-1, \varepsilon\} \cup \{\langle 0 \rangle, \ldots, \langle b \rangle\} \cup \{[0], \ldots, [b]\}$. A *two-way graded alternating parity tree automaton (2GAPA)* is a FEA with transition function $\delta : Q \times \Sigma \to \mathcal{B}(D'_b \times Q)$, i.e., there are no $\langle\mathsf{root}\rangle$ or $[\mathsf{root}]$ transitions.

A FEA **A** with counting bound b and s states is convertible into an 2GAPA with the same index and counting bound and with $O(s \cdot b)$ states, accepting the *tree-encoding* of each forest in $\mathscr{L}(\mathbf{A})$ [Bonatti *et al.*, 2008]. The *tree encoding* of a forest F is the tree obtained from F by placing a new root above the roots of F.

We now show how to eliminate graded transitions from a 2GAPA. For this, we need to restrict the attention to trees with a bounded branching degree $k \geq 1$. We call a tree $T \subseteq \{1, \ldots, k\}^*$ with $\mathsf{roots}(T) = \{\varepsilon\}$ a *k-tree*.

Definition 4.1. A *two-way alternating parity automaton* (2APA) over infinite Σ-labeled k-trees is a tuple $\mathbf{A} = \langle \Sigma, k, Q, \delta, q_0, F \rangle$, where Σ, Q, q_0, and F are as for FEAs, and the transition function is $\delta : Q \times \Sigma \to \mathcal{B}([-1..k] \times Q)$, with $[-1..k] = \{-1, \varepsilon, 1, \ldots, k\}$.

A *run* $\langle T_r, r \rangle$ of a 2APA over a labeled k-tree $\langle T, V \rangle$ is a $T \times Q$-labeled tree satisfying: $\varepsilon \in T_r$; $r(\varepsilon) = (\varepsilon, q_0)$; and for each $y \in T_r$ with $r(y) = (x, q)$ and $\delta(q, V(x)) = \varphi$, there is a (possibly empty) set $\{(d_1, q_1), \ldots, (d_n, q_n)\} \subseteq [-1..k] \times Q$ that satisfies φ and such that, for all $i \in \{1, \ldots, n\}$, $y \cdot i \in T_r$, $x \cdot d_i$ is defined, and $r(y \cdot i) = (x \cdot d_i, q_i)$.

Lemma 4.2. *Let $\mathbf{A} = \langle \Sigma, b, Q, \delta, q_0, F \rangle$ be a 2GAPA and $k \geq 1$. There is a 2APA $\mathbf{A}'$ with the same index and $O(|Q| \cdot b \cdot k)$ states accepting the same set of Σ-labeled k-trees as $\mathbf{A}$.*

Proof (sketch). We let $\mathbf{A}' = \langle \Sigma, k, Q \uplus Q', \delta', q_0, F' \rangle$, where

$$Q' = \{\langle i, q, j \rangle, [i, q, j] \mid q \in Q, 0 \leq i \leq b{+}1, 1 \leq j \leq k{+}1\}.$$

For each $\sigma \in \Sigma$, the transition function δ' is defined as follows. First, for all $q \in Q$, $\delta'(q, \sigma)$ is obtained from $\delta(q, \sigma)$ by replacing each $(\langle n \rangle, q)$ with $(\varepsilon, \langle n{+}1, q, 1 \rangle)$ and each $([n], q)$ with $(\varepsilon, [n{+}1, q, 1])$. For $1 \leq i \leq b{+}1$ and $1 \leq j \leq k$, we define:

$$\delta'(\langle i, q, j \rangle, \sigma) = ((j, q) \wedge (\varepsilon, \langle i{-}1, q, j{+}i \rangle)) \vee (\varepsilon, \langle i, q, j{+}1 \rangle)$$
$$\delta'([i, q, j], \sigma) = ((j, q) \wedge (\varepsilon, [i, q, j{+}1])) \vee (\varepsilon, [i{-}1, q, j{+}1])$$

and additionally, we have:

$$\delta'(\langle 0, q, j \rangle, \sigma) = \mathbf{t}, \quad \delta'([0, q, j], \sigma) = \mathbf{f}, \text{ for } 1 \leq j \leq k{+}1;$$
$$\delta'(\langle i, q, k{+}1 \rangle, \sigma) = \mathbf{f}, \delta'([i, q, k{+}1], \sigma) = \mathbf{t}, \text{ for } 1 \leq i \leq b{+}1.$$

Intuitively, from state $\langle i, q, j \rangle$, a copy of $\mathbf{A}'$ is sent off in state q, to at least i successor nodes starting from the j-th one. Similarly, from state $[i, q, j]$, no copy of $\mathbf{A}'$ is sent off in state q, for at most $i{-}1$ successor nodes starting from the j-th one. Finally, if $F = (G_1, \ldots, G_{n-1}, Q)$, we have

$$F' = (G_1, \ldots, G_{n-1}, Q \uplus Q').$$

One can show that a Σ-labeled k-tree is accepted by $\mathbf{A}'$ iff it is accepted by $\mathbf{A}$. $\square$

Note that Lemma 4.2 does not ensure equivalence between the 2GAPA and the resulting 2APA, since a 2GAPA may accept trees of arbitrary degree. However, it is sufficient for our purposes: an analysis of the proof of Proposition 3.3 reveals that, if $\mathcal{K} \not\models q$ for $\mathcal{K}$ in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$,[4] then there is a counterexample quasi-forest model $\mathcal{I}$ of degree bounded

---

[4]We assume w.l.o.g. that all concepts appearing in q are atomic, and that all concept and role names occurring in q occur also in $\mathcal{K}$.

by $b_{\mathcal{K}} \cdot |Cl(C)|$. Hence, the tree encoding of $\mathcal{I}$ has degree bounded by the maximum k of $b_{\mathcal{K}} \cdot |Cl(C)|$ and $|\mathbf{I}_{\mathcal{K}}|$.

For deciding the entailment of a P2RPQ q, we use the technique in [Calvanese *et al.*, 2007] extended to $\mathcal{ZOIQ}$ (nominals in q are handled essentially as the query constants there, while Self is handled as in [Ortiz, 2008]). For a k-tree T whose nodes are labeled with subsets of $\Theta(C)$, let $\mathcal{I}_T$ be the interpretation of $\mathcal{K}$ represented by T. We define from q a 2APA $\mathbf{A}_q^{\mathcal{X}}$ that accepts a labeled k-tree $T^{\mathcal{X}}$ iff it explicitly finds a match for $\mathcal{I}_{T^{\mathcal{X}}}$ and q on the nodes of $T^{\mathcal{X}}$. Technically, the node labels of $T^{\mathcal{X}}$ are allowed to contain, besides elements of $\Theta(C)$, also elements of $\mathcal{X}$, where $\mathcal{X}$ is the set of (existentially quantified) variables in q. The elements of $\mathcal{X}$ are treated as atomic concepts that are enforced to hold in a single node of $T^{\mathcal{X}}$ (on a tree-structure, such a condition can be easily enforced by means of a 2APA), and $\mathbf{A}_q^{\mathcal{X}}$ relies on such elements to check for a match for $\mathcal{I}_{T^{\mathcal{X}}}$ and q. We then convert $\mathbf{A}_q^{\mathcal{X}}$ to a one-way nondeterministic parity automaton (1NPA) $\mathbf{A}_q^1$, from which we then *project out*[5] the elements of $\mathcal{X}$, obtaining a 1NPA $\mathbf{A}_q^2$. In this way, $\mathbf{A}_q^2$ accepts a k-tree T whose nodes are labeled with subsets of $\Theta(C)$, iff there exists a match for $\mathcal{I}_T$ and q. By *complementing* $\mathbf{A}_q^2$, we obtain a 1NPA $\mathbf{A}_{\neg q}$ accepting a k-tree T iff there is no match for $\mathcal{I}_T$ and q. Finally, to check $\mathcal{K} \models q$, we transform the FEA $\mathbf{A}_{C_{\mathcal{K}}}$ to a 2GAPA, then to a 2APA (cf. Lemma 4.2), and finally to a 1NPA, which we intersect with $\mathbf{A}_{\neg q}$. A complexity analysis of the various operations allows us to show the following:

Theorem 4.3. *Given a KB $\mathcal{K}$ in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$ and a P2PRQ q, deciding $\mathcal{K} \models q$ is in 2EXPTIME in the total size of q and $\mathcal{K}$ (under unary number coding in number restrictions).*

To address query containment, we extend the relationship with query answering, which is well-known for plain CQs in the relational case, to our richer setting. Indeed, $\mathcal{K} \models q_1 \subseteq q_2$ iff $\mathcal{K}_{q_1} \models q_2$, where $\mathcal{K}_{q_1} = \langle \mathcal{A}_{q_1}, \mathcal{T} \rangle$ is the KB obtained from $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ by first "freezing" $q_1 = \exists \vec{x}. \varphi(\vec{x})$, i.e., considering each variable in $\vec{x}$ as a fresh individual in $\mathcal{K}_{q_1}$, and then asserting $\varphi(\vec{x})$ to hold in $\mathcal{A}_{q_1}$. When $\varphi(\vec{x})$ is (or can be reduced to) a conjunction $\bigwedge_{1 \leq i \leq n} \alpha_i$ of atoms, where each α_i is of the form $C(z)$ or $S(z, z')$, with S a simple role, we have that $\mathcal{A}_{q_1} = \mathcal{A} \cup \{\alpha_i \mid 1 \leq i \leq n\}$, i.e., the frozen q_i can be directly represented as an ABox. Otherwise, we can represent the whole of $\varphi(\vec{x})$ by means of a single ABox assertion $C_{\varphi(\vec{x})}(a)$, where a is a fresh individual, and $C_{\varphi(\vec{x})}$ is the concept obtained from $\varphi(\vec{x})$ by replacing $\wedge$ by $\sqcap$, $\vee$ by $\sqcup$, each atom $C(z)$ by $\neg\{z\} \sqcup C$, and each atom $T(z, z')$ by $\neg\{z\} \sqcup \exists T.\{z'\}$. Note that in the latter case we need to introduce nominals, even when they were not present in $\mathcal{K}$.

Theorem 4.4. *$\mathcal{K} \models q_1 \subseteq q_2$ is in 2EXPTIME wrt. the total size of q_1, q_2, and $\mathcal{K}$ (i) if $\mathcal{K}$ is a $\mathcal{ZOQ}$ or $\mathcal{ZOI}$ KB and q_1, q_2 are P2RPQs over $\mathcal{K}$, or (ii) if $\mathcal{K}$ is a $\mathcal{ZIQ}$ KB, q_1 is a conjunctive query, and q_2 is a P2RPQ over $\mathcal{K}$.*

5 Reasoning in the $\mathcal{SR}$ family

The automata techniques devised above can be also fruitfully exploited for fragments of the DL.

---

[5]Projection cannot be done directly on a two-way automaton.

$\mathcal{SROIQ}$ is similar to $\mathcal{ZOIQ}$, but lacks Boolean and regular role expressions. Instead, it has an RBox $\mathcal{R}$ comprising (*i*) *role inclusions* $R_1 \circ \cdots \circ R_n \sqsubseteq R$ under certain restrictions, and (*ii*) *assertions about roles* $\mathsf{Irr}(R)$, $\mathsf{Ref}(R)$, $\mathsf{Sym}(R)$, $\mathsf{Dis}(R, R')$ [Horrocks *et al.*, 2008]. Its sublogics $\mathcal{SRIQ}$, $\mathcal{SROQ}$, $\mathcal{SROI}$ are analogous to $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$.

To exploit our automata-based algorithms for reasoning in (sublogics of) $\mathcal{SROIQ}$, we can transform each $\mathcal{SROIQ}$ KB $\mathcal{K}$ into a $\mathcal{ZOIQ}$ KB $\Psi(\mathcal{K})$. The rewriting $\Psi(\mathcal{K})$ is like the one in [Ortiz, 2008] from $\mathcal{SRIQ}$ to $\mathcal{ZIQ}$ (alias $\mathcal{ALCQIb}^+_{reg}$); as nominal concepts cause no change, we do not repeat it here. Intuitively, the rewriting replaces each role R in $\mathcal{K}$ with a regular expression ρ_R. We note that this need not be done for the roles in the ABox, which can be treated by closing the role assertions wrt. the RBox. The models of $\mathcal{K}$ are models of the resulting $\Psi(\mathcal{K})$, and conversely, each model $\mathcal{I}$ of $\Psi(\mathcal{K})$ can be turned into a model of $\mathcal{K}$ by setting $R^{\mathcal{I}} = (\rho_R)^{\mathcal{I}}$ (cf. [Kazakov, 2008]). Assertions about roles are simulated using BRIAs and CIAs. Based on this, we obtain:

Proposition 5.1. *A $\mathcal{SROIQ}$ KB $\mathcal{K}$ can be rewritten into an equisatisfiable $\mathcal{ZOIQ}$ KB $\Psi(\mathcal{K})$. Further, if $\mathcal{K}$ is in $\mathcal{SRIQ}$, $\mathcal{SROI}$, or $\mathcal{SROQ}$, then $\Psi(\mathcal{K})$ is in $\mathcal{ZIQ}$, $\mathcal{ZOI}$, or $\mathcal{ZOQ}$, respectively.*

The $\mathcal{ZOIQ}$ KB $\Psi(\mathcal{K})$ can be constructed in time polynomial in the combined sizes of $\mathcal{A}$, $\mathcal{T}$, and the largest regular expression ρ_R used, denoted by $\rho_{\mathcal{K}}$ (which can be exponential in $\mathcal{R}$) [Ortiz, 2008]. Hence we obtain:

Theorem 5.2. *Satisfiability of a KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ in any of $\mathcal{SRIQ}$, $\mathcal{SROI}$, and $\mathcal{SROQ}$ is in* EXPTIME *wrt. the total size of $\mathcal{T}$, $\mathcal{A}$, and $\rho_{\mathcal{K}}$, and in* 2EXPTIME *wrt. the size of $\mathcal{K}$.*

For $\mathcal{SRIQ}$, this is known to be optimal [Kazakov, 2008]. Note that this holds even if the number restrictions are coded in binary, and that we obtain a single exponential upper bound whenever $\rho_{\mathcal{K}}$ is polynomial in $\mathcal{R}$.

By making again use of the rewriting above, we can reduce also query answering in $\mathcal{SROIQ}$ to $\mathcal{ZOIQ}$.

Proposition 5.3. *Let $\mathcal{K}$ be a $\mathcal{SROIQ}$ KB and q a P2RPQ over $\mathcal{K}$, and let q' be obtained from q by replacing each occurrence of each role R by ρ_R. Then $\mathcal{K} \models q$ iff $\Psi(\mathcal{K}) \models q'$.*

Note that the rewriting of q into q' may introduce regular expressions, even if they were not originally present in q. To verify $\mathcal{K} \models q_1 \subseteq q_2$, for two P2RPQs q_1 and q_2, then we can proceed by "freezing" q_1 and treating it as an ABox, as described in Section 4. We obtain the following upper bounds:

Theorem 5.4. *Given $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ and q_1, q_2 P2RPQs over $\mathcal{K}$, $\mathcal{K} \models q_2$ and $\mathcal{K} \models q_1 \subseteq q_2$ are decidable in* 2EXPTIME *in the total size of $\mathcal{A}$, $\mathcal{T}$, $\rho_{\mathcal{K}}$, q_1, q_2 and in* 3EXPTIME *in the total size of $\mathcal{K}$, q_1, q_2 (assuming unary coding of numbers in the number restrictions) when (i) $\mathcal{K}$ is a $\mathcal{SROQ}$ or $\mathcal{SROI}$ KB, or (ii) $\mathcal{K}$ is a $\mathcal{SRIQ}$ KB and q_1 is a conjunctive query.*

Our results also apply to the corresponding $\mathcal{SR}$ logics extended with safe Boolean roles, as in [Rudolph *et al.*, 2008].

6 Conclusion

In this paper, we have substantially pushed the frontier of decidability for query entailment and containment over very expressive DLs, and in particular for the DLs of the $\mathcal{SR}$ family

that underlie relevant fragments of OWL. Our techniques rely heavily on the quasi-forest model property of the considered DLs, and their applicability for settings where this fails is not apparent. Indeed, the fact that Proposition 3.3 fails in $\mathcal{ALCB}$, the extension of $\mathcal{ALC}$ with arbitrary role negation, already implies undecidability of query entailment [Pratt-Hartmann, 2008]. We remark that [Pratt-Hartmann, 2008] shows also decidability of CQ answering for the guarded two-variable fragment of FOL, but the latter captures neither nominals nor regular expressions over roles. In fact, the decidability of $\mathcal{ZOIQ}$ remains open, even for KB satisfiability.

References

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

[Bonatti *et al.*, 2008] Piero Bonatti, Carsten Lutz, Aniello Murano, and Moshe Y. Vardi. The complexity of enriched μ-calculi. *Logical Methods in Computer Science*, 4(3:11):1–27, 2008.

[Calvanese *et al.*, 2000] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Containment of conjunctive regular path queries with inverse. In *Proc. of KR 2000*.

[Calvanese *et al.*, 2002] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. 2ATAs make DLs easy. In *Proc. of DL 2002*, volume 53 of *CEUR,* ceur-ws.org.

[Calvanese *et al.*, 2007] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proc. of AAAI 2007*.

[Cuenca Grau *et al.*, 2008] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *J. of Web Semantics*, 6(4):309–322, 2008.

[Glimm *et al.*, 2008] Birte Glimm, Ian Horrocks, and Ulrike Sattler. Unions of conjunctive queries in $\mathcal{SHOQ}$. In *Proc. of KR 2008*.

[Horrocks *et al.*, 2006] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of KR 2006*.

[Kazakov, 2008] Yevgeny Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proc. of KR 2008*.

[Ortiz *et al.*, 2008] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter. Data complexity of query answering in expressive description logics via tableaux. *J. of Automated Reasoning*, 41(1):61–98, 2008.

[Ortiz, 2008] Magdalena Ortiz. An automata-based algorithm for description logics around $\mathcal{SRIQ}$. In *Proc. of LANMR 2008*, volume 408 of *CEUR,* ceur-ws.org.

[Pratt-Hartmann, 2008] Ian Pratt-Hartmann. Data complexity of the two-variable fragment with counting quantifiers. Available at http://arxiv.org/abs/0806.1636/, 2008.

[Rudolph *et al.*, 2008] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Cheap boolean role constructors for description logics. In *Proc. of JELIA 2008*, volume 5293 of *LNCS*.

[Sattler and Vardi, 2001] Ulrike Sattler and Moshe Y. Vardi. The hybrid μ-calculus. In *Proc. of IJCAR 2001*, pages 76–91.

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