Data Complexity in Expressive Description Logics With Path Expressions (Extended Abstract)

Bartosz Bednarczyk[1,2]

[1]Computational Logic Group, Technische Universität Dresden, Germany
[2]Institute of Computer Science, University of Wrocław, Poland

1. Introduction

Among the plethora of various features available in extensions of the basic DL called $\mathcal{ALC}$, an especially prominent one is $\cdot_{\mathsf{reg}}$, supported by the popular $\mathcal{Z}$-family of DLs [1]. Using $\cdot_{\mathsf{reg}}$, one can specify regular path constraints and hence, allow the user to navigate the underlying graph data-structure. In recent years, many extensions of $\mathcal{ALC}_{\mathsf{reg}}$ for ontology-engineering were proposed [2, 3, 4], and the complexity of their reasoning problems is mostly well-understood [1, 5, 6]. Many years ago, Vardi [7] proposed the vital notion of *data complexity*. When measuring the complexity of the knowledge-base (KB) *satisfiability problem* (KBSat), we treat the ontology (TBox) as fixed upfront and only the user's data (ABox) varies. The *satisfiability* problem for DLs is usually NP-complete w.r.t. the data complexity, including the two-variable counting logic [8] (encoding DLs up to $\mathcal{ALCBIOQ}\mathsf{Self}$) as well as $\mathcal{SROIQ}$ [9], the logical core of OWL2. Regarding DLs with path expressions, NP-completeness of $\mathcal{ALCI}_{\mathsf{reg}}^{\mathsf{Self}}$ [10] was established recently. This also applies to the (non)entailment of positive two-way regular path queries.

In our recent IJCAI paper [11] we study the data complexity of $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$, namely the maximal decidable fragments of $\mathcal{ZOIQ}$ that possess the so-called *quasi-forest model property*, a suitable generalisation of the well-known forest model property for $\mathcal{ALC}$. For the uniformity of our approach, we actually focus on the satisfiability problem for full $\mathcal{ZOIQ}$ but over quasi-forests. We obtain:

Theorem 1. *KBSat for $\mathcal{ZOIQ}$ over quasi-forests is NP-complete w.r.t. the data complexity. In particular, knowledge-base satisfiability of $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ is NP-complete w.r.t. the data complexity.* ◀

This completes the data complexity landscape for decidable fragments of $\mathcal{ZOIQ}$ that remained open for more than a decade, and reproves known results on the $\mathcal{SR}$ family by Kazakov [9].

Recall that *regular path queries* are queries of the form $\mathcal{R}(x, y)$ for a regular expression $\mathcal{R}$ (with tests). The built-in support of regular path expression in the $\mathcal{Z}$ family of DLs allow us then to reduce the non-entailment problem $(\mathcal{A}, \mathcal{T}) \not\models \mathcal{R}(x, y)$ to the satisfiability of $(\mathcal{A}, \mathcal{T} \cup \{\top \sqsubseteq \neg\exists\mathcal{R}.\top\})$. Hence:

Theorem 2. *The entailment problem of regular path queries for $\mathcal{ZOIQ}$-KBs over quasi-forests is coNP-complete w.r.t. the data complexity. In particular, this applies to $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$, and to the corresponding fragments of OWL2, namely $\mathcal{SRIQ}$, $\mathcal{SROQ}$, and $\mathcal{SROI}$.* ◀

We are currently working on lifting the results of Theorem 2 to the class of positive two-way regular path queries. We expect to publish them as the journal version of our IJCAI paper, somewhere next year.

As a bonus result, we show how our algorithm used to establish Theorem 1 can be adapted to the entailment of unions rooted (connected and containing at least one answer variable) conjunctive queries over $\mathcal{ZIQ}$-KB (in terms of combined complexity). The coNExpTime upper bound for the rooted entailment over $\mathcal{ZIQ}$-KBs is supplemented with a novel matching lower bound for $\mathcal{ALC}$ extended with Self.

Theorem 3. *The query entailment problem over $\mathcal{ZIQ}$-KB is coNExpTime-complete for the class of (unions of) rooted conjunctive queries. The lower bound holds already for $\mathcal{ALC}\mathsf{Self}$.* ◀

DL 2024: 37th International Workshop on Description Logics, June 18–21, 2024, Bergen, Norway
bartek@cs.uni.wroc.pl (B. Bednarczyk)
https://bartoszjanbednarczyk.github.io/ (B. Bednarczyk)
0000-0002-8267-7554 (B. Bednarczyk)

2. Preliminaries

We fix countably-infinite pairwise-disjoint sets $\mathbf{N_I}, \mathbf{N_C}, \mathbf{N_R}$ of *individual*, *concept*, and *role names*. We assume that the reader is familiar with the standard definitions concerning description logics (DLs) and we do not recall them here. The set of *simple roles* $\mathbf{N}_{\mathbf{R}}^{\mathrm{sp}}$ is defined with the grammar $s ::= r \in \mathbf{N_R} \mid s^- \mid s \cap s \mid s \cup s \mid s \setminus s$. Simple roles are interpreted as expected by invoking the underlying set-theoretic operations. $\mathcal{ZOIQ}$-KBs $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ are, as usual, composed of ABoxes $\mathcal{A}$ and TBoxes $\mathcal{T}$. We assume that $\mathcal{A}$ only contains axioms $\pm r(\mathrm{a}, \mathrm{b})$ and $\pm \mathrm{A}(\mathrm{a})$ for non-complex concepts A. $\mathcal{ZOIQ}$-TBoxes are polynomial-time reducible into the following Scott's normal form.

$$\mathrm{A} \equiv \mathrm{B},\ \mathrm{A} \equiv \neg\mathrm{B},\ \mathrm{A} \equiv \{\mathrm{o}\},\ \mathrm{A} \equiv \mathrm{B} \sqcap \mathrm{B}',\ r = s, \mathrm{A} \equiv \exists\mathcal{A}.\top,\ \mathrm{A} \equiv \exists r.\mathsf{Self},\ \mathrm{A} \equiv (\geq n\ r).\top$$

for $\mathrm{A}, \mathrm{B}, \mathrm{B}' \in \mathbf{N_C} \cup \{\top, \bot\}$, $\mathrm{o} \in \mathbf{N_I}$, $r \in \mathbf{N_R}$, $s \in \mathbf{N}_{\mathbf{R}}^{\mathrm{sp}}$, $n \in \mathbb{N}$, and an automaton $\mathcal{A}$ over the finite alphabet composed of simple roles and tests C? for atomic concepts C. As usual, the equivalence $\equiv$ replaces the two concept inclusions $\sqsubseteq$. We interpret $\exists\mathcal{A}.\top$ as the set of all elements d for which there is a path ρ starting from d that realises $\mathcal{A}$ (written: $\rho \models \mathcal{A}$), namely $\mathcal{A}$ accepts some word of the form $w_1 r_1 \ldots w_{|\rho|-1} r_{|\rho|-1} w_{|\rho|}$, where $r_i \in \mathbf{N}_{\mathbf{R}}^{\mathrm{sp}}$ and w_i are (possibly empty) sequences of tests, satisfying $(\rho_i, \rho_{i+1}) \in r_i^{\mathcal{I}}$ and $\rho_i \in \mathrm{C}^{\mathcal{I}}$ for all $i \leq |\rho|$ and tests C? in w_i. The other DL features are defined as usual. We define the DLs $\mathcal{ZOI}$, $\mathcal{ZIQ}$, and $\mathcal{ZOQ}$, respectively, by dropping (a) number restrictions, (b) nominals, and (c) role inverses from the syntax of $\mathcal{ZOIQ}$.

The key ingredients of our paper are *quasi-forests* [1]. We introduce them under the standard set-theoretic reconstruction of the notion of an $\mathbb{N}$-forest as a prefix-closed subset of $\mathbb{N}^+$ without ε.

Definition 1. *Let $\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}$ and $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$ be finite subsets of $\mathbf{N_I}$, $\mathrm{Root} \in \mathbf{N_C}$, and $child, edge \in \mathbf{N_R}$. An interpretation $\mathcal{I}$ is an $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$*-quasi-forest *if its domain $\Delta^{\mathcal{I}}$ is an $\mathbb{N}$-forest,*

$$\mathrm{Root}^{\mathcal{I}} = \Delta^{\mathcal{I}} \cap \mathbb{N} = \{a^{\mathcal{I}} \mid a \in \mathbf{N_I}\} = \{a^{\mathcal{I}} \mid a \in (\mathbf{N}_{\mathbf{I}}^{\mathcal{A}} \cup \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})\},$$
$$child^{\mathcal{I}} = \{(\mathrm{d}, \mathrm{d}\cdot n) \mid \mathrm{d}, \mathrm{d}\cdot n \in \Delta^{\mathcal{I}}, n \in \mathbb{N}\},$$
$$edge^{\mathcal{I}} = \textstyle\bigcup_{r \in \mathbf{N}_{\mathbf{R}}^{sp}} r^{\mathcal{I}},$$

and for all roles $r^{\mathcal{I}}$ and all pairs $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ at least one of the following conditions hold: (i) d and e belong to $\mathrm{Root}^{\mathcal{I}}$, (ii) one of d or e equals $o^{\mathcal{I}}$ for some name $o \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$, (iii) $\mathrm{d} = \mathrm{e}$, (iv) $(\mathrm{d}, \mathrm{e}) \in child^{\mathcal{I}}$, or (v) $(\mathrm{d}, \mathrm{e}) \in (child^-)^{\mathcal{I}}$. A quasi-forest model *of $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ is an $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest satisfying $\mathcal{K}$, where $\mathsf{ind}(\mathcal{A})$ and $\mathsf{ind}(\mathcal{T})$ are sets of all individual names from $\mathcal{A}$ and $\mathcal{T}$.* ◀

The names from $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$ are dubbed *nominals*, and are denoted with decorated letters o. We employ the notions from graph theory such as *node*, *root*, *child*, *parent*, or *descendant*, defined as expected in accordance with $\Delta^{\mathcal{I}}$, $child^{\mathcal{I}}$ and $\mathrm{Root}^{\mathcal{I}}$. For instance, d is a descendant of c whenever $(\mathrm{c}, \mathrm{d}) \in (child^{\mathcal{I}})^+$. A quasi-forest is N-bounded if all nodes have at most N children.

A slightly wider notion of quasi-forests was employed by Calvanese et al. [1, Prop. 3.3] and Ortiz [4, L. 3.4.1, Thm. 3.4.2] to reason about the $\mathcal{Z}$ family. Based on their results we prove:

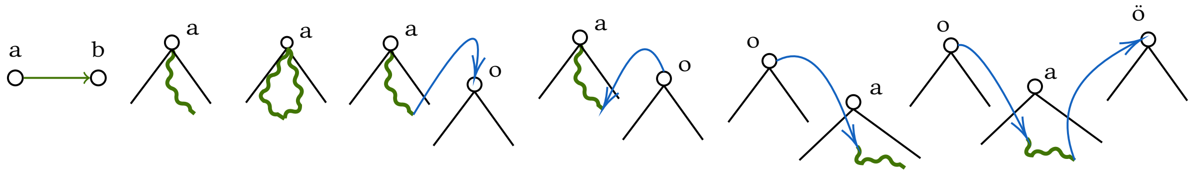
Lemma 1. *Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be a KB of $\mathcal{ZOQ}$, $\mathcal{ZOI}$, or $\mathcal{ZIQ}$. $\mathcal{K}$ is satisfiable iff $\mathcal{K}$ has a quasi-forest model. Moreover, for all (unions of) CQs q, we have $\mathcal{K} \not\models q$ iff there exists an N-bounded quasi-forest model of $\mathcal{K}$ violating q (for some N exponential in $|\mathcal{T}|$). Deciding if a $\mathcal{ZOIQ}$-KB has a quasi-forest model is ExpTime-complete w.r.t. the combined complexity.* ◀

3. Data Complexity: An Overview and Some Tricks

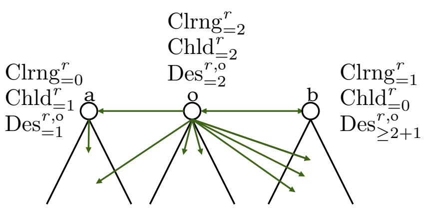
To establish NP-completeness of the satisfiability for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ in a uniform and elegant way, we focus on the satisfiability of $\mathcal{ZOIQ}$ over quasi-forests. As the proof is *highly* technical, we only mention a few of its important ingredients and provide a very rough sketch below.

One of the key ingredients is an exponential time algorithm for deciding if a $\mathcal{ZOIQ}$-KB has a quasi-forest model [4, L. 3.4.1, Thm. 3.4.2]. While this algorithm cannot be used directly to solve the satisfiability problem in optimal time, we stress that we can still employ it as a black-box to decide quasi-forest satisfiability of KBs that have sizes independent from the ABox. In our approach, we intend to construct a quasi-forest model of an input $\mathcal{ZOIQ}$-KB in two steps, *i.e.* we construct its root part (dubbed the *clearing*) separately from its subtrees. Our algorithm first pre-computes (an exponential w.r.t. the size of the TBox but of constant size if the TBox is fixed) set of quasi-forest-satisfiable $\mathcal{ZOIQ}$-concepts that indicate possible subtrees that can be "plugged in" to the clearing of the intended model. Then it guesses (in NP) the intended clearing and verifies its consistency in PTime based on the pre-computed concepts and roles. For the feasibility of our "modular construction" a lot of bookkeeping needs to be done. Most importantly, certain *decorations* are employed to "relativise" and decide the satisfaction of *automata concepts* and *number restrictions* by elements in an incomplete, fragmented forest.

I. The first type of decorations, given an automaton $\mathcal{A}$, aggregate information about existing paths realising $\mathcal{A}$ and starting at one of the roots of the intended model. As a single such path may visit several subtrees, we cut such paths into relevant pieces and summarise them by means of "shortcut" roles and $\mathcal{ZOIQ}$-concepts describing paths fully contained inside a single subtree. One of the core results is a complete characterisation of how paths in quasi-forest look like, and how they can be decomposed into interesting pieces, axiomatisable in $\mathcal{ZOIQ}$. Such "basic" paths are depicted below (the decorated o denote nominals, while the other letters denote arbitrary roots).



II. The second type of decorations "localise" counting in the presence of nominals, as the nominals may have successors outside their own subtree and the clearing. For instance, suppose that we deal with the number restriction $(\geq 2\ r).\top$ in an $(\{a, b\}, \{o\})$-quasi-forest $\mathcal{I}$ sketched below (with r depicted as a green arrow). For different thresholds $\mathfrak{t} \in \{=0, =1, =2, >2\}$ we label the roots of $\mathcal{I}$ with concepts $\mathrm{Clrng}^r_{\mathfrak{t}}$, $\mathrm{Cld}^r_{\mathfrak{t}}$, and $\mathrm{Des}^{r,o}_{\mathfrak{t}}$, informing how many r-successor a given root has among (i) the clearing, (b) its children, and how many (c) for each nominal o, and how many of its descendants are r-successors of o. The decorated quasi-forest is:



These two "small tricks", obfuscated by various technical difficulties, are the core ideas behind our quasi-forest-satisfiability algorithm. We encourage the reader to check the full paper for details.

4. Hardness of Rooted Entailment in $\mathcal{ALC}$Self: An Overview and Tricks

We reduce from the NExpTime-complete $2^n \times 2^n$ torus tiling problem [12, Cor. 4.15]. For its instance $(\mathcal{D}, \bar{c})$, we construct a rooted CQ q and an $\mathcal{ALC}$Self-KB $\mathcal{K}$ such that $\mathcal{K} \not\models q$ iff $(\mathcal{D}, \bar{c})$ has a solution. Models of $\mathcal{K}$ represent $2^n \times 2^n$ tori (possibly incorrectly) labelled with tiles, while matches of q detect violations of tilings. We represent tori by binary trees of height $2n$, where every leaf determines a position (x, y) in $2^n \times 2^n$ and carries a selection of concepts $\mathrm{Ad}_1^{b_1}, \ldots, \mathrm{Ad}_n^{b_n}$ and $\mathrm{Ad}_{n+1}^{b_{n+1}}, \ldots, \mathrm{Ad}_{2n}^{b_{2n}}$, for bit-strings $b_1 \ldots b_n$ and $b_{n+1} \ldots b_{2n}$ encoding x and y. Unusually, we equip every leaf representing (x, y) with three extra children, encoding (x, y), $(x{+}1, y)$ and $(x, y{+}1)$ (counting modulo 2^n). This "localises" vertical and horizontal successors of each element. Call the resulting structures *treepods*.

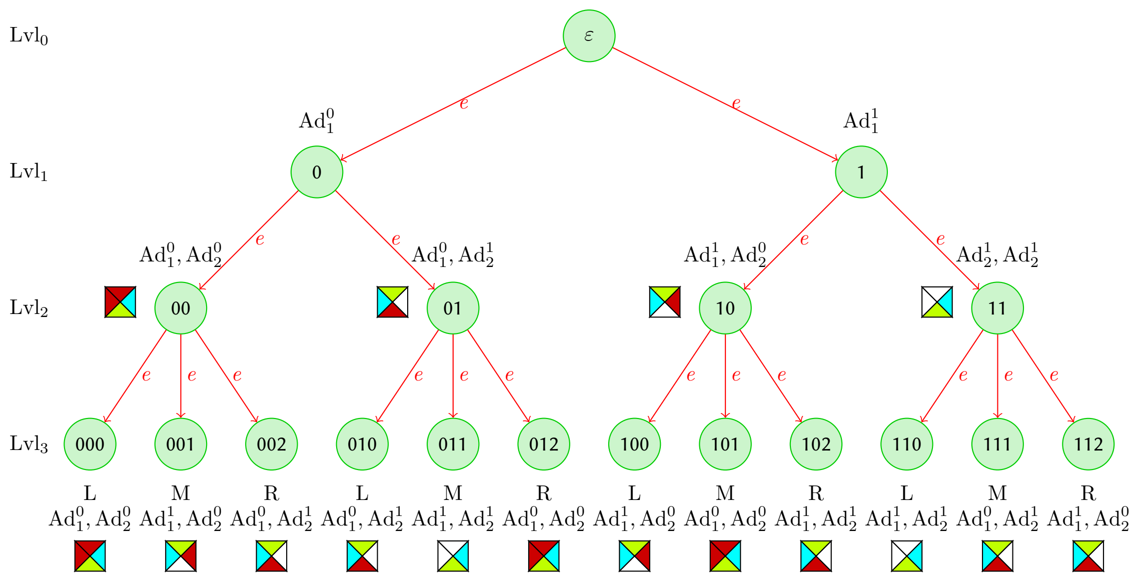


Figure 1: An example treepod for $n = 1$.

The verification of tiling relations in treepods can be implemented locally in $\mathcal{ALC}$, but only under a naïve assumption that all the nodes representing the same position of a torus carry precisely the same tile. Such a "tile equality test" will be implemented with the query. To make this feasible, we design an alternative way of encoding positions and tiles in treepods by means of *tentacles*, namely the outgoing paths endowed with various self-loops replacing the leaves of treepods. An example tentacle replacing the node 100 in the above treepod is depicted on Figure 2.

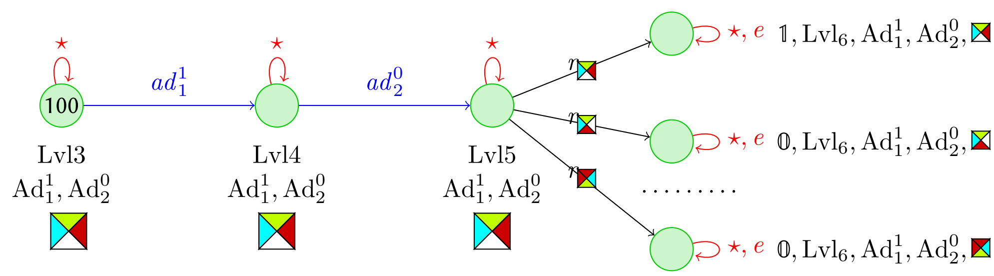


Figure 2: An example tentacle replacing the node 100 in the treepod from Figure 1.

By replacing each leaf of a treepod with a suitable tentacle, we obtain *jellyfishes*. Note that the node 100 that "stores its address" with concepts Ad_1^1, Ad_2^0, and $\boxtimes$, has now an outgoing $ad_1^1 ad_2^0 r_{\boxtimes} \mathbb{1}$?-path. Similarly, 012 has an outgoing $ad_1^0 ad_2^0 r_{\boxtimes} \mathbb{1}$? path. Our $\mathcal{ALC}$Self-KB $\mathcal{K}$ axiomatises jellyfishes, while our query q compares such path-based addresses of nodes. Self-loops are crucial here as they are used to simulate disjunction over paths.

Acknowledgements I thank my PhD supervisor Sebastian Rudolph for his constant support and extreme patience. This work was supported by the ERC Consolidator Grant No. 771779 (DeciGUT). The full version of this paper is available on ArXiV, accessible via: https://arxiv.org/abs/2406.07095

References

[1] D. Calvanese, T. Eiter, M. Ortiz, Regular Path Queries in Expressive Description Logics with Nominals, in: IJCAI, 2009.

[2] M. Bienvenu, D. Calvanese, M. Ortiz, M. Simkus, Nested Regular Path Queries in Description Logics, in: KR, 2014.

[3] D. Calvanese, M. Ortiz, M. Simkus, Verification of Evolving Graph-structured Data under Expressive Path Constraints, in: ICDT, 2016.

[4] M. Ortiz, Query Answering in Expressive Description Logics: Techniques and Complexity Results, Ph.D. thesis, TU Wien, AT, 2010.

[5] B. Bednarczyk, S. Rudolph, Worst-Case Optimal Querying of Very Expressive Description Logics with Path Expressions and Succinct Counting, in: IJCAI, 2019.

[6] B. Bednarczyk, E. Kieroński, Finite Entailment of Local Queries in the Z Family of Description Logics, in: AAAI, 2022.

[7] M. Y. Vardi, The Complexity of Relational Query Languages (Extended Abstract), in: STOC, 1982.

[8] I. Pratt-Hartmann, Data-Complexity of the Two-Variable Fragment with Counting Quantifiers, Information and Computation (2009).

[9] Y. Kazakov, RIQ and SROIQ are harder than SHOIQ, in: KR, 2008, pp. 274–284.

[10] J. C. Jung, C. Lutz, M. Martel, T. Schneider, Querying the Unary Negation Fragment with Regular Path Expressions, in: ICDT, 2018.

[11] B. Bednarczyk, Data Complexity in Expressive Description Logics with Path Expressions, in: IJCAI, 2024.

[12] C. Lutz, The Complexity of Description Logics with Concrete Domains, Ph.D. thesis, RWTH Aachen University, Germany, 2002.