Extending $\mathcal{ALCQIO}$ with Trees

Tomer Kotek, Mantas Šimkus, Helmut Veith and Florian Zuleger
TU Vienna
Vienna, Austria
Email: {kotek,veith,zuleger}@forsyte.at, simkus@dbai.tuwien.ac.at

***Abstract*—We study the description logic $\mathcal{ALCQIO}$, which extends the standard description logic $\mathcal{ALC}$ with nominals, inverses and counting quantifiers. $\mathcal{ALCQIO}$ is a fragment of first order logic and thus cannot define trees. We consider the satisfiability problem of $\mathcal{ALCQIO}$ over finite structures in which k relations are interpreted as forests of directed trees with unbounded outdegrees.**

We show that the finite satisfiability problem of $\mathcal{ALCQIO}$ with forests is polynomial-time reducible to finite satisfiability of $\mathcal{ALCQIO}$. As a consequence, we get that finite satisfiability is NEXPTIME-complete. Description logics with transitive closure constructors or fixed points have been studied before, but we give the first decidability result of the finite satisfiability problem for a description logic that contains nominals, inverse roles, and counting quantifiers and can define trees.

I. Introduction

Description Logics are a well established family of logics for Knowledge Representation and Reasoning [3]. They model the domain of interest in terms of *concepts* (classes of objects) and *roles* (binary relations between objects). These features make description logics very useful to formally describe and reason about graph-structured information. The usefulness of description logics is witnessed e.g. by the W3C choosing description logics to provide the logical foundations to the standard Web Ontology Language (OWL) [22]. Another application of description logics is formalization and static analysis of UML class diagrams and ER diagrams, which are basic modeling artifacts in object-oriented software development [4] and database design [2]. In these settings, standard reasoning services provided by description logics can be used to verify, e.g., the consistency of a diagram.

Description logics extended with various forms of reachability have been studied in the literature, though the focus has been mostly on arbitrary rather than finite structures. No extensions of $\mathcal{ALCQIO}$ with reachability or transitive closure are known to be decidable on finite structures. Finite satisfiability in $\mu\mathcal{ALCQIO}$, the extension of $\mathcal{ALCQIO}$ with fixed points, is known to be undecidable [6]. The description logic $\mu\mathcal{ALCQO}$ without nominals is EXPTIME-complete [5]. Close correspondences between description logics extended with fixpoints and variants of the μ-calculus have also been identified [11], [28]. The description logic $\mathcal{ALC}_{reg}$, which allows regular expressions on roles but does not have counting quantifiers, nominals or inverses, is decidable and is a variant of propositional dynamic logic (PDL), a logic for reasoning about program behavior (see [27]). Based on the results from [5], [8] proved the decidability of $\mathcal{ZOI}$, $\mathcal{ZOQ}$ and $\mathcal{ZIQ}$, logics extending $\mathcal{ALC}_{reg}$ with any combination of two constructs among counting quantifiers, nominals and inverses, as well as *self* and Boolean role operations, but left the case of all three constructs $\mathcal{ZOIQ}$ open. [12] proved decidability over arbitrary structures for a description logic extending $\mathcal{ALCQIO}$ which has both transitive closure and counting quantifiers, under syntactical restrictions which guarantee that the counting quantifiers and the transitive closure do not interact. Description logics with complex interactions of transitive roles (but no transitive closure) and counting quantifiers were studied e.g. in [15], [29].

A. Motivation

Our main motivation for studying the finite satisfiability problem of $\mathcal{ALCQIO}$ is to use $\mathcal{ALCQIO}$ as the underlying logic of a new type of analysis of programs with dynamic data structures. While model checking has been very successful for the verification of programs in the last few decades, programs with dynamic data structures still pose a considerable challenge. Since the memory of programs with dynamic data structures is essentially a directed finite graph of unbounded size, special decidable logics need to be tailored for the task. The verification community has devoted considerable attention to developing logics and analyses of the graph-theoretic structure of the data structures induced by the pointers in the memory (e.g. singly- or doubly-linked lists, or trees). However, the content of these data structures has been largely ignored[1]. This is at odds with the fact that the *goal* of having data structures in the memory is to stored data, manipulate it, and react to it. State of the art verification techniques are successful at handling low level errors which have to do with memory safety or the implementation of data structures. However, any analysis of programs which ignores the data in the data structures cannot be used to verify *functional correctness* of programs, i.e. to verify that programs operate correctly with respect to the requirements on the relationships between entities handled by the program.

This is an author's version of the article: T. Kotek, M. imkus, H. Veith and F. Zuleger, "Extending ALCQIO with Trees," 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2015), Kyoto, 2015, pp. 511-522. doi: 10.1109/LICS.2015.54 The final version of this paper is available at https://ieeexplore.ieee.org/document/7174908/.

[1]A special case which has received attention is arithmetic data.

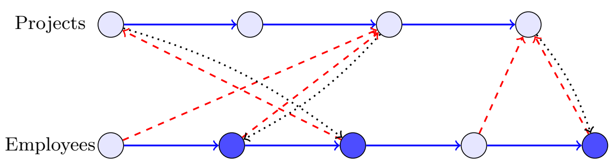


Fig. I.1. The employee list and the project list are built using *next* pointers (solid arrows). The *worksFor* and *managedBy* pointers (dashed respectively dotted arrows) relate the employees and projects.

Consider for example the information system of a company. This program stores data for employees, projects, departments and other entities, and relationships between them, such as which employee works for which project or who manages each project. High level requirements on the system may include e.g. that each employee works for at most one project, or that a project may only be managed by an employee who works for that project. Fig. I.1 depicts a partial memory snapshot where the employees and projects are stored in lists. The *next* pointer is a *shape pointer*, which is used to implement the data structure; the *worksFor* and *managedBy* pointers are *content pointers*, which encode data.

The example program uses the heap to store database-like information, i.e., information that is naturally understood in terms of entity-relationship diagrams. In such programs, the data structures play the role of an *in-memory database*. Since description logics have been used extensively to reason about entity-relationships and on databases, they are a natural choice of a family of logics on which to base an analysis of the heap. The difference between the pure database setting and the program analysis setting is the twofold use of pointers as shape pointers and as content pointers. While the description logics in the literature are good at handling content pointers, they are not well-suited to reason about structures such as lists where reachability is inherent. Understanding to which description logics trees can be added without major decidability issues is a necessary stepping stone toward achieving concrete powerful tools which use description logics in a verification setting. In [9] we have shown that a description logic closely related to $\mathcal{ALCQIO}$ can naturally represent the content of data structures in the above example. We developed there a verification method for content analysis based on a description logic. However, the description logic of [9] was not suitable to describe shape. Therefore we used a separation logic as a logic for shapes and the two-variable fragment of first order logic with counting and trees as a superlogic to tie the previous logics together. The extension of $\mathcal{ALCQIO}$ with trees in the current paper simplifies the rather cumbersome method of [9] considerably. Moreover, unlike [9], this extension allows to analyze programs with an *arbitrary* number of memory-sharing data structures.

The current paper is part of a larger effort by our group to adapt Description Logics for program analysis. We intend to use suitable description logics as assertion logics for new forms of program analysis. Since these program analyses focus on the content rather than the combinatorial structure of the data, we call this analysis *content analysis*. [9] describes this approach in depth.

B. Our contribution

We develop decision procedures for the satisfiability problem of $\mathcal{ALCQIO}$ terminologies over finite structures with built-in forests. That is, we study satisfiability over finite structures in which an *a priori* unbounded number of relations are guaranteed to be interpreted as forests. The number of children of any vertex in such a forest is unbounded. The forests can be further restricted using appropriate $\mathcal{ALCQIO}$ axioms to be trees, binary trees, or successor relations.

The main result of this paper are algorithms which decide the finite satisfiability problem of $\mathcal{ALCQIO}$ over finite structures with built-in forests. The algorithms are reductions to finite satisfiability in $\mathcal{ALCQIO}$ without built-in forests. The algorithms run in NEXPTIME, which is optimal since finite satisfiability of $\mathcal{ALCQIO}$ is already NEXPTIME-hard. We note that our complexity results hold even when the numbers in the counting quantifiers are encoded in binary. We extend our results to Boolean terminologies, or in other words, to the logic $\mathcal{ALCQIO}_b$ obtained by closing the set of $\mathcal{ALCQIO}$ axioms under the Boolean connectives. Note that $\mathcal{ALCQIO}$ does not have the tree model property which accounts for lower complexity in sublogics of $\mathcal{ALCQIO}$, see Corollary 5.21 and the ensuing discussion in [31].

A short abstract based on an earlier formulation of this result was presented at the Description Logic workshop 2014 [18].

C. The two-variable fragment and $\mathcal{ALCQIO}$

$\mathcal{ALCQIO}$ is contained in the extension of the two-variable fragment of first order logic with counting quantifiers C^2. $\mathcal{ALCQIO}$ and the two-variable fragment of first order logic FO^2 (without counting) are incomparable. $\mathcal{ALCQIO}$ contains $\mathcal{ALC}$, which is a syntactic variant of the multimodal logic K. All quantifiers in $\mathcal{ALCQIO}$ are guarded by binary relations symbols (atomic roles). To aid the reader not familiar with description logics, we give an alternative presentation of $\mathcal{ALCQIO}$ in Section II-C directly as a fragment of first order logic.

The satisfiability and finite satisfiability problems of the two-variable fragment FO^2 of first order logic and its extensions have received considerable attention. The finite satisfiability problem of the guarded two-variable fragment with counting but without constant symbols, which contains $\mathcal{ALCQI}$ but not $\mathcal{ALCQIO}$, was shown to be EXPTIME-complete in [25]. Satisfiability and finite satisfiability of C^2 formulae are NEXPTIME-complete [24] (see also [14], [23]). Finite satisfiability of C^2 with two forests was shown in [10] to be NEXPTIME-complete. Satisfiability and finite satisfiability of C^2 with a single equivalence relation was shown to be NEXPTIME-complete, while C^2 with two equivalence relations is undecidable [26]. Decidability of extensions of FO^2 with successor relations, order relations, and equivalence relations without counting were studied [13], [16], [17], [20], [21], [30].

Although our results are similar in spirit to the results of the breakthrough paper [10], the results in our paper and in [10] are incomparable:

(i) C^2 contains $\mathcal{ALCQIO}$.
(ii) [10] allows two forests, whereas we allow an unbounded number of forests. The decidability of C^2 with three successor relations is not known, while $\mathcal{ALCQIO}$ with three successor relations is covered by our results.
(iii) We allow the vertices in our forests to have unboundedly many children, while [10] deals only with forests with bounded out-degrees.
(iv) Our proof is compositional in the sense that we give a reduction to finite satisfiability of $\mathcal{ALCQIO}$, whereas [10] gives a direct proof of decidability.

II. The Formalism and Examples

A. The description logic $\mathcal{ALCQIO}$

From the point of view of finite model theory, $\mathcal{ALCQIO}$ is a syntactic variant of a fragment of first order logic. In description logic terminology, binary relation symbols are called *atomic roles* and unary relation symbols are called *atomic concepts*. Let $\mathsf{N_R}$ and $\mathsf{N_C}$ denote the countable infinite sets of atomic roles and atomic concepts respectively. Concepts and roles are built inductively using constructors. Atomic concepts and atomic roles are respectively concepts and roles. The various constructors available to build concept and roles determine the particular description logics, giving rise to a wide family of logics with varying expressivity, and decidability and complexity of reasoning. We have an additional countable infinite set of symbols $\mathsf{N_i}$ called *individuals*. The members of $\mathsf{N_i}$ will be used to construct special concepts called *nominals*.

The semantics of concepts and roles is given in terms of structures, where atomic concepts and atomic roles are interpreted as unary and binary relations in a structure, respectively.

Inclusion axioms play the role of the formulae of $\mathcal{ALCQIO}$, and *terminologies* play the role of theories of $\mathcal{ALCQIO}$ formulae. Their semantics is given again in terms of structures and is based on the semantics of concepts and roles. An inclusion axiom $C \sqsubseteq D$ asserts that the interpretation of the concept C is contained in the interpretation of the concept D. A terminology is a set of inclusion axioms. A terminology Φ asserts that all the inclusion axioms in Φ hold.

Definition 1 (Syntax of $\mathcal{ALCQIO}$)**.** *The set of* roles *and* concepts *of $\mathcal{ALCQIO}$ are defined inductively:*

- *Atomic concepts $A \in \mathsf{N_C}$ are concepts;*
- *Atomic roles $r \in \mathsf{N_R}$ are roles;*
- *Individuals $o \in \mathsf{N_i}$ are concepts (called* nominals*)* [2]
- *If r is a role, then r^- is a role;*
- *If r is a role, C, D are concepts and n is a positive integer, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\exists r.C$ and $\exists^{\leq n} r.C$ are concepts.*

[2]For simplicity, we deviate slightly from standard description logic notation here. For every $o \in \mathsf{N_i}$, we use the notation o rather than $\{o\}$ for the corresponding nominal.

For any concepts C, D, $C \sqsubseteq D$ is an inclusion axiom*. Any finite set of inclusion axioms is a* terminology.

We will denote terminologies by lowercase and uppercase greek letters.

A *structure* (or *interpretation*) is a tuple $\mathcal{M} = (M, \cdot)$, where M is a finite set (the *universe*), and $\cdot$ is an *interpretation function*, which assigns to each atomic concept $C \in \mathsf{N_C}$ a unary relation $C^{\mathcal{M}} \subseteq M$, to each atomic role $R \in \mathsf{N_R}$ a binary relation $R^{\mathcal{M}}$ over M, and to each nominal $o \in \mathsf{N_i}$ an unary relation $o^{\mathcal{M}} \subseteq M$ *of size* $|o^{\mathcal{M}}| = 1$.

In this paper, all structures are finite. Satisfiability and implication always refer to finite structures only.

Definition 2 (Semantics of $\mathcal{ALCQIO}$)**.** *The semantics of concepts, roles, inclusion axioms and terminologies in $\mathcal{ALCQIO}$ is given in terms of structures. The function $\cdot^{\mathcal{M}}$ is extended to the remaining concepts and roles inductively below.*

$$\begin{array}{lcl}
(C \sqcap D)^{\mathcal{M}} & = & C^{\mathcal{M}} \cap D^{\mathcal{M}} \\
(C \sqcup D)^{\mathcal{M}} & = & C^{\mathcal{M}} \cup D^{\mathcal{M}} \\
(\neg C)^{\mathcal{M}} & = & M \setminus C^{\mathcal{M}} \\
(r^-)^{\mathcal{M}} & = & \{(e, e') \mid (e', e) \in r^{\mathcal{M}})\} \\
(\exists r.C)^{\mathcal{M}} & = & \{e \mid \exists e' : (e, e') \in r^{\mathcal{M}},\ e' \in C^{\mathcal{M}}\} \\
(\exists^{\leq n} r.C)^{\mathcal{M}} & = & \{e \mid \exists^{\leq n} e' : (e, e') \in r^{\mathcal{M}},\ e' \in C^{\mathcal{M}}\}
\end{array}$$

We say $\mathcal{M}$ *satisfies* an inclusion axiom $C \sqsubseteq D$ and write $\mathcal{M} \models C \sqsubseteq D$ if $C^{\mathcal{M}} \subseteq D^{\mathcal{M}}$. We say $\mathcal{M}$ *satisfies* a terminology Φ and write $\mathcal{M} \models \Phi$ if $\mathcal{M} \models \varphi$ for every inclusion axiom $\varphi \in \Phi$. If $\mathcal{M} \models \Phi$, then $\mathcal{M}$ is a *model* of Φ. We say Φ_1 *implies* Φ_2 and write $\Phi_1 \models \Phi_2$ if every model of Φ_1 is also a model of Φ_2.

We will use the following abbreviations.

– $\top = C \sqcup \neg C$, where C is an arbitrary concept and $\bot = \neg\top$;
– $C \equiv D$ for the two inclusion axioms $C \sqsubseteq D$ and $D \sqsubseteq C$;
– $\exists^{=n} r.C$ for the concept $\exists^{\leq n} r.C \sqcap \neg\exists^{\leq n-1} r.C$;
– $\exists^{\geq n} r.C$ for the concept $\neg\exists^{\leq n-1} r.C$;
– $\mathrm{func}(f)$ for the inclusion axiom $\top \sqsubseteq \exists^{\leq 1} f.\top$.
– $\exists r$ for the concept $\exists r.\top$; similarly for $\exists^{\leq n} r$, $\exists^{\geq n} r$, $\exists^{=n} r$;

Note that $\top^{\mathcal{M}} = M$ and $\bot^{\mathcal{M}} = \emptyset$ for any structure $\mathcal{M}$ with universe M. For a terminology Φ, we denote by $|\Phi|$ the length of Φ as a string.

Throughout the paper will be use the notation $[k] = \{1, \ldots, k\}$, for a $k \in \mathbb{N}$, to denote the interval of numbers from 1 to k.

B. $\mathcal{ALCQIO}$ with forests

Definition 3 (Class $\mathbb{F}(s_1, \ldots, s_k)$)**.** *Let $s_1, \ldots, s_k$ be roles in* $\mathsf{N_R}$. *The class $\mathbb{F}(s_1, \ldots, s_k)$ is the class of finite structures $\mathcal{M}$ where, for each $i \in [k]$, the predicate s_i is interpreted as a forest (that is, a directed acyclic graph such that the indegree of each non-root node is 1).*

When k is clear from the context, we write $\mathbb{F}$ instead of $\mathbb{F}(s_1, \ldots, s_k)$.

We study the satisfiability problem of $\mathcal{ALCQIO}$ in the class $\mathbb{F}(s_1, \ldots, s_k)$, i.e. the problem of whether, for a given terminology Φ, there exists a structure $\mathcal{M} \in \mathbb{F}(s_1, \ldots, s_k)$ such that $\mathcal{M} \models \Phi$.

Example 1 (Trees). *Given a structure $\mathcal{M} \in \mathbb{F}(s_1, \ldots, s_k)$ with universe M, consider the graph $D_i^{\mathcal{M}} = \langle M, s_i^{\mathcal{M}} \rangle$. It is guaranteed that $s_i^{\mathcal{M}}$ is a forest. There is a $\mathcal{ALCQIO}$ terminology Φ_i^{tree} which axiomatizes that $D_i^{\mathcal{M}}$ is a tree. Φ_i^{tree} uses a nominal o_i for the root of $D_i^{\mathcal{M}}$:*

$$\Phi_i^{tree} = \{\exists s_i^- \equiv \neg o_i\}$$

Since an element $e \in M$ has an incoming s_i edge iff e is not a root in the forest $D_i^{\mathcal{M}}$, Φ_i^{tree} guarantees that $D_i^{\mathcal{M}}$ is a tree by guaranteeing that there is exactly one root in $D_i^{\mathcal{M}}$, namely $o_i^{\mathcal{M}}$.

Example 2 (Successor relations). *Given a structure $\mathcal{M} \in \mathbb{F}(s_1, \ldots, s_k)$ with universe M, consider the graph $D_i^{\mathcal{M}} = \langle M, s_i^{\mathcal{M}} \rangle$. A successor relation is a tree in which every vertex has outdegree at most 1. Hence, the following terminology axiomatizes that $D_i^{\mathcal{M}}$ is a successor relation:*

$$\Phi_i^{succ} = \{\exists s_i^- \equiv \neg o_i,\ \top \sqsubseteq \exists^{\leq 1} s_i\}$$

Example 3 (Binary trees and d-ary trees). *We show how to axiomatize a binary tree using s_i and new concepts L_i and R_i. We extend Φ_i^{tree} with the three axioms*

$$\begin{array}{rcl} \alpha_i^1 & = & L_i \sqcup R_i \equiv \neg o_i \\ \alpha_i^2 & = & L_i \sqcap R_i \equiv \bot \\ \alpha_i^3 & = & \top \sqsubseteq \left(\exists^{\leq 1} s_i.L_i\right) \sqcap \left(\exists^{\leq 1} s_i.R_i\right) \end{array}$$

Intuitively, $L_i^{\mathcal{M}}$ and $R_i^{\mathcal{M}}$ encode the elements of M which are left respectively right children in $D_i^{\mathcal{M}}$. α_i^1 and α_i^2 guarantee that $L_i^{\mathcal{M}}$ and $R_i^{\mathcal{M}}$ partition $M \backslash \{o_i^{\mathcal{M}}\}$. α_i^3 guarantees that every node has at most one child in $L_i^{\mathcal{M}}$ and at most one child in $R_i^{\mathcal{M}}$.

We can now use s_i, L_i, and R_i as a binary tree. For example, the inclusion axiom $\exists s_i.L_i \sqsubseteq \exists s_i.R_i$ expresses that every node that has a left child also has a right child.

The case of d-ary trees proceeds similarly using d new concepts which partition $M \backslash o_i^{\mathcal{M}}$.

Example 4 (Multiple successor relations). *Since each terminology Φ_i^{succ} guarantees that s_i is interpreted as a successor relation over the structures of $\mathbb{F}$, the terminology*

$$\psi = \bigcup_{i=1}^{k} \Phi_i^{succ}$$

guarantees simultaneously for all $i \in [k]$ that s_i is interpreted as a successor relation. Hence, we can axiomatize an a priori unbounded number of successor relations over the same universe. Similarly, we can axiomatize an unbounded number of trees or binary trees.

C. $\mathcal{ALCQIO}$ and first order logic

$\mathcal{ALCQIO}$ has a fairly standard reduction to the two-variable fragment of first order logic with counting C^2 (see e.g. [7]). In this section we give an equivalent definition of $\mathcal{ALCQIO}$ directly as a fragment $\mathcal{L}$ of C^2. This section appears for clarity of exposition only and is not required for the remainder of the paper.

It is convenient to abuse notation slightly in this section as follows. Unlike the convention in description logics that nominals are concepts whose interpretation is of size 1, which we used to define the semantics of $\mathcal{ALCQIO}$, in this section only we treat individuals $o \in \mathsf{N_i}$ as constant symbols. Let τ be the vocabulary $\tau = \mathsf{N_R} \cup \mathsf{N_C} \cup \mathsf{N_i}$, where each $r \in \mathsf{N_R}$ is a binary relation symbol, each $C \in \mathsf{N_C}$ is a unary relation symbol, and each $o \in \mathsf{N_i}$ is a constant symbol.

Definition 4 ($\mathcal{L}$). *First we define two sets of formulae $\mathcal{L}_z$, $z \in \{x, y\}$, such that $\mathcal{L}_z$ contains only formulae with one free variable z. Let $\bar{z} \in \{x, y\}$ such that $z \neq \bar{z}$.*

- *For every $C \in \mathsf{N_C}$, $C(z)$ belongs to $\mathcal{L}_z$;*
- *For every $o \in \mathsf{N_i}$, $z \approx o$ belongs to $\mathcal{L}_z$;*
- *For every $\varphi_z, \psi_z \in \mathcal{L}_z$, $\neg\varphi_z, \varphi_z \wedge \psi_z, \varphi_z \vee \psi_z$ belong to $\mathcal{L}_z$;*
- *For every $\varphi_{\bar{z}} \in \mathcal{L}_{\bar{z}}$, $r \in \mathsf{N_R}$, and $n \in \mathbb{N}$,*

$$\begin{array}{l} \exists \bar{z}.r(z, \bar{z}) \wedge \varphi_{\bar{z}},\ \exists^{\leq n} \bar{z}.r(z, \bar{z}) \wedge \varphi_{\bar{z}} \\ \exists \bar{z}.r(\bar{z}, z) \wedge \varphi_{\bar{z}},\ \exists^{\leq n} \bar{z}.r(\bar{z}, z) \wedge \varphi_{\bar{z}} \end{array}$$

  belong to $\mathcal{L}_z$.

$\mathcal{L}$ is the set of sentences $\forall x(\varphi_x \rightarrow \psi_x)$ where $\varphi_x, \psi_x \in \mathcal{L}_x$.

Note that for every $\varphi_z \in \mathcal{L}_z$, the formula obtained by switching between x and y in φ_z belongs to $\mathcal{L}_{\bar{z}}$.

Lemma 1. *There are functions $\mathfrak{T} : \mathcal{ALCQIO} \rightarrow \mathcal{L}$ and $\mathfrak{S} : \mathcal{L} \rightarrow \mathcal{ALCQIO}$ such that:*

(i) For every $\mathcal{ALCQIO}$ inclusion φ, φ and $\mathfrak{T}(\varphi)$ agree on the truth value of all τ-structures.

(ii) For every $\varphi \in \mathcal{L}$, φ and $\mathfrak{S}(\varphi)$ agree on the truth value of all τ-structures.

(iii) $\mathfrak{T}$ and $\mathfrak{S}$ are inverse functions.

Below we give the translation functions $\mathfrak{T}$ and $\mathfrak{S}$.

Definition 5 ($\mathfrak{T}$). *We define $\mathfrak{T} : \mathcal{ALCQIO} \rightarrow \mathcal{L}$ as follows: For $z \in \{x, y\}$, we set*

$$\begin{array}{lcl} \mathfrak{T}_z(C) & = & C(z) \\ \mathfrak{T}_z(o) & = & z \approx o \\ \mathfrak{T}_z(C \sqcap D) & = & \mathfrak{T}_z(C) \wedge \mathfrak{T}_z(D) \\ \mathfrak{T}_z(C \sqcup D) & = & \mathfrak{T}_z(C) \vee \mathfrak{T}_z(D) \\ \mathfrak{T}_z(\neg C) & = & \neg\mathfrak{T}_z(C) \\ \mathfrak{T}_z(\exists r.C) & = & \exists \bar{z}.r(z, \bar{z}) \wedge \mathfrak{T}_{\bar{z}}(C) \\ \mathfrak{T}_z(\exists r^-.C) & = & \exists \bar{z}.r(\bar{z}, z) \wedge \mathfrak{T}_{\bar{z}}(C) \\ \mathfrak{T}_z(\exists^{\leq n} r.C) & = & \exists^{\leq n} \bar{z}.r(z, \bar{z}) \wedge \mathfrak{T}_{\bar{z}}(C) \\ \mathfrak{T}_z(\exists^{\leq n} r^-.C) & = & \exists^{\leq n} \bar{z}.r(\bar{z}, z) \wedge \mathfrak{T}_{\bar{z}}(C) \end{array}$$

and define

$$\mathfrak{T}(C \sqsubseteq D) = \forall x.\mathfrak{T}_x(C) \rightarrow \mathfrak{T}_x(D).$$

Definition 6 ($\mathfrak{S}$). *We define $\mathfrak{S} : \mathcal{L} \rightarrow \mathcal{ALCQIO}$ as follows:*

For $z \in \{x, y\}$, we set

$$\begin{array}{lcl}
\mathfrak{S}(C(z)) & = & C \\
\mathfrak{S}(z \approx o) & = & o \\
\mathfrak{S}(\varphi_z \wedge \psi_z) & = & \mathfrak{S}(\varphi_z) \sqcap \mathfrak{S}(\psi_z) \\
\mathfrak{S}(\varphi_z \vee \psi_z) & = & \mathfrak{S}(\varphi_z) \sqcup \mathfrak{S}(\psi_z) \\
\mathfrak{S}(\neg\varphi_z) & = & \neg\mathfrak{S}(\varphi_z) \\
\mathfrak{S}(\exists \bar{z}.r(z, \bar{z}) \wedge \varphi_{\bar{z}}) & = & \exists r.\mathfrak{S}(\varphi_{\bar{z}}) \\
\mathfrak{S}(\exists^{\leq n} \bar{z}.r(z, \bar{z}) \wedge \varphi_{\bar{z}}) & = & \exists^{\leq n} r.\mathfrak{S}(\varphi_{\bar{z}}) \\
\mathfrak{S}(\exists \bar{z}.r(\bar{z}, z) \wedge \varphi_{\bar{z}}) & = & \exists r^{-}.\mathfrak{S}(\varphi_{\bar{z}}) \\
\mathfrak{S}(\exists^{\leq n} \bar{z}.r(\bar{z}, z) \wedge \varphi_{\bar{z}}) & = & \exists^{\leq n} r.\mathfrak{S}(\varphi_{\bar{z}})
\end{array}$$

and define

$$\mathfrak{S}(\forall x(\varphi_x \to \psi_x)) = \mathfrak{S}(\varphi_x) \sqsubseteq \mathfrak{S}(\psi_x).$$

III. Decision procedures for $\mathcal{ALCQIO}$ with forests

In this section we prove the existence a NEXPTIME decision procedure for finite satisfiability of $\mathcal{ALCQIO}$ terminologies with forests:

Theorem 1. *Finite satisfiability of $\mathcal{ALCQIO}$ terminologies over $\mathbb{F}(s_1, \ldots, s_k)$ is NEXPTIME-complete.*

We first outline the proof of Theorem 1. The proof proceeds by reduction from finite satisfiability of $\mathcal{ALCQIO}$ terminologies over $\mathbb{F}$ to finite satisfiability of $\mathcal{ALCQIO}$ terminologies over arbitrary finite structures.

Let Φ be a $\mathcal{ALCQIO}$ terminology for which we want to compute whether it is satisfiable by a structure in $\mathbb{F}$. We will compute a $\mathcal{ALCQIO}$ terminology Ψ such that Ψ is satisfiable by a finite structure iff Φ is satisfiable by a structure in $\mathbb{F}$. We call a finite model $\mathcal{M}$ of Φ *standard*, if $\mathcal{M} \in \mathbb{F}$, and *non-standard*, otherwise. A naive approach would be to set Ψ to be the union of Φ and a new terminology $\Psi_{\mathbb{F}}$ such that $\Psi_{\mathbb{F}}$ is satisfied by all the structures of $\mathbb{F}(s_1, \ldots, s_k)$, and is not satisfied by any other finite structure. However, being a forest is not expressible in $\mathcal{ALCQIO}$, so there is no such terminology $\Psi_{\mathbb{F}}$ in $\mathcal{ALCQIO}$. Instead, we will augment Φ to a terminology $\Psi = \Psi_\Phi$ such that (1) Φ and Ψ agree on all the structures of $\mathbb{F}$, and (2) whenever a non-standard model of Ψ exists, so does a standard model.

We use the following definition: A structure is *quasi-standard*, if every element of the universe has at most one incoming s_i-edge, for every $i \in [k]$. Quasi-standardness almost expresses that a relation s_i is a forest, except that elements might be reachable from a s_i-cycle instead of a s_i-root. We show that being quasi-standard is expressible in $\mathcal{ALCQIO}$.

Under certain conditions, it is possible to repeatedly apply an operation $\triangleright$, which turns non-standard but quasi-standard models into standard models, by eliminating the said cycles. The existence of a non-standard but quasi-standard model then implies the existence of a standard model. A sufficient condition under which quasi-standard models can be turned to standard models using $\triangleright$ is that the existence of so-called *useful labelings*. Useful labelings mimic an order relation on the types of the elements in the universe and guarantee that applying the operation $\triangleright$ makes progress towards a standard model. We show that having useful labelings is expressible in $\mathcal{ALCQIO}$.

As a result we get a decision procedure for satisfiability of Φ, which amounts to adding to Φ the requirements that models are quasi-standard and have useful labelings. The resulting $\mathcal{ALCQIO}$ terminology Ψ is finitely satisfiable iff Φ is satisfiable over $\mathbb{F}(s_1, \ldots, s_k)$. A decision procedure which is tight in terms of complexity is given in Section III-D. In Section III-C we give a simpler but complexity-wise suboptimal decision procedure. The decision procedure in Section III-D follows the same plan, and differs only in the construction and size of the terminology expressing the existence of useful labelings.

A. Types and the operation $\triangleright$

Let Φ be a terminology. We write $C \in \Phi$ if there exists an inclusion axiom $\varphi \in \Phi$ such that C appears in φ as a sub-concept.

Definition 7 ($\triangleright$)**.** *Let $r \in \mathsf{N_R}$, let $b_0, b_1 \in M$, and $\mathfrak{t} = (b_0, b_1, r)$. Let $\mathcal{M}_{\mathfrak{t}\triangleright}$ be the structure such that $\mathcal{M}$ and $\mathcal{M}_{\mathfrak{t}\triangleright}$ have the same universe M and the same interpretations of every atomic concept, nominal and atomic role except for r, and $r^{\mathcal{M}_{\mathfrak{t}\triangleright}} = r^{\mathcal{M}} \setminus \{(a, b_i) \mid (a, b_i) \in r \text{ and } i \in \{0, 1\}\} \cup \{(a, b_{1-i}) \mid (a, b_i) \in r \text{ and } i \in \{0, 1\}\}$.*

For the main property of $\triangleright$ we need the notion of types:

Definition 8 (Types)**.** *We define $\mathrm{TYPES}_\Phi = \boldsymbol{2}^{\{C \mid C \in \Phi\}}$ as the powerset over the set of concepts appearing in Φ. Let $\mathcal{M}$ be a structure $\mathcal{M}$ and $u \in M$. We denote by $\overline{tp}^{\Phi}_{\mathcal{M}}(u) \in \mathrm{TYPES}_\Phi$ the set of concepts $C \in \Phi$ such that $u \in C^{\mathcal{M}}$. We call $\overline{tp}^{\Phi}_{\mathcal{M}}(u)$ the* type *of u. We sometimes omit the subscript $\mathcal{M}$ when it is clear from the context.*

We note that the size of TYPES_Φ is at most $2^{|\Phi|}$.

Lemma 2. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two structures with the same universe M. If for all $u \in M$ we have $\overline{tp}^{\Phi}_{\mathcal{M}_1}(u) = \overline{tp}^{\Phi}_{\mathcal{M}_2}(u)$, then $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on Φ.*

Proof: Since Φ is a set of inclusion axioms, it suffices to show $\mathcal{M}_1 \models C \sqsubseteq D$ iff $\mathcal{M}_2 \models C \sqsubseteq D$ for all of the inclusion assertions $C \sqsubseteq D \in \Phi$. Let $C \sqsubseteq D$ be such an inclusion assertion. For $u \in M$, $u \in C^{\mathcal{M}_1}$ iff $u \in C^{\mathcal{M}_2}$, and $u \in D^{\mathcal{M}_1}$ iff $u \in D^{\mathcal{M}_2}$. Hence, $C^{\mathcal{M}_1} = C^{\mathcal{M}_2}$ and $D^{\mathcal{M}_1} = D^{\mathcal{M}_2}$, implying $\mathcal{M}_1 \models C \sqsubseteq D$ iff $\mathcal{M}_2 \models C \sqsubseteq D$. ∎

The crucial property of $\triangleright$ is that $\mathcal{M}$ and $\mathcal{M}_{\mathfrak{t}\triangleright}$ agree on Φ if b_0 and b_1 have the same type:

Lemma 3. *Let $r \in \mathsf{N_R}$, let $b_0, b_1 \in M$ such that $\overline{tp}^{\Phi}_{\mathcal{M}}(b_0) = \overline{tp}^{\Phi}_{\mathcal{M}}(b_1)$, and $\mathfrak{t} = (b_0, b_1, r)$.*
(1) $C^{\mathcal{M}} = C^{\mathcal{M}_{\mathfrak{t}\triangleright}}$ for all $C \in \Phi$.
Consequently:
(2) For every $u \in M$, $\overline{tp}^{\Phi}_{\mathcal{M}}(u) = \overline{tp}^{\Phi}_{\mathcal{M}_{\mathfrak{t}\triangleright}}(u)$.

Proof:

The proof of (1) proceeds by induction on the construction of the concepts, showing that $C^{\mathcal{M}} = C^{\mathcal{M}_{\mathfrak{t}\triangleright}}$ for all $C \in \Phi$. For ease of notation we write $\mathcal{M}_1 = \mathcal{M}$ and $\mathcal{M}_2 = \mathcal{M}_{\mathfrak{t}\triangleright}$.

1) If $A \in \mathsf{N_C}$, then $A^{\mathcal{M}_1} = A^{\mathcal{M}_2}$ since none of the atomic concepts change between $\mathcal{M}_1$ and $\mathcal{M}_2$.
2) If $o \in \mathsf{N_i}$, then similarly, there is no change.
3) If C_1 and C_2 are concepts satisfying the induction hypothesis, then $C_1 \sqcap C_2$, $C_1 \sqcup C_2$ and $\neg C_1$ also satisfy the claim, e.g., $(C_1 \sqcap C_2)^{\mathcal{M}_1} = C_1^{\mathcal{M}_1} \cap C_2^{\mathcal{M}_1} = C_1^{\mathcal{M}_2} \cap C_2^{\mathcal{M}_2} = (C_1 \sqcap C_2)^{\mathcal{M}_2}$.
4) For a role t, a concept C and a non-negative integer n, we consider the concepts $\exists t.C$, $\exists^{\leq n} t.C$, $\exists t^-.C$ and $\exists^{\leq n} t^-.C$:
   a) If $t \neq r$, then $\left(\exists t.C^{\mathcal{M}_1}\right) = \left(\exists t.C\right)^{\mathcal{M}_2}$ since $t^{\mathcal{M}_1} = t^{\mathcal{M}_2}$ and by induction $C^{\mathcal{M}_1} = C^{\mathcal{M}_2}$. Similarly, this holds for $\exists^{\leq n} t.C$, $\exists t^-.C$ and $\exists^{\leq n} t^-.C$.
   b) If $t = r$:

   The concepts $\exists \mathbf{r}^-.\mathbf{C}$ and $\exists^{\leq \mathbf{n}} \mathbf{r}^-.\mathbf{C}$: For every $u \in M$, $i = 1, 2$, we define

$$Q_i(u) = \left\{ v \mid (v, u) \in r^{\mathcal{M}_i} \text{ and } v \in C^{\mathcal{M}_i} \right\}$$

   We fix some $u \in M \setminus \{b_0, b_1\}$. We have $v \in Q_1(u)$ iff $v \in Q_2(u)$ using that $(v, u) \in (r)^{\mathcal{M}_1}$ iff $(v, u) \in (r)^{\mathcal{M}_2}$ and that by induction $C^{\mathcal{M}_1} = C^{\mathcal{M}_2}$. Thus, $Q_1(u) = Q_2(u)$. Therefore, $u \in (\exists r^-.C)^{\mathcal{M}_1}$ iff $u \in (\exists r^-.C)^{\mathcal{M}_2}$ and $u \in (\exists^{\leq n} r^-.C)^{\mathcal{M}_1}$ iff $u \in (\exists^{\leq n} r^-.C)^{\mathcal{M}_2}$.
   We now consider $u \in \{b_0, b_1\}$: We have $v \in Q_1(b_i)$ iff $v \in Q_2(b_{1-i})$ using that $(v, b_i) \in (r)^{\mathcal{M}_1}$ iff $(v, b_{1-i}) \in (r)^{\mathcal{M}_2}$ and that by induction $C^{\mathcal{M}_1} = C^{\mathcal{M}_2}$. Thus, $Q_1(b_i) = Q_2(b_{1-i})$ for $i \in \{0, 1\}$. Because of $\overline{tp}^{\Phi}_{\mathcal{M}_1}(b_0) = \overline{tp}^{\Phi}_{\mathcal{M}_1}(b_1)$ we get $b_i \in (\exists r^-.C)^{\mathcal{M}_1}$ iff $b_i \in (\exists r^-.C)^{\mathcal{M}_2}$ and $b_i \in (\exists^{\leq n} r^-.C)^{\mathcal{M}_1}$ iff $b_i \in (\exists^{\leq n} r^-.C)^{\mathcal{M}_2}$ for $i \in \{0, 1\}$.
   In summary, $(\exists r^-.C)^{\mathcal{M}_1} = (\exists r^-.C)^{\mathcal{M}_2}$ and $(\exists^{\leq n} r^-.C)^{\mathcal{M}_1} = (\exists^{\leq n} r^-.C)^{\mathcal{M}_2}$.

   The concepts $\exists \mathbf{r}.\mathbf{C}$ and $\exists^{\leq \mathbf{n}} \mathbf{r}.\mathbf{C}$: For every $u \in M$, $i = 1, 2$, we define

$$Q_i(u) = \left\{ v \mid (u, v) \in r^{\mathcal{M}_i} \text{ and } v \in C^{\mathcal{M}_i} \right\}$$

   We fix some $u \in M$. For every $v \notin \{b_0, b_1\}$ we have $v \in Q_1(u)$ iff $v \in Q_2(u)$, using that $(u, v) \in (r)^{\mathcal{M}_1}$ iff $(u, v) \in (r)^{\mathcal{M}_2}$ and that by induction $C^{\mathcal{M}_1} = C^{\mathcal{M}_2}$. We now consider $v \in \{b_0, b_1\}$: We have $b_i \in Q_1(u)$ iff $(u, b_i) \in r^{\mathcal{M}_1}$ and $b_i \in C^{\mathcal{M}_1}$ iff $(u, b_i) \in r^{\mathcal{M}_1}$ and $b_{1-i} \in C^{\mathcal{M}_1}$ (because $\overline{tp}^{\Phi}_{\mathcal{M}_1}(b_0) = \overline{tp}^{\Phi}_{\mathcal{M}_1}(b_1)$) iff $(u, b_{1-i}) \in r^{\mathcal{M}_2}$ and $b_{1-i} \in C^{\mathcal{M}_1}$ (by the definition of the operation $\rhd$) iff $(u, b_{1-i}) \in r^{\mathcal{M}_2}$ and $b_{1-i} \in C^{\mathcal{M}_2}$ (by induction assumption) iff $b_{1-i} \in Q_2(u)$. In summary, $|Q_1(u)| = |Q_2(u)|$ for all $u \in M$. Thus, $(\exists r.C)^{\mathcal{M}_1} = (\exists r.C)^{\mathcal{M}_2}$ and $\exists^{\leq n} r.C)^{\mathcal{M}_1} = u \in (\exists^{\leq n} r.C)^{\mathcal{M}_2}$.

As a direct conclusion from (1), we get (2). ■

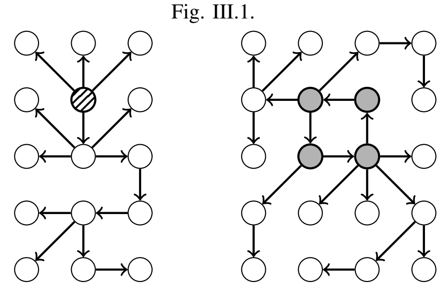
B. Quasi-standard structures and useful labelings

Here we define quasi-standard structures and useful labelings precisely and prove that they capture satisfiability in $\mathbb{F}$ (Lemma 5).

Definition 9 (i-Quasi-standard structure)**.** *Let $i \in [k]$ and let $\mathcal{M}$ be a structure with universe M. We define the directed graph $D_i^{\mathcal{M}} = \langle M, s_i^{\mathcal{M}} \rangle$. We say $\mathcal{M}$ is* i-quasi-standard *if every element $u \in M$ has at most one incoming edge in $D_i^{\mathcal{M}}$. Moreover, we define the set of* roots *$R_i^{\mathcal{M}} = \{u \in M \mid u$ has no incoming edge in $D_i^{\mathcal{M}}\}$.*

For an i-quasi-standard structure $\mathcal{M}$, $D_i^{\mathcal{M}}$ is the disjoint union of trees and *tree-like cycles*. A tree-like cycle is a directed graph which can be obtained from a directed cycle by attaching directed trees to the cycle's vertices by the trees' roots. Figure III.1 shows an example of $D_i^{\mathcal{M}}$ for an i-quasi-standard structure. It is the disjoint union of a tree and a tree-like cycle. The vertex filled with north east lines is the root of the tree. The gray vertices are the vertices of the unique cycle in the tree-like cycle. Whether a structure is i-quasi-standard is independent of the interpretation of any symbol other than s_i.

Fig. III.1.



Observe that if $\mathcal{M}$ is i-quasi-standard, then $D_i^{\mathcal{M}}$ is a forest iff $\mathcal{M}$ satisfies additionally that *every $u \in M$ is reachable from $R_i^{\mathcal{M}}$.*

Definition 10 (Quasi-standard structure)**.** *A structure $\mathcal{M}$ is* quasi-standard *if it is i-quasi-standard for every $i \in [k]$.*

The i-useful labelings we define next mimic linear orderings on the types of the elements in M that can be obtained from a Depth-First Search (DFS) run on $D_i^{\mathcal{M}}$ starting from elements in $R_i^{\mathcal{M}}$.

Definition 11 (Useful Labeling)**.** *Let $\mathcal{M}$ be a structure with universe M. Let $1 \leq i \leq k$. A function $f_i : M \to [|\mathrm{TYPES}_\Phi|]$ is an i-*useful labeling *for $\mathcal{M}$ if the following two conditions hold:*
(a) $f_i(u) = f_i(v)$ implies $\overline{tp}^{\Phi}_{\mathcal{M}}(u) = \overline{tp}^{\Phi}_{\mathcal{M}}(v)$ for all $u, v \in M$, and
(b) for every element $u \in M$, either $u \in R_i^{\mathcal{M}}$, or there exist elements $v, w \in M$ such that $f_i(u) = f_i(v)$, $f_i(w) < f_i(v)$ and the graph $D_i^{\mathcal{M}}$ has an edge (w, v).

Lemma 4. *Let $\mathcal{M} \in \mathbb{F}(s_1, \ldots, s_k)$ be a structure. Then, there are i-useful labelings f_i for $\mathcal{M}$, for every $i \in [k]$.*

Proof: $D_i^{\mathcal{M}}$ is a forest for every $i \in [k]$ because of $\mathcal{M} \in \mathbb{F}(s_1, \ldots, s_k)$. We fix some $i \in [k]$. We explain how to build an i-useful labeling for $\mathcal{M}$ by executing a Depth-First Search (DFS) from the elements in $R_i^{\mathcal{M}}$. If an element u is visited during the DFS, u is assigned a number according to its type

$\overline{tp}^{\Phi}_{\mathcal{M}}(u)$. If the type $\overline{tp}^{\Phi}_{\mathcal{M}}(u)$ has not appeared during the DFS yet, u is assigned the smallest number in $[|\mathrm{TYPES}_\Phi|]$ that has not been used so far; if the type has already appeared, u is assigned the number associated with this type. Let f_i be the labeling resulting from this process. We show that f_i is an i-useful labeling for $\mathcal{M}$.

By construction f_i is a function from M to $[|\mathrm{TYPES}_\Phi|]$. Moreover, for all $u, v \in M$ is holds that $f_i(u) = f_i(v)$ iff $\overline{tp}^{\Phi}_{\mathcal{M}}(u) = \overline{tp}^{\Phi}_{\mathcal{M}}(v)$ (*). It remains to show that for every element $u \in M$, either $u \in R_i^{\mathcal{M}}$, or there exist elements $v, w \in M$ such that $f_i(u) = f_i(v)$, $f_i(w) < f_i(v)$ and the graph $D_i^{\mathcal{M}}$ has an edge (w, v): Let $u \in M \setminus R_i^{\mathcal{M}}$. We proceed by a case distinction: (1) The type $\overline{tp}^{\Phi}_{\mathcal{M}}(u)$ of u has not been seen during the DFS before u is visited. Because of $u \notin R_i^{\mathcal{M}}$ there is a predecessor $w \in M$ of u in $D_i^{\mathcal{M}}$ through which u has been reached during the DFS. Because w has been reached before u and because u is assigned the smallest number in $[|\mathrm{TYPES}_\Phi|]$ that has not been used so far, we have $f_i(w) < f_i(u)$. (2) The type $\overline{tp}^{\Phi}_{\mathcal{M}}(u)$ of u has already been seen during the DFS before u is visited. Let $v \in M$ be the first node with type $\overline{tp}^{\Phi}_{\mathcal{M}}(v) = \overline{tp}^{\Phi}_{\mathcal{M}}(u)$ to be visited during the DFS. By case (1) there is a predecessor w of v in $D_i^{\mathcal{M}}$ with $f_i(w) < f_i(v)$. By (*) we have $f_i(u) = f_i(v)$. Thus, the claim follows. ■

This gives direction $\Rightarrow$ of the following lemma:

Lemma 5. *Let Φ be a terminology. Φ is satisfiable over $\mathbb{F}(s_1, \ldots, s_k)$ iff there is a quasi-standard structure $\mathcal{M}$ with $\mathcal{M} \models \Phi$ and there are i-useful labelings for every $i \in [k]$.*

Next, we introduce definitions that will be needed for the proof of direction $\Leftarrow$.

Let $D = (V, E)$ be a directed graph. $\mathit{Reach}_D(X) = Y$ denotes the set $Y \subseteq V$ of elements that are *reachable* from $X \subseteq V$ in D.

Definition 12 (Base and Values). *Let f be an i-useful labeling for $\mathcal{M}$. We call a set $X \subseteq M$ a* base *for $D_i^{\mathcal{M}}$ if $\mathit{Reach}_{D_i^{\mathcal{M}}}(X) = M$. We call a member x of a base X a* base element*. We define the* value

$$val_f(X) = \sum_{x \in X \setminus R_i^{\mathcal{M}}} f(x)$$

of a base X to be the sum over the label values of the base elements of X that are not in $R_i^{\mathcal{M}}$. We define the value

$$val_f(D_i^{\mathcal{M}}) = \min\{val_f(X) \mid X \text{ is a base for } D_i^{\mathcal{M}}\}$$

of the graph $D_i^{\mathcal{M}}$ to be the minimum of the values of its bases. We omit the subscript f in $val(X)$ and $val(D_i^{\mathcal{M}})$ when f is clear from the context.

Intuitively, values $val_f(D_i^{\mathcal{M}})$ capture how close the graph $D_i^{\mathcal{M}}$ is to being a forest:

Lemma 6. *Let f be an i-useful labeling for $\mathcal{M}$. $val_f(D_i^{\mathcal{M}}) = 0$ iff $D_i^{\mathcal{M}}$ is a forest.*

Proof: Assume $D_i^{\mathcal{M}}$ is a forest. Then $R_i^{\mathcal{M}}$ is a base for $D_i^{\mathcal{M}}$. Thus $val(R_i^{\mathcal{M}}) = 0$, which implies $val(D_i^{\mathcal{M}}) = 0$.

Conversely assume $val(D_i^{\mathcal{M}}) = 0$. Then there is a base X for $D_i^{\mathcal{M}}$ with $val(X) = 0$. Because f maps all nodes to positive values, we must have $X \subseteq R_i^{\mathcal{M}}$. Thus, every node in $D_i^{\mathcal{M}}$ is reachable from $R_i^{\mathcal{M}}$. ■

Lemma 7. *Let f be an i-useful labeling for $\mathcal{M}$. If X is a base, then $R_i^{\mathcal{M}} \subseteq X$.*

Proof: If v is a root, then it has no incoming edge in $D_i^{\mathcal{M}}$. Hence, for v to be reachable from X, v must belong to X. ■

The following lemma states a property of bases in quasi-standard structures:

Lemma 8. *Let $\mathcal{M}$ be a structure that is i-quasi-standard. Let f be an i-useful labeling for $\mathcal{M}$. Let X be a base for $D_i^{\mathcal{M}}$ with $val_f(X) = val_f(D_i^{\mathcal{M}})$. Then every base element $x \in X$ either belongs to $R_i^{\mathcal{M}}$ or to a cycle of $D_i^{\mathcal{M}}$.*

Proof: We fix some base element $x \in X$. Let us assume that x does not belong to $R_i^{\mathcal{M}}$. Let x_0 be the predecessor of x in $D_i^{\mathcal{M}}$. There exists $y \in X$ such that x_0 is reachable from y in $D_i^{\mathcal{M}}$. Hence, x is reachable from y. Assume $x \neq y$. Then, $X' = X \setminus \{x\}$ would be a base of $D_i^{\mathcal{M}}$ with $val(X') < val(X)$, contradiction. Thus, $x = y$ and x and x_0 are reachable from each other in $D_i^{\mathcal{M}}$, i.e. x belongs to a cycle. ■

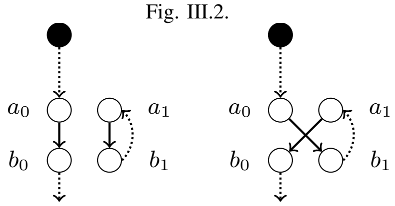
The next lemma, Lemma 9, shows that $\rhd$ can be applied to $D_i^{\mathcal{M}}$ for some $i \in [k]$ with $val_{f_i}(D_i^{\mathcal{M}}) > 0$ such that $val(D_i^{\mathcal{M}})$ decreases.

Lemma 9. *Let $\mathcal{M}$ be a quasi-standard structure with $\mathcal{M} \models \Phi$ such that there are ℓ-useful labelings for every $\ell \in [k]$. Let $i \in [k]$. If $val_{f_i}(D_i^{\mathcal{M}}) > 0$, then there is a tuple $\mathfrak{t} = (b_0, b_1, s_i)$ such that*

1) *For all $\ell \neq i$, $D_\ell^{\mathcal{M}} = D_\ell^{\mathcal{M}_{\mathfrak{t}\rhd}}$.*
2) *For all $u \in M$, $\overline{tp}^{\Phi}_{\mathcal{M}}(u) = \overline{tp}^{\Phi}_{\mathcal{M}_{\mathfrak{t}\rhd}}(u)$.*
3) *$\mathcal{M}_{\mathfrak{t}\rhd} \models \Phi$.*
4) *$val_{f_i}(D_i^{\mathcal{M}}) > val_{f_i}(D_i^{\mathcal{M}_{\mathfrak{t}\rhd}})$.*
5) *$val_{f_\ell}(D_\ell^{\mathcal{M}}) = val_{f_\ell}(D_\ell^{\mathcal{M}_{\mathfrak{t}\rhd}})$ for all $\ell \neq i$.*
6) *$\mathcal{M}_{\mathfrak{t}\rhd}$ is quasi-standard.*
7) *f_ℓ is an ℓ-useful labeling for $\mathcal{M}_{\mathfrak{t}\rhd}$ for all $\ell \in [k]$.*

First we give an intuition on the proof of Lemma 9. The full proof appears below. We fix some base X for $D_i^{\mathcal{M}}$ with $val(X) = val(D_i^{\mathcal{M}}) > 0$. We choose a base element $b_1 \in X \setminus R_i^{\mathcal{M}}$. Because f_i is an i-useful labeling for $\mathcal{M}$ there are $a_0, b_0 \in M$ such that $f_i(b_0) = f_i(b_1)$, $f_i(a_0) < f_i(b_0)$ and the graph $D_i^{\mathcal{M}}$ has an edge (a_0, b_0). By Lemma 8, b_1 belongs to a cycle in $D_i^{\mathcal{M}}$. Let a_1 denote the predecessor of b_1 on this cycle. a_0 and b_1 cannot belong to the same cycle in $D_i^{\mathcal{M}}$ by the minimality of X. Figure III.2 shows the result (II) of applying $\rhd$ on (I). The black vertex belongs to the base X, dotted arrows denote paths in $D_i^{\mathcal{M}}$, and solid arrows are edges in $D_i^{\mathcal{M}}$. Applying $\rhd$ increases the reachability of the structure: all vertices in (II) are now reachable from the black vertex. However, in the special case where the black vertex and b_1 coincide, a new cycle has been created. In both cases we have that $X' = X \setminus \{b_1\} \cup \{a_0\}$ is a base for $D_i^{\mathcal{M}_{\mathfrak{t}\rhd}}$ with $val(X') < val(X)$ and that $D_i^{\mathcal{M}_{\mathfrak{t}\rhd}}$ remains quasi-standard.

Fig. III.2.



(I) (II)

Proof of Lemma 9: We assume $val(D_i^{\mathcal{M}}) > 0$. Let X be a base for $D_i^{\mathcal{M}}$ such that $val(X) = val(D_i^{\mathcal{M}})$. Because of $val(D_i^{\mathcal{M}}) > 0$, there is a base element $b_1 \in X$ with $b_1 \notin R_i^{\mathcal{M}}$. By the i-usefulness of f_i, there are $a_0, b_0 \in M$ such that $f_i(b_0) = f_i(b_1)$, $f_i(a_0) < f_i(b_0)$ and the graph $D_i^{\mathcal{M}}$ has an edge (a_0, b_0). We set $\mathfrak{t} = (b_0, b_1, s_i)$. For all $\ell \in [k]$ we define the shorthand $D_\ell^{\triangleright} = D_\ell^{\mathcal{M}_{\mathfrak{t}\triangleright}}$.

By Lemma 8 b_1 belongs to some cycle C in $D_i^{\mathcal{M}}$. We denote the predecessor of b_1 in C by a_1. C contains a path π_{b_1,a_1} in $D_i^{\mathcal{M}}$ from b_1 to a_1 that does not go through (a_1, b_1). We note that π_{b_1,a_1} does not contain b_0; otherwise $X' = X \setminus \{b_1\} \cup \{a_0\}$ would be a basis for $D_i^{\mathcal{M}}$ with $val(D_i^{\mathcal{M}}) = val(X) > val(X')$, contradiction. Hence π_{b_1,a_1} does not go through (a_0, b_0). Consequently, π_{b_1,a_1} exists in $D_i^{\triangleright}$.

1) We have $D_\ell^{\mathcal{M}} = D_\ell^{\triangleright}$ for all $\ell \neq i$ because only the relation s_i is changed by the operation $\triangleright$ applied with $\mathfrak{t} = (b_0, b_1, s_i)$.
2) By Lemma 3, $\overline{tp}_{\mathcal{M}}^{\Phi}(u) = \overline{tp}_{\mathcal{M}_{\mathfrak{t}\triangleright}}^{\Phi}(u)$ for all $u \in M$.
3) By 2) and Lemma 2 $\mathcal{M}$ and $\mathcal{M}_{\mathfrak{t}\triangleright}$ agree on Φ.
4) We show that $X' = X \setminus \{b_1\} \cup \{a_0\}$ is a base for $D_i^{\triangleright}$. We have $val(X) > val(X')$ by $f_i(b_0) = f_i(b_1)$ and $f_i(a_0) < f_i(b_0)$. This is sufficient to establish $val(D_i^{\mathcal{M}}) = val(X) > val(X') \geq val(D_i^{\triangleright})$.
   We consider some node $v \in D_i^{\mathcal{M}}$. Because X is a basis, v is reachable in $D_i^{\mathcal{M}}$ from some $u \in X$ by some path π. We will prove that v is reachable in $D_i^{\triangleright}$ from some $u' \in X'$. We introduce $Z = \{a_0, b_0, a_1, b_1\}$ as a shorthand and proceed by a case distinction.
   Case 1: π does not contain any node from Z (in particular $u \neq b_1$). Then, π also witnesses that v is reachable from $u \in X'$ by π in $D_i^{\triangleright}$.
   Case 2: π contains a node from Z. Then there is a decomposition of π into two paths π_1 and π_2, i.e., $\pi = \pi_1\pi_2$, such that π_2 starts with a node $z \in Z$ but otherwise does not visit Z. We construct a path π_0 from a_0 to z using a suitable combination of the edge (a_0, b_1), the path π_{b_1,a_1} and the edge (a_1, b_0). Then the composition $\pi' = \pi_0\pi_2$ witnesses that v is reachable from $a_0 \in X'$ by π' in $D_i^{\triangleright}$.
5) Follows directly from 1) and 2).
6) Follows directly from 1) for every $\ell \in [k]$ with $i \neq \ell$. For $i = \ell$ this follows because by the definition of $\mathcal{M}_{\mathfrak{t}\triangleright}$ all nodes in $D_i^{\mathcal{M}}$ and $D_i^{\triangleright}$ have the same number of incoming s_i edges.
7) Follows directly from 1) and 2) for every $\ell \in [k]$ with $i \neq \ell$. Because f_i is an i-useful labeling for $D_i^{\mathcal{M}}$ we have that (a) $f_i(u) = f_i(v)$ implies $\overline{tp}_{\mathcal{M}}^{\Phi}(u) = \overline{tp}_{\mathcal{M}}^{\Phi}(v)$ for all $u, v \in M$ and (b) for every element $u \in M$, either $u \in R_i^{\mathcal{M}}$, or there exist elements $v, w \in M$ such that $f_i(u) = f_i(v)$, $f_i(w) < f_i(v)$ and the graph $D_i^{\mathcal{M}}$ has an edge (w, v). We have that $\overline{tp}_{\mathcal{M}}(u) = \overline{tp}_{\mathcal{M}_{\mathfrak{t}\triangleright}}^{\Phi}(u)$ for all $u \in M$ by 2). Because the operation $\triangleright$ changed only the edges (a_0, b_0) and (a_1, b_1) this almost shows that f_i is an i-useful labeling for $D_i^{\triangleright}$. It remains to argue that for every element u with $f_i(u) = f_i(b_0)$ $(= f_i(b_1))$ there exist elements $v, w \in M$ such that $f_i(u) = f_i(v)$, $f_i(w) < f_i(v)$ and the graph $D_i^{\triangleright}$ has an edge (w, v). This fact is witnessed by the edge (a_0, b_1).

■

Finally, we show that the repeated application of $\triangleright$ on a quasi-standard structure with useful labelings leads eventually to a structure satisfying Φ:

Proof of Lemma 5: By Lemma 4, $\mathcal{M} \in \mathbb{F}(s_1, \ldots, s_k)$ implies that there are i-useful labelings f_i for $\mathcal{M}$, for every $i \in [k]$.

For the other direction, there is a quasi-standard structure $\mathcal{M}$ with i-useful labelings for every $i \in [k]$. There is a sequence $\mathcal{M} = \mathcal{M}_1, \ldots, \mathcal{M}_p = \mathcal{M}'$ of structures such that each $\mathcal{M}_{j+1}$ is obtained from $\mathcal{M}_j$ by one application of $\triangleright$ and such that $val(D_i^{\mathcal{M}'}) = 0$ for all i. There is such a sequence because

1) the premise of Lemma 9 holds for $\mathcal{M}$,
2) for all j, if the premise of Lemma 9 holds for $\mathcal{M}_j$ and ϕ, then the premise of Lemma 9 holds for $\mathcal{M}_{j+1}$, and
3) the tuple $(val(D_1^{\mathcal{M}}), \cdots, val(D_k^{\mathcal{M}}))$ is decreasing with regard to the component-wise ordering of k-tuples over $\mathbb{N}$, so eventually $(0, \ldots, 0)$ must be reached.

By Lemma 6, $D_i^{\mathcal{M}'}$ is a forest, for all i. ■

C. *From satisfiability over $\mathbb{F}$ to plain satisfiability*

Here we show how to express the propert of being quasi-standard and the existence of useful labelings in $\mathcal{ALCQIO}$ and prove that $\mathcal{ALCQIO}$ over $\mathbb{F}$ is decidable. Expressing that a structure is quasi-standard is easy:

Lemma 10. *There exists a $\mathcal{ALCQIO}$-terminology quasi such that $\mathcal{M} \models$ quasi iff $\mathcal{M}$ is quasi-standard.*

Proof: Let

$$quasi = \{\mathrm{func}(s_i^-) : i \in [k]\}.$$

The axiom $\mathrm{func}(s_i^-)$ says that s_i^- is a partial function, i.e. that $|\{e' : (e', e) \in s_i^{\mathcal{M}}\}| \leq 1$ for every $e \in M$. Hence, the terminology $quasi$ defines the property of being quasi-standard.

■

Next we define a set of structures $ORD^{exp}(\Phi)$ that represent models of Φ and at the same time also contain useful labelings. After this definition we will show that $ORD^{exp}(\Phi)$ can be defined inside the logic $\mathcal{ALCQIO}$.

Definition 13. *Let Φ be a $\mathcal{ALCQIO}$ terminology. Let $q = |\mathrm{TYPES}_\Phi|$.*

Let $f_1, \dots, f_k, ord$ be fresh atomic roles, M be a fresh atomic concept, and $o_1, \dots, o_q$ be fresh nominals. Let $\mathcal{N}$ be a structure with universe N.

We denote the substructure of $\mathcal{N}$ with universe $M^{\mathcal{N}}$ by $\mathcal{M}$. We denote the set $N \setminus M^{\mathcal{N}}$ by $O^{\mathcal{N}}$. The structure $\mathcal{N}$ belongs to $ORD^{exp}(\Phi)$ if the following conditions hold:

1) *$\mathcal{M}$ satisfies Φ.*
2) *N is partitioned into $M^{\mathcal{N}}$ and $O^{\mathcal{N}} = \{o_1^{\mathcal{N}}, \dots, o_q^{\mathcal{N}}\}$.*
3) *We have that $(o_{j_1}^{\mathcal{N}}, o_{j_2}^{\mathcal{N}}) \in ord^{\mathcal{N}}$ iff $j_1 < j_2$.*
4) *$f_i^{\mathcal{N}}$ is a function from $M^{\mathcal{N}}$ to $O^{\mathcal{N}}$, for every $i \in [k]$.*
5) *$f_i^{\mathcal{N}}$ is an i-useful labeling for $\mathcal{M}$, using $O^{\mathcal{N}}$ for the natural numbers $[q]$ and ord for the order on the natural numbers in Definition 11, for every $i \in [k]$.*

Lemma 11. *$ORD^{exp}(\Phi)$ is non-empty iff there is a model $\mathcal{M}$ of Φ with i-useful labelings for $\mathcal{M}$ for every $i \in [k]$.*

Proof: Let $\mathcal{M}$ be a model of Φ with i-useful labelings f_i for every $i \in [k]$. We define a model $\mathcal{N}$ whose universe N is the disjoint union of the universe of $\mathcal{M}$ and $[q]$ by

- $M^{\mathcal{N}}$ is the universe of $\mathcal{M}$,
- $o_j^{\mathcal{N}} := j$,
- $ord^{\mathcal{N}} = \{(j_1, j_2) \mid 1 \le j_1 < j_2 \le q\}$,
- $f_i^{\mathcal{N}} := f_i$, and
- $C^{\mathcal{N}} = C^{\mathcal{M}}$ for all $C \in \Phi$.

Clearly, $\mathcal{N}$ satisfies properties 1,2, 3, 4 and 5 of Definition 13.

Let $\mathcal{N} \in ORD^{exp}(\Phi)$. Let $\mathcal{M}$ be the substructure of $\mathcal{N}$ with universe $M^{\mathcal{N}}$. By property 1, $\mathcal{M}$ satisfies Φ. By property 2, $O^{\mathcal{N}} = \{o_1^{\mathcal{N}}, \dots, o_q^{\mathcal{N}}\}$. By property 3, $(o_{j_1}^{\mathcal{N}}, o_{j_2}^{\mathcal{N}}) \in ord^{\mathcal{N}}$ iff $j_1 < j_2$. Thus $(O^{\mathcal{N}}, ord^{\mathcal{N}})$ is isomorphic to $([q], \le)$. By property 4, $f_i^{\mathcal{N}}$ is a function from $M^{\mathcal{N}}$ to $O^{\mathcal{N}}$, for every $i \in [k]$. By property 5, $f_i^{\mathcal{N}}$ is an i-useful labeling for $\mathcal{M}$, using $O^{\mathcal{N}}$ for the natural numbers $[q]$ and $ord^{\mathcal{N}}$ for the order on the natural numbers in Definition 11, for every $i \in [k]$. Because $(O^{\mathcal{N}}, ord^{\mathcal{N}})$ is isomorphic to $([q], \le)$, the last property implies that $f_i^{\mathcal{N}}$ is isomorphic to an i-useful labeling, for every $i \in [k]$. ∎

Lemma 12. *For every $\mathcal{ALCQIO}$ terminology Φ there exists a terminology $ext(\Phi)$ such that $ext(\Phi)$ defines $ORD^{exp}(\Phi)$.*

Proof: We set $ext(\Phi) = \Theta^1 \cup \Theta^2 \cup \Theta^3 \cup \Theta^4 \cup \Theta^5$. Θ^X defines the property X in Definition 13. Θ^5 is the union of Θ^{5a} and Θ^{5b} following (a) and (b) in Definition 11.

- For every atomic concept A, let $g(A) = A \sqcap M$. For every concept C, $g(C)$ is obtained by replacing its subconcepts with their g image and intersecting with M (e.g., $g(C_1 \sqcup C_2) = (g(C_1) \sqcup g(C_2)) \sqcap M$). Let $g(\Phi)$ be obtained from Φ by replacing every inclusion axiom $C \sqsubseteq D \in \Phi$ by $g(C) \sqsubseteq g(D)$. Let $\mathcal{M}$ be the substructure of $\mathcal{N}$ with universe $M^{\mathcal{N}}$ by $\mathcal{M}$. We have $\mathcal{M} \models \Phi$ iff $\mathcal{N} \models g(\Phi)$ (this holds because we have $C^{\mathcal{M}} = g(C)^{\mathcal{N}}$ for all $C \in \Phi$). Let $\Theta^1 = g(\Phi)$.
- Let
$$\Theta^2 = \{\neg M \equiv (o_1 \sqcup \dots \sqcup o_q)\}\,.$$
Θ^2 says that the universe of $N \setminus M^{\mathcal{N}} = \{o_1^{\mathcal{N}}, \dots, o_q^{\mathcal{N}}\} = O^{\mathcal{N}}$.
- Let Θ^3 be the terminology containing
$$o_{j_1} \sqsubseteq \exists ord.o_{j_2}, \text{ if } j_1 < j_2,$$
and
$$o_{j_1} \sqsubseteq \neg\exists ord.o_{j_2}, \text{ if } j_1 \ge j_2,$$
for $1 \le j_1, j_2 \le q$. Θ^3 says that
$$o_{j_1}^{\mathcal{N}} \in \{u \mid (u, o_{j_2}^{\mathcal{N}}) \in ord^{\mathcal{N}}\}, \text{ if } j_1 < j_2,$$
and
$$o_{j_1}^{\mathcal{N}} \notin \{u \mid (u, o_{j_2}^{\mathcal{N}}) \in ord^{\mathcal{N}}\}, \text{ if } j_1 \ge j_2,$$
i.e., $(o_{j_1}^{\mathcal{N}}, o_{j_2}^{\mathcal{N}}) \in ord^{\mathcal{N}}$ iff $j_1 < j_2$.
- Let Θ^4 be the terminology containing
$$\text{func}(f_i),\ (\exists f_i \equiv M)\,,\ \left(\exists f_i^- \sqsubseteq \neg M\right)$$
for $i \in [k]$. Θ^4 says that $f_i^{\mathcal{N}}$ is a function from $M^{\mathcal{N}}$ to $N \setminus M^{\mathcal{N}} = O^{\mathcal{N}}$.
- Let Θ^{5a} be the terminology containing
$$\left(\exists f_i^-.C\right) \sqcap \left(\exists f_i^-.\neg C\right) \equiv \bot$$
for all $i \in [k]$ and $C \in \Phi$. Θ^{5a} says that if $u, v \in M^{\mathcal{N}}$ point to the same nominal (i.e., $f_i^{\mathcal{N}}(u) = f_i^{\mathcal{N}}(v)$), then they must agree on every concept $C \in \Phi$ (i.e., $u \in C^{\mathcal{N}}$ iff $v \in C^{\mathcal{N}}$), thus $\overline{tp}_{\mathcal{M}}^{\Phi}(u) = \overline{tp}_{\mathcal{M}}^{\Phi}(v)$.
- Let Θ^{5b} be the terminology containing
$$o_\ell \sqcap \exists f_i^- \exists s_i^- \sqsubseteq \exists ord^-.\exists f_i^-.\exists s_i.\exists f_i.o_\ell$$
for all $1 \le \ell \le q$ and $i \in [k]$. Θ^{5b} says that if there is a non-root element $u \in M^{\mathcal{N}}$ pointing to some nominal $o_\ell^{\mathcal{N}}$ with $f_i^{\mathcal{N}}$, then there is $\ell' < \ell$ and $w \in M^{\mathcal{N}}$ pointing to $o_{\ell'}^{\mathcal{N}}$ with $f_i^{\mathcal{N}}$ that has an s_i-successor $v \in M^{\mathcal{N}}$ pointing to $o_\ell^{\mathcal{N}}$ with $f_i^{\mathcal{N}}$.

∎

Theorem 2. *Let Φ be a terminology. There is a $\mathcal{ALCQIO}$ terminology Ψ such that Φ is finitely satisfiable over $\mathbb{F}(s_1, \dots, s_k)$ iff Ψ is finitely satisfiable.*

Proof: The lemma follows from Lemmas 5, 10, 11, and 12 by setting $\Psi = ext(\Phi \cup quasi)$. ∎

D. A NEXPTIME decision procedure

The algorithm in Theorem 2 produces, for a terminology Φ, a terminology whose size is exponential in the size of Φ. Most of the constructions along the proof introduce only a polynomial growth, except for the nominals in Definition 13 and the concepts that use them. We discuss here how to effectively compute an $\mathcal{ALCQIO}$-terminology of polynomial size in Φ, which introduces the required linear ordering of exponential length without use of the nominals. Since satisfiability in $\mathcal{ALCQIO}$ is NEXPTIME-complete [19], [24], so is satisfiability over $\mathbb{F}$. We sketch the idea first.

In Section III-C a structure $\mathcal{N} \in ORD^{exp}(\Phi)$ with universe N represents a model $\mathcal{M}$ of Φ with universe M and at the same time also contains useful labelings for $\mathcal{M}$. Here, we define a set of structures $ORD^{poly}(\Phi)$ in a different though similar way. Let $y = |\{C \mid C \in \Phi\}|$. We introduce new concepts $P_1, \dots, P_y$ and use them to require that $O := N \backslash M$ is of size 2^y and that $succ$ is interpreted as a successor relation on O. We think of the reflexive-transitive closure of $succ^{\mathcal{N}}$ as $ord^{\mathcal{N}}$ from Definition 13, but we will not compute $ord^{\mathcal{N}}$ explicitly. For every binary word $b_y \cdots b_1$, there will be exactly one element of O in $\bigcap_{j:b_j=1} P_j^{\mathcal{N}} \cap \bigcap_{j:b_j=0} \neg P_j^{\mathcal{N}}$. I.e., $P_j^{\mathcal{M}}$ represents elements whose corresponding binary word has $b_j = 1$. $succ^{\mathcal{N}}$ will be induced by the usual successor relation on binary words of length y: an element $u \in O$ is the successor of $v \in O$ in $succ^{\mathcal{N}}$ iff there is ℓ such that (1) u and v agree on $P_j^{\mathcal{N}}$, for all $j > \ell$, (2) $u \in P_\ell^{\mathcal{N}}$ and $v \notin P_\ell^{\mathcal{N}}$ and (3) $v \in P_j^{\mathcal{N}}$ and $u \notin P_j^{\mathcal{N}}$, for all $j < \ell$. Similarly as in Definition 13, the functions $f_i^{\mathcal{N}}$ need to be useful labelings, using O for the numbers $[2^y]$ and $(succ^{\mathcal{N}})^*$ for the linear order on natural numbers in Definition 11. Importantly, we do not define the transitive closure $\left(succ^{\mathcal{N}}\right)^*$ explicitly. Instead, we exploit the fact that $b_y \cdots b_1$ is less than $d_y \cdots d_1$ iff there exists an index j such that $b_y \cdots b_{j+1} = d_y \cdots d_{j+1}$, $b_j = 0$ and $d_j = 1$.

Definition 14. *Let Φ be a $\mathcal{ALCQIO}$ terminology. Let $y = |\{C \mid C \in \Phi\}|$. Let $succ, f_1, \dots, f_k$ be fresh atomic roles, $M, P_1, \dots, P_y$, be fresh atomic concepts, o_{start} be a fresh nominal.*

Let $\mathcal{N}$ be a structure with universe N. We denote the substructure of $\mathcal{N}$ with universe $M^{\mathcal{N}}$ by $\mathcal{M}$. We denote the set $N \setminus M^{\mathcal{N}}$ by $O^{\mathcal{N}}$. We denote by $eval : O^{\mathcal{N}} \to [2^y]$ the function that maps an element $u \in O^{\mathcal{N}}$ to $eval(u) = 1 + \sum_{j:u \in P_j^{\mathcal{N}}} 2^{j-1}$. We denote by $\left(succ^{\mathcal{N}}\right)^*$ the reflexive-transitive closure of $succ^{\mathcal{N}}$. The structure $\mathcal{N}$ belongs to $ORD^{poly}(\Phi)$ if the following conditions hold:

1) *$\mathcal{M}$ satisfies Φ.*
2) *We have $O^{\mathcal{N}} = \{o_{start}^{\mathcal{N}}\} \cup \bigcup_{1 \le j \le y} P_j^{\mathcal{N}}$.*
3) *$eval$ is a bijective function, $eval(o_{start}^{\mathcal{N}}) = 1$, and $succ(u) = v$ iff $eval(u) + 1 = eval(v)$ for all $u, v \in O^{\mathcal{N}}$.*
4) *$f_i^{\mathcal{N}}$ is a function from $M^{\mathcal{N}}$ to $O^{\mathcal{N}}$, for every $i \in [k]$.*
5) *$f_i^{\mathcal{N}}$ is an i-useful labeling for $\mathcal{M}$, using $O^{\mathcal{N}}$ for the natural numbers $[2^y]$ and $\left(succ^{\mathcal{N}}\right)^*$ for the order on the natural numbers in Definition 11, for every $i \in [k]$.*

Lemma 13. *$ORD^{poly}(\Phi)$ is non-empty iff there is a model $\mathcal{M}$ of Φ with i-useful labelings for $\mathcal{M}$ for every $i \in [k]$.*

The proof of the above lemma is similar to the proof of Lemma 11.

Lemma 14. *For every $\mathcal{ALCQIO}$-terminology Φ there exists $\mathcal{ALCQIO}$-terminology $ext(\Phi)$, of size polynomial in y with $y = |\{C \mid C \in \Phi\}|$, such that $ext(\Phi)$ defines $ORD^{poly}(\Phi)$.*

Proof: We set $ext(\Phi) = \Theta^1 \cup \Theta^2 \cup \Theta^3 \cup \Theta^4 \cup \Theta^5$. Θ^X defines the property X in Definition 14. Θ^5 is the union of Θ^{5a} and Θ^{5b} following (a) and (b) in Definition 11.

For the sake of readability we introduce some additional fresh concepts of the form C_j, $C_{<j}$, $C_{>j}$, and $E_{i,j}$ beyond the statement of the theorem. These concepts can be either added as fresh concepts or used as abbreviations; the resulting terminology $ext(\Phi)$ will be of polynomial size in both cases.

The terminologies Θ^1, Θ^4 and Θ^{5a} are the same as in the proof of Lemma 12.

Let

$$\Theta^2 = \{\neg M \equiv o_{start} \sqcup \bigsqcup_{1 \le j \le y} P_j\}\,.$$

Θ^2 says that $O^{\mathcal{N}} = N \setminus M^{\mathcal{N}} = \{o_{start}^{\mathcal{N}}\} \cup \bigcup_{1 \le j \le y} P_j^{\mathcal{N}}$.

Let $\Theta^3 = \zeta_{consec} \cup \zeta_{func} \cup \zeta_{first} \cup \zeta_{last}$. ζ_{consec} axiomatizes that the successor relation mimics the binary words: two words $b_y \cdots b_1$ and $d_y \cdots d_1$ are consecutive in $succ$ iff there exists an index j such that $b_j \cdots b_1 = 01^{j-1}$, $d_j \cdots d_1 = 10^{j-1}$, and $b_y \cdots b_{j+1} = d_y \cdots d_{j+1}$. We introduce concepts of the form C_j, $C_{<j}$, $C_{>j}$:

$$\begin{aligned} C_j &\equiv \neg P_j \sqcap \exists succ.P_j \\ C_{<j} &\equiv \bigsqcap_{j_0 < j} (P_{j_0} \sqcap \exists succ.\neg P_{j_0}) \\ C_{>j} &\equiv \bigsqcap_{j < j_0 \le y} (P_{j_0} \sqcap \exists succ.P_{j_0} \sqcup \\ &\qquad \neg P_{j_0} \sqcap \exists succ.\neg P_{j_0}) \end{aligned}$$

$$\zeta_{consec} = \Big\{\neg M \sqcap \neg (P_1 \sqcap \cdots \sqcap P_y) \sqsubseteq \bigsqcup_{1 \le j \le y} C_{<j} \sqcap C_j \sqcap C_{>j}\Big\}$$

The terminologies

$$\begin{aligned} \zeta_{func} &= \{\mathrm{func}(succ)\} \\ \zeta_{first} &= \{o_{start} \equiv (\neg P_1 \sqcap \cdots \sqcap \neg P_y)\} \\ \zeta_{last} &= \{\exists succ \equiv \neg M \sqcap \neg (P_1 \sqcap \cdots \sqcap P_y)\} \end{aligned}$$

specify that $succ$ is a (partial) function, that $(\neg P_1 \sqcap \cdots \sqcap \neg P_y)^{\mathcal{N}}$ contains exactly the single element $o_{start}^{\mathcal{N}}$, and that exactly the elements in $O^{\mathcal{N}} \setminus (P_1 \sqcap \cdots \sqcap P_y)^{\mathcal{N}}$ have a successor. The above stated facts imply that for every binary word $b_y \cdots b_1 \in \{0,1\}^y$ there is exactly one element of $O^{\mathcal{N}}$ in $(\bigsqcap_{j:b_j=1} P_j \sqcap \bigsqcap_{j:b_j=0} \neg P_j)^{\mathcal{N}}$.

We do not define the transitive closure $\left(succ^{\mathcal{N}}\right)^*$ explicitly. Instead, we exploit the fact that $b_y \cdots b_1$ is less than $d_y \cdots d_1$ iff there exists an index j such that $b_y \cdots b_{j+1} = d_y \cdots d_{j+1}$, $b_j = 0$ and $d_j = 1$. We introduce concepts $E_{i,j}^{\mathcal{N}}$, for every $i \in [k]$ and $j \in [y]$, which will contain all of the elements $u \in M$ such that the types of u and $(s_i^-)^{\mathcal{N}}(u)$ agree on membership in $P_y^{\mathcal{N}}, \dots, P_{j+1}^{\mathcal{N}}$, $u \in P_j^{\mathcal{N}}$ and $(s_i^-)^{\mathcal{N}}(u) \notin P_j^{\mathcal{N}}$:

$$\begin{aligned} E_{i,j} \equiv\ & (\exists f_i.P_j) \sqcap \left(\exists s_i^-.\exists f_i.\neg P_j\right) \sqcap \\ & \bigsqcap_{j+1 \le j_0 \le y} \Big((\exists f_i.P_{j_0}) \sqcap \left(\exists s_i^-.\exists f_i.P_{j_0}\right) \sqcup \\ & \qquad (\exists f_i.\neg P_{j_0}) \sqcap \left(\exists s_i^-.\exists f_i.\neg P_{j_0}\right) \Big) \end{aligned}$$

In other words, $E_{i,j}^{\mathcal{N}}$ is the set of elements $u \in M$ such that the value of u is strictly larger than the value of $(s_i^-)^{\mathcal{N}}(u)$. Θ^{5b} consists of the axioms

$$\exists s_i^- \sqsubseteq \exists f_i.\exists f_i^-.\left(\bigsqcup_{j \in [y]} E_{i,j}\right)$$

for every $i \in [k]$. These axioms guarantee that every non root u element has an element v of the same type whose s_i-predecessor has a strictly smaller type. ■

Theorem 1. *(1) Let Φ be a terminology. There is a polynomial time computable $\mathcal{ALCQIO}$ terminology Ψ such that Φ is finitely satisfiable over $\mathbb{F}(s_1, \ldots, s_k)$ iff Ψ is finitely satisfiable.*
(2) Finite satisfiability of $\mathcal{ALCQIO}$ terminologies over $\mathbb{F}$ is NEXPTIME-complete.

Proof: (1) follows from Lemmas 5, 10, 13 and 14 by setting $\Psi = ext(\Phi \cup quasi)$.

Due to [24] and the fact that $\mathcal{ALCQIO}$ is a fragment of C^2, satisfiability of $\mathcal{ALCQIO}$ over arbitrary finite structures is in NEXPTIME. Due to (1) we get that finite satisfiability over $\mathbb{F}$ is also in NEXPTIME. Since finite satisfiability in $\mathcal{ALCQIO}$ is NEXPTIME-hard [19, Theorem 5], and finite satisfiability of $\mathcal{ALCQIO}$ is reducible to satisfiability over $\mathbb{F}$ (by setting $k = 0$) and we get (2). ■

Note that the complexity result holds even under binary encoding of numbers in counting quantifiers, since this is true for the upper bound in [24].

IV. Boolean terminologies

Theorem 1 extends to Boolean combinations of axioms. Let $\mathcal{ALCQIO}_b$ be the set of formulae obtained from $\mathcal{ALCQIO}$ by applying the connectives $\vee, \wedge, \neg$ any finite number of times. The semantics of $\mathcal{ALCQIO}$ extends naturally to $\mathcal{ALCQIO}_b$. Finite satisfiability in $\mathcal{ALCQIO}_b$ can be reduced to finite satisfiability in $\mathcal{ALCQIO}$: [3]

Lemma 15. *Let Φ be a $\mathcal{ALCQIO}_b$ formula. There exists a $\mathcal{ALCQIO}$ terminology Ψ such that Φ is satisfiable iff Ψ is satisfiable, and the size of Ψ is linear in the size of Φ. More precisely:*
1) *If $\mathcal{M}$ is a structure satisfying Φ, then there exists a structure $\mathcal{N}$ such that $\mathcal{N} \models \Psi$, $\mathcal{N}$ has the same universe as $\mathcal{M}$, and $\mathcal{N}$ and agrees with $\mathcal{M}$ on the interpretation of the symbols which occur in Φ.*
2) *If $\mathcal{N}$ is a structure satisfying Ψ, then $\mathcal{N}$ satisfies Φ.*

Proof: We prove the claim by induction on the construction of formulae in $\mathcal{ALCQIO}_b$. The claim we prove is slightly augmented as follows:
- We assume without loss of generality that Φ is given in negation normal form (NNF).
- We may assume without loss of generality that if Φ is satisfiable, then it is satisfiable by a structure of size strictly larger than 1.

Base.
– If $\Phi = C \sqsubseteq D$, then $C \sqsubseteq D$ is an $\mathcal{ALCQIO}$ inclusion.
– If $\Phi = \neg(C \sqsubseteq D)$, then let o be a fresh nominal, and let $\Psi = \{o \sqsubseteq C,\ D \sqsubseteq \neg o\}$. Ψ is a $\mathcal{ALCQIO}$ terminology.

[3]Note that the Scott normal form used in [24] for C^2 can be used to reduce $\mathcal{ALCQIO}$ terminologies to a conjunction of C^2 formulae with quantifier depth 2 in prenex normal form, but those conjuncts are not equivalent to $\mathcal{ALCQIO}$ axioms.

Closure. Let $\Phi_1, \Phi_2 \in \mathcal{ALCQIO}_b$ in NNF. Let Ψ_1, Ψ_2 be the $\mathcal{ALCQIO}$ terminologies guaranteed for Φ_1, Φ_2.
1) $\Phi = \Phi_1 \wedge \Phi_2$: Let $\Psi = \Psi_1 \cup \Psi_2$.
2) $\Phi = \Phi_1 \vee \Phi_2$. Let r be a fresh role and o_1, o_2, o_X, o_Y be fresh nominals. Let Ψ_{prep} be the terminology
$$\{(o_X \sqsubseteq \neg o_Y),\ (o_1 \sqsubseteq \neg o_2),\ (o_X \sqcup o_Y \equiv o_1 \sqcup o_2),\\ (\exists r.o_X \equiv \top),\ (\exists r.o_Y \equiv \bot)\}\,.$$
For a structure $\mathcal{M}$ with universe M, $\mathcal{M} \models \Psi_{prep}$ iff
a) $o_X^{\mathcal{M}} \neq o_Y^{\mathcal{M}}$
b) $o_1^{\mathcal{M}} \neq o_2^{\mathcal{M}}$
c) Either $o_1^{\mathcal{M}} = o_X^{\mathcal{M}}$ and $o_2^{\mathcal{M}} = o_Y^{\mathcal{M}}$,
or $o_1^{\mathcal{M}} = o_Y^{\mathcal{M}}$ and $o_2^{\mathcal{M}} = o_X^{\mathcal{M}}$.
d) Either $(\exists r.o_1)^{\mathcal{M}} = M$ and $(\exists r.o_2)^{\mathcal{M}} = \emptyset$,
or $(\exists r.o_1)^{\mathcal{M}} = \emptyset$ and $(\exists r.o_2)^{\mathcal{M}} = M$.

For $i \in \{1, 2\}$, let Θ_i be the terminology obtained from Ψ_i by replacing every axiom $C \sqsubseteq D$ with $C \sqcap \exists r.o_i \sqsubseteq D \sqcap \exists r.o_i$. Let $\Psi = \Psi_{prep} \cup \Theta_1 \cup \Theta_2$. The desired property follows directly from the claim:

Claim 1. *Let $\mathcal{N}$ be a structure such that $\mathcal{N} \models \Psi_{prep}$.*
a) If $(\exists r.o_1)^{\mathcal{N}} = M$, then $\mathcal{N} \models \Psi$ iff $\mathcal{N} \models \Psi_1$.
b) If $(\exists r.o_1)^{\mathcal{N}} = \emptyset$, then $\mathcal{N} \models \Psi$ iff $\mathcal{N} \models \Psi_2$.

Proof:
a) Assume $(\exists r.o_1)^{\mathcal{N}} = M$. Then $(\exists r.o_2)^{\mathcal{N}} = \emptyset$. For every axiom $C \sqsubseteq D$ in Ψ_1, $(C \sqcap \exists r.o_1)^{\mathcal{N}} = C^{\mathcal{N}} \cap M = C^{\mathcal{N}}$ and $(D \sqcap \exists r.o_1)^{\mathcal{N}} = D^{\mathcal{N}} \cap M = D^{\mathcal{N}}$. Hence, $\mathcal{N} \models C \sqsubseteq D$ iff $\mathcal{N} \models C \sqcap \exists r.o_1 \sqsubseteq D \sqcap \exists r.o_1$. Thus, $\mathcal{M} \models \Psi_1$ iff $\mathcal{N} \models \Theta_1$.
For every axiom $C \sqsubseteq D$ in Ψ_2, $(C \sqcap \exists r.o_2)^{\mathcal{N}} = C^{\mathcal{N}} \cap \emptyset = \emptyset$ and $(D \sqcap \exists r.o_2)^{\mathcal{N}} = D^{\mathcal{N}} \cap \emptyset = \emptyset$. Hence, $\mathcal{N} \models C \sqcap \exists r.o_2 \sqsubseteq D \sqcap \exists r.o_2$. Since Θ_2 is a negation free Boolean combination of axioms, $\mathcal{N} \models \Theta_2$.
b) Assume $(\exists r.o_1)^{\mathcal{N}} = \emptyset$. Then $(\exists r.o_2)^{\mathcal{N}} = M$. This case is symmetric to the previous case.

■

From Lemma 15 and Theorem 1 we get:

Theorem 3. *Finite satisfiability of $\mathcal{ALCQIO}_b$ formulae over $\mathbb{F}(s_1, \ldots, s_k)$ is NEXPTIME-complete.*

Using the above we can characterize the complexity of finite implication of $\mathcal{ALCQIO}$ terminologies over $\mathbb{F}$. Let Φ_1 and Φ_2 be $\mathcal{ALCQIO}$ terminologies. The $\mathcal{ALCQIO}_b$ formula
$$\Gamma_{imp} = \bigwedge_{\alpha \in \Phi_1} \alpha \wedge \bigvee_{\alpha \in \Phi_2} \neg\alpha$$
is satisfied by a structure $\mathcal{M}$ iff $\mathcal{M} \models \Phi_1$ and $\mathcal{M} \not\models \Phi_2$. Hence, Γ_{imp} is not satisfiable over $\mathbb{F}$ iff Φ_1 implies Φ_2 over $\mathbb{F}$. This gives the coNEXPTIME upper bound for the implication problem. For the lower bound note that the satisfiability problem can be reduced to the implication problem: Φ is satisfiable iff Φ does not imply the unsatisfiable terminology $\{o \sqsubseteq \neg o\}$.

Theorem 4. *Finite implication of $\mathcal{ALCQIO}_b$ formulae over $\mathbb{F}(s_1, \ldots, s_k)$ is coNEXPTIME-complete.*

V. CONCLUSION

The main result of this paper is an algorithm for the finite satisfiability problem of the description logic $\mathcal{ALCQIO}$ with an arbitrary number of relations which are guaranteed to be interpreted as forests. $\mathcal{ALCQIO}$ with built-in forests is well-suited for analysis of content of dynamic data structures in programs [9]: forests and inverses are used for axiomatizing data structures such as lists and binary trees, nominals are used for program variables, counting quantifiers are used to obtain the functionality of pointers, and Boolean terminologies are needed for writing verification conditions. The weakest precondition from [9], used to reason over program behaviour, applies to $\mathcal{ALCQIO}$ with a small change eliminating Boolean operations on roles (which can be done using the fact that pointers are functions). A related verification problem of whether constraints on graph databases expressed in description logics are preserved under data evolution was studied in [1].

Since the memory of a program is finite, we were mainly interested in the finite satisfiability problem. The proofs in this paper do not immediately extend to the satisfiability problem over arbitrary (i.e. not necessary finite) structures. Technically, this is because it is no longer guaranteed that any quasi-standard model can be transformed into a standard model in Lemma 5 by a *finite* number of applications of the operation $\triangleright$. This is an interesting phenomenon, since it is often the case that the satisfiability problem of logics is easier to approach and has a more elegant solution than the finite satisfiability problem.

Content analysis requires expressive decidable logics whose structures contain both binary relations which are restricted to be tree-like and binary relations which are arbitrary. The logics should be able to express rather strong properties of the tree-like relations such as reachability. At the same time, since the structures contain binary relations which might not be tree-like, the expressivity of these logics on the whole structure should be restricted in order to ensure decidability. We believe such logics may have other applications, e.g. for modeling networks. Developing such logics the result of the current paper is an important open problem.

ACKNOWLEDGMENT

Tomer Kotek, Helmut Veith and Florian Zuleger were supported by the Austrian National Research Network S11403-N23 (RiSE) of the Austrian Science Fund (FWF) and by the Vienna Science and Technology Fund (WWTF) through grants PROSEED and ICT12-059. Mantas Šimkus was supported by the FWF project P25518 and the WWTF project ICT12-15.

REFERENCES

[1] Shqiponja Ahmetaj, Diego Calvanese, Magdalena Ortiz, and Mantas Šimkus. Managing change in graph-structured data using description logics. In *Proc. of AAAI 2014*, pages 966–973. AAAI Press, 2014.

[2] Alessandro Artale, Diego Calvanese, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. Reasoning over extended ER models. In *Proc. of ER 2007*, volume 4801, pages 277–292, 2007.

[3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.

[4] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. *Artif. Intell.*, 168(1-2):70–118, 2005.

[5] Piero A. Bonatti, Carsten Lutz, Aniello Murano, and Moshe Y. Vardi. The complexity of enriched μ-calculi. In *Proc. of ICALP 2006*, volume 4052 of *LNCS*, pages 540–551. Springer, 2006.

[6] Piero A. Bonatti and Adriano Peron. On the undecidability of logics with converse, nominals, recursion and counting. *Artificial Intelligence*, 158(1):75–96, 2004.

[7] Alex Borgida. On the relative expressiveness of description logics and predicate logics. *Artif. Intell.*, 82(1-2):353–367, 1996.

[8] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *Proc. of IJCAI 2009*, pages 714–720, 2009.

[9] Diego Calvanese, Tomer Kotek, Mantas Šimkus, Helmut Veith, and Florian Zuleger. Shape and content - A database-theoretic perspective on the analysis of data structures. In *Proc. of iFM 2014*, pages 3–17, 2014.

[10] Witold Charatonik and Piotr Witkowski. Two-variable logic with counting and trees. In *Proc. of LICS 2013*, pages 73–82, 2013.

[11] Giuseppe De Giacomo and Maurizio Lenzerini. Concept language with number restrictions and fixpoints, and its relationship with mu-calculus. In *Proc. of ECAI 1994*, pages 411–415, 1994.

[12] Chan Le Duc, Myriam Lamolle, and Olivier Curé. A decision procedure for SHOIQ with transitive closure of roles. In *Proc. of ISWC 2013*, pages 264–279, 2013.

[13] Diego Figueira. Satisfiability for two-variable logic with two successor relations on finite linear orders. *arXiv preprint arXiv:1204.2495*, 2012.

[14] Erich Grädel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *Proc. of LICS 1997*, pages 306–317. IEEE, 1997.

[15] Yevgeny Kazakov, Ulrike Sattler, and Evgeny Zolin. How many legs do I have? Non-simple roles in number restrictions revisited. In *Proc. of LPAR 2007*, pages 303–317. Springer, 2007.

[16] Emanuel Kieronski. Decidability issues for two-variable logics with several linear orders. In *Proc. of CSL 2011*, pages 337–351. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[17] Emanuel Kieronski and Lidia Tendera. On finite satisfiability of two-variable first-order logic with equivalence relations. In *Proc. of LICS 2009*, pages 123–132. IEEE, 2009.

[18] Tomer Kotek, Mantas Simkus, Helmut Veith, and Florian Zuleger. Towards a description logic for program analysis: Extending ALCQIO with reachability. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.*, pages 591–594, 2014.

[19] Carsten Lutz. An improved NExpTime-hardness result for description logic ALC extended with inverse roles, nominals, and counting. LTCS-Report 04-07, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Dresden, Germany, 2005. Available online at http://www.cs.man.ac.uk/ ezolin/dl/bib/Improved_NExpTime_hardness_for_ALCFIO_(LT 07).ps.zip.

[20] Amaldev Manuel. Two variables and two successors. In *Proc. of MFCS 2010*, pages 513–524. Springer, 2010.

[21] Martin Otto. Two variable first-order logic over ordered domains. *The Journal of Symbolic Logic*, 66(02):685–702, 2001.

[22] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27.10.09. Available at http://www.w3.org/TR/owl2-overview/.

[23] Leszek Pacholski, Wieslaw Szwast, and Lidia Tendera. Complexity of two-variable logic with counting. In *Proc. of LICS 1997*, pages 318–327. IEEE, 1997.

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

[25] Ian Pratt-Hartmann. Complexity of the guarded two-variable fragment with counting quantifiers. *Journal of Logic Computation*, 17:133–155, 2007.

[26] Ian Pratt-Hartmann. Logics with counting and equivalence. In *Proc. of CSL-LICS 2014*, pages 76:1–76:10. ACM, 2014.

[27] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI 1991*, pages 466–471, 1991.

[28] Klaus Schild. Terminological cycles and the propositional μ-calculus. In *Proc. of KR 1994*, pages 509–520. Morgan Kaufmann, 1994.

[29] Lutz Schröder and Dirk Pattinson. How many toes do I have? Parthood and number restrictions in description logics. In *Proc. of KR 2008*, pages 307–317, 2008.

[30] Thomas Schwentick and Thomas Zeume. Two-variable logic with two order relations. *Logical Methods in Computer Science*, 8(1), 2012.

[31] Stephan Tobias. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.