Constructive Satisfiability Procedure for $\mathcal{ALC}^P(\mathcal{Z})$ (Preliminary Report)

Nadia Labai[1,2], Martin Homola[2], and Magdalena Ortiz[1]

[1] Institute of Information Systems, TU Wien, Favoritenstrasse 9-11, A-1040 Vienna, Austria
[2] Comenius University in Bratislava, Mlynská dolina, 84248 Bratislava, Slovakia

Abstract. The newly introduced $\mathcal{ALC}^P(\mathcal{D})$ extends $\mathcal{ALC}$ with path constraints over concrete domains $\mathcal{D}$. Previous work showed decidability of the satisfiability problem in $\mathcal{ALC}^P(\mathcal{D})$ for various concrete domains, including integer domains. However, the proof of this general result did not provide a complexity upper bound on the problem, or an explicit algorithm for model construction, which is necessary for some reasoning tasks such as abduction. In this paper, we initiate an investigation of the complexity of the satisfiability and model construction problems in $\mathcal{ALC}^P(\mathcal{Z})$, where $\mathcal{Z} = \langle \mathbb{Z}, =, < \rangle$. As a preliminary result towards establishing complexity upper bounds, we present a procedure for deciding satisfiability in a constructive way.

Keywords: Description logics, concrete domains, satisfiability

1 Introduction

Reasoning in description logics (DL) with the addition of data from concrete domains is a highly relevant research problem motivated by the practical need to include data values and express constraints on top of them in ontologies. Different extensions of DLs with concrete domains were investigated [2,11,12,9,7], and based on this research concrete domains were also incorporated into OWL [5].

The language $\mathcal{ALC}^P(\mathcal{D})$, recently introduced by Carapelle and Turhan [4], takes further steps in this direction. It extends individuals with multiple registers holding values from the universe of a domain structure $\mathcal{D}$, and enables the expression of rich path-constraints that relate, using the relations of $\mathcal{D}$, multiple registers from distinct individuals connected by a certain path. Carapelle and Turhan [4] provided a general decidability result for the satisfiability problem which holds for any negation-closed domain structure satisfying the so-called EHD (existence of a homomorphism is definable) property. Among such domain structures are, e.g., the natural numbers and integers augmented with equality, linear, semi-linear, and even lexicographic orders. However, these results do not establish a complexity upper bound for the problem. As the proof relies largely on reductions which do not offer an obviously constructive way to verify satisfiability, it does not give rise to an algorithm for constructing or enumerating models. However, such algorithms are necessary for certain reasoning tasks such as query answering [14,13,10] and ABox abduction [6,8,15].

In this paper, we take initial steps towards a constructive satisfiability procedure for $\mathcal{ALC}^P(\mathcal{Z})$, where $\mathcal{Z} = \langle \mathbb{Z}, =, < \rangle$ is the structure induced on the integers by the equality and linear order relations. Our satisfiability procedure is split into two parts. Firstly, $\mathcal{ALC}$ satisfiability is checked for an abstraction of the TBox and the target concept, similarly to Carapelle and Turhan [4]. Our main contribution is in the second step, where rather than relying on the generic decidability of the existence of a homomorphism necessary for satisfiability, we construct a representation of the numeric part of the model in the form of a so-called integer graph. Our results open the door for future research on reasoning tasks for $\mathcal{ALC}^P(\mathcal{Z})$ that require model enumeration, as well as complexity analysis and practicable algorithms.

The paper is structured as follows. After providing necessary background in Section 2, we introduce integer graphs in Section 3, and discuss their relationship to models of $\mathcal{ALC}^P(\mathcal{Z})$ concepts. In Sections 4 and 5 we present procedures for constructing integer graphs and extracting a satisfying interpretation from them. Finally, in Section 6, we discuss the results and future research. Due to space limitations, the proofs are sketched or omitted.

2 Preliminaries for $\mathcal{ALC}^P(\mathcal{Z})$

The $\mathcal{ALC}^P(\mathcal{D})$ description logics were defined by Carapelle and Turhan [4] in general form. We present definitions specialized to $\mathcal{Z} = \langle \mathbb{Z}, =, < \rangle$. For $k \in \mathbb{Z}^+$, denote $[k] = \{1, \ldots, k\}$. A *constraint* $c(x_1, \ldots, x_k)$ of arity k is a Boolean combination of atomic constraints $\theta(x_i, x_j)$, where $\theta \in \{=, <\}$ and $i, j \in [k]$. We write $\mathcal{Z} \models c(a_1, \ldots, a_k)$ if the constraint is satisfied in $\mathcal{Z}$ by the assignment $x_i \mapsto a_i$.

Example 1. Throughout the paper we return to the same examples to demonstrate concepts and definitions. We underline symbols introduced in the examples to distinguish them from the rest of the paper.

Let $\underline{\theta}_1(y, z) := y < z$ and $\underline{\theta}_2(x, z) := z < x$. The constraint $\underline{\theta}_{1,2}(x, y, z) := \underline{\theta}_1(y, z) \wedge \underline{\theta}_2(x, z)$ has arity 3, and we have $\mathcal{Z} \models \underline{\theta}_{1,2}(3, 1, 2)$.

Let N_C, N_R, and Reg be countably infinite sets of *concept names*, *role names*, and *register names*, respectively. A sequence $P = r_1 \cdots r_n$, where $r_i \in N_R$ for $i \in [n]$, is a *role path* of length n. $\mathcal{ALC}^P(\mathcal{Z})$ concepts C are inductively defined by:

$$C, D := A \mid \neg C \mid (C \sqcap D) \mid \exists r.C \mid \exists P.c(\mathrm{S}^{i_1} x_1, \ldots, \mathrm{S}^{i_k} x_k)$$

where $A \in N_C$, $r \in N_R$, P is a role path of length $n \geq 0$, c is a constraint of arity k, $x_1, \ldots, x_k \in \mathsf{Reg}$, and $0 \leq i_1, \ldots, i_k \leq n$. We call $\exists P.c(\mathrm{S}^{i_1} x_1, \ldots, \mathrm{S}^{i_k} x_k)$ a *path constraint*. The symbol S appearing in the path constraint stands for *successor*, as the term $\mathrm{S}^i x$ points to the register variable x of the i-th element on the path P.

A $\mathcal{Z}$*-interpretation* $\mathcal{I}$ is a tuple $(\Delta^\mathcal{I}, \cdot^\mathcal{I}, \gamma)$, where $\Delta^\mathcal{I}$ is the domain, $\cdot^\mathcal{I}$ is the interpretation function, and $\gamma : \Delta^\mathcal{I} \times \mathsf{Reg} \to \mathbb{Z}$ is the *valuation function*, assigning an integer value to each register variable of each element in the domain. The interpretation function maps each concept name $A \in N_C$ to some $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and each role name $r \in N_R$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The interpretation function is then extended to $\neg C, C \sqcap D, \exists r.C$ as usual.[3] Let $P = r_1, \ldots, r_n$ be a role path of length n.

[3] Refer, e.g., to the DL Handbook [1].

Then $P^{\mathcal{I}}$ is defined by: $P^{\mathcal{I}} = \{(v_0, \ldots, v_n) \in (\Delta^{\mathcal{I}})^{n+1} \mid (v_{i-1}, v_i) \in r_i^{\mathcal{I}} \text{ for } i \in [n]\}$, and $(\exists P.c(\mathrm{S}^{i_1}x_1, \ldots, \mathrm{S}^{i_k}x_k))^{\mathcal{I}}$ is defined by:

$$\{v \in \Delta^{\mathcal{I}} \mid \exists (v_0, \ldots, v_n) \in P^{\mathcal{I}} \text{ s.t. } v_0 = v, \text{ and } \mathcal{Z} \models c(\gamma(v_{i_1}, x_1), \ldots, \gamma(v_{i_k}, x_k))\}$$

We explain the interpretation of $\exists P.c(\mathrm{S}^{i_1}x_1, \ldots, \mathrm{S}^{i_k}x_k)$ further. If an element $v \in \Delta^{\mathcal{I}}$ is in $(\exists P.c(\mathrm{S}^{i_1}x_1, \ldots, \mathrm{S}^{i_k}x_k))^{\mathcal{I}}$, then there are $v_1, \ldots, v_n \in \Delta^{\mathcal{I}}$ such that there is a path starting at v which matches the pattern P, that is $(v, v_1, \ldots, v_n) \in P^{\mathcal{I}}$, and such that for the assignment $\gamma(v_{j_i}, x_j) \mapsto a_j$, we have $\mathcal{Z} \models c(a_1, \ldots, a_k)$.

Example 2. Let $\underline{C} := \exists \underline{rst}. \left(\underline{\theta}_{1,2}(x, y, z) \wedge \underline{\theta}_{3,4}(\mathrm{S}^1 x, \mathrm{S}^2 x, \mathrm{S}^3 x)\right)$ where:

$$\begin{aligned}
\underline{\theta}_{3,4}(\mathrm{S}^1 x, \mathrm{S}^2 x, \mathrm{S}^3 x) &:= \underline{\theta}_3(\mathrm{S}^1 x, \mathrm{S}^2 x) \wedge \underline{\theta}_4(\mathrm{S}^2 x, \mathrm{S}^3 x) \\
\underline{\theta}_3(\mathrm{S}^1 x, \mathrm{S}^2 x) &:= \mathrm{S}^1 x < \mathrm{S}^2 x \\
\underline{\theta}_4(\mathrm{S}^2 x, \mathrm{S}^3 x) &:= \mathrm{S}^2 x = \mathrm{S}^3 x
\end{aligned}$$

The interpretation of $\underline{C}$ would contain elements $\underline{v}_0$ from which a path $\underline{rst}$ originates, meaning there are elements $\underline{v}_i$, $i \in [3]$ for which $(\underline{v}_0, \underline{v}_1) \in r^{\mathcal{I}}$, $(\underline{v}_1, \underline{v}_2) \in s^{\mathcal{I}}$, and $(\underline{v}_2, \underline{v}_3) \in t^{\mathcal{I}}$ and such that 1. the x register of $\underline{v}_0$ holds a larger value than its z register, which holds a larger value than its y register, that is $\gamma(\underline{v}_0, y) < \gamma(\underline{v}_0, z) < \gamma(\underline{v}_0, x)$, and 2. the x register of $\underline{v}_1$ holds a smaller value than the x register of $\underline{v}_2$, which is equal to the value in the x register of $\underline{v}_3$. In other words, $\gamma(\underline{v}_1, x) < \gamma(\underline{v}_2, x) = \gamma(\underline{v}_3, x)$.

As usual, $C \sqcup D := \neg(\neg C \sqcap \neg D)$ and $\forall R.C = \neg \exists R.\neg C$, $\top := A \sqcup \neg A$ are defined as syntactic sugar. In addition, also define $\forall P.c(\mathrm{S}^{i_1}x_1, \ldots, \mathrm{S}^{i_k}x_k) := \neg \exists P.\neg c(\mathrm{S}^{i_1}x_1, \ldots, \mathrm{S}^{i_k}x_k)$ and $\exists P.C := \exists r_1.\exists r_2.\ldots.\exists r_n.C$. A *General Concept Inclusion (GCI)* is an expression $C \sqsubseteq D$, where C and D are concepts. A *TBox* is a finite set of GCIs. For a concept C and TBox $\mathcal{T}$, denote by $\mathsf{Reg}_{C,\mathcal{T}}$ the set of register names appearing in C and $\mathcal{T}$. Models and satisfiability are also defined as usual: let $\mathcal{T}$ be a TBox and let C be a concept. A $\mathcal{Z}$-interpretation $\mathcal{I}$ *models* $\mathcal{T}$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every GCI $C \sqsubseteq D$ in $\mathcal{T}$. We denote $\mathcal{I} \models \mathcal{T}$ in this case. C is *satisfiable w.r.t.* $\mathcal{T}$ iff there is a $\mathcal{Z}$-interpretation $\mathcal{I}$ which models $\mathcal{T}$ and $C^{\mathcal{I}} \neq \emptyset$. We denote $\mathcal{I} \models_{\mathcal{T}} C$ in this case.

We say $\mathcal{I}$ is *tree-shaped* if $\Delta^{\mathcal{I}} \subseteq \Sigma^{\star}$ is (isomorphic to) a prefix-closed set of strings over some finite alphabet Σ, and if additionally, for $u, v \in \Delta^{\mathcal{I}}$, we have $u \to v$ if and only if $v = u\sigma$ for some $\sigma \in \Sigma$, where $\to = \bigcup_{r \in N_R} r^{\mathcal{I}}$. The tree model property is a well-known result for $\mathcal{ALC}$, [1], which also propagates to $\mathcal{ALC}^P(\mathcal{Z})$:

Theorem 1 ([4]). *Let C be a concept and $\mathcal{T}$ a TBox in $\mathcal{ALC}^P(\mathcal{Z})$. C is satisfiable w.r.t. $\mathcal{T}$ iff there exists a tree-shaped $\mathcal{Z}$-interpretation $\mathcal{I}$ such that $\mathcal{I} \models_{\mathcal{T}} C$.*

Relying further on the treatment in [4], we also assume that all concepts are given in *constraint normal form*, where negations only appear before concept names, and all path-constraints are either of the form $\exists P.c$ where c is a conjunction of atomic constraints, or of the form $\forall P.c$ where c is a disjunction of atomic constraints.

Fix a concept C and a TBox $\mathcal{T}$ in constraint normal form. For an atomic constraint $\theta(\mathrm{S}^i x, \mathrm{S}^j y)$, we call $d = \max\{i, j\}$ the *depth* of θ, and for a constraint c, the depth is the maximal depth of the atomic constraints in c. Denote by Θ the set of atomic constraints occurring in C and $\mathcal{T}$. Let $\mathbf{B} = \{B_1, \ldots, B_{|\Theta|}\}$ be a set of fresh concept names, called *placeholders*, which do not appear in C or $\mathcal{T}$.

We can now define *abstracted concepts* and *constraint graphs*, which allow us to split the satisfiability check into testing usual $\mathcal{ALC}$-satisfiability of the abstracted TBox, and verifying the constraints in the appropriate constraint graph can be satisfied.

Definition 2 (Abstracted concepts). *Let $P = r_1 \dots r_n$ and let $\theta_1, \dots, \theta_m \in \Theta$ have depths $d_1 \leq \dots \leq d_m$, respectively. Define $d_0 = 0$ and $d_{m+1} = n$. For a conjunction c of $\theta_1, \dots, \theta_m$, the* abstraction *of $\exists P.c(\mathsf{S}^{i_1} x_1, \dots, \mathsf{S}^{i_k} x_k)$ is defined as:*

$$\exists P_1.(B_1 \sqcap \exists P_2.(B_2 \sqcap \cdots \exists P_m.(B_m \sqcap \exists P_{m+1}.\top) \cdots))$$

where $\exists P_i$ is short for $\exists r_{d_{i-1}+1} \dots \exists r_{d_i}$. If $d_i = d_{i+1}$ then $\exists P_{i+1}$ is empty. For a disjunction c' of $\theta_1, \dots, \theta_m$, the abstraction of $\forall P.c'(\mathsf{S}^{i_1} x_1, \dots, \mathsf{S}^{i_k} x_k)$ is defined as:

$$\forall P_1.(B_1 \sqcup \forall P_2.(B_2 \sqcup \cdots \forall P_m.(B_m \sqcup \forall P_{m+1}.\bot) \cdots)).$$

Define C_a and $\mathcal{T}_a$ to be the $\mathcal{ALC}$ concept and TBox, respectively, obtained by replacing every path constraint with its abstraction.

Example 3. The abstraction of $\underline{C}$ is given as follows. We have $\underline{n} = 3$, $\underline{m} = 4$, $\underline{d}_0 = \underline{d}_1 = \underline{d}_2 = 0$, $\underline{d}_3 = 2$, and $\underline{d}_4 = \underline{d}_5 = 3$. We have that $\exists \underline{P}_1$, $\exists \underline{P}_2$, and $\exists \underline{P}_5$ are empty, and $\exists \underline{P}_3 = \exists \underline{r} \exists \underline{s}$, and $\exists \underline{P}_4 = \exists \underline{t}$. All in all, we have:

$$\underline{C}_a := \underline{B}_1 \sqcap \left(\underline{B}_2 \sqcap \left(\exists \underline{r} \exists \underline{s}.\underline{B}_3 \sqcap \left(\exists \underline{t}.\underline{B}_4 \sqcap (\top)\right)\right)\right)$$

An ordinary interpretation $\underline{\mathcal{I}}$ that models $\underline{C}_a$ is shown in Figure 1, where a^i is shorthand for i repetitions of a. Note that elements in the interpretation of a path-constraint are at the origins of the paths, whereas in the abstraction, the placeholders are satisfied by elements along the path, with the d-th element satisfying the placeholders for constraints of depth d.

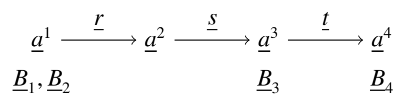


Fig. 1: An ordinary model of $\underline{C}_a$.

For a graph G, we denote by $V(G)$ its set of vertices and by $E(G)$ its set of edges.

Definition 3 (Constraint graph). *Given an ordinary tree-shaped interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, the* constraint graph *of $\mathcal{I}$ is the edge-labeled graph $\mathcal{G}_{\mathcal{I}} = (G, \mu)$ where $V(G) = \Delta^{\mathcal{I}} \times \mathsf{Reg}_{C,\mathcal{T}}$, and $\mu : E(G) \to \{l_=, l_<\}$, and $E(G)$ are as follows. For every $(u, x), (v, y) \in V(G)$ and $\theta \in \{=, <\}$, we have $((v, y), (u, x)) \in E(G)$ and $\mu((v, y), (u, x)) = l_\theta$ iff 1. u is a prefix of v, and 2. there exists $B_j \in \mathbf{B}$ such that $\theta_j = \theta(\mathsf{S}^{|v|-|u|} y, x)$ or $\theta_j = \theta(x, \mathsf{S}^{|v|-|u|} y)$, and $v \in B_j^{\mathcal{I}}$. Let $\bar{p}$ be a (finite or infinite) path in $\mathcal{I}$. The constraint graph $\mathcal{G}_{\mathcal{I},\bar{p}}$ is given by the labeled subgraph of $\mathcal{G}_{\mathcal{I}}$ induced by $\{(u, x) \mid u \in \Delta^{\mathcal{I}}, u$ is on the path $\bar{p}, x \in \mathsf{Reg}_{C,\mathcal{T}}\}$.*

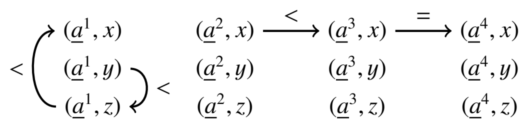


Fig. 2: The constraint graph $\mathcal{G}_{\underline{I}}$ of $\underline{I}$.

Example 4. The constraint graph of $\underline{I}$ from Example 3 is shown in Figure 2.

Theorem 4 ([4]). *C is satisfiable w.r.t. $\mathcal{T}$ if and only if there is an ordinary tree-shaped interpretation $\mathcal{I}$ s.t. $\mathcal{I} \models_{\mathcal{T}_a} C_a$ and s.t. there is a homomorphism from $\mathcal{G}_\mathcal{I}$ to $\mathcal{Z}$.*

A tableau reasoner may be used to verify that $\mathcal{I} \models_{\mathcal{T}_a} C_a$. As $\mathcal{I}$ may be infinite, such a reasoner constructs a finite representation of $\mathcal{I}$ in which the infinite parts of $\mathcal{I}$ are represented by paths between blocking and blocked nodes. The infinite version can be obtained by a construction known as unraveling. For more details, see [1]. The following observation will be later essential in our constructions:

Remark 5. W.l.o.g., we may assume that each path between a blocking node and its respective blocked node is of length $n \geq d$ where d is the maximal depth of a constraint appearing in $\mathcal{T}$ and C. This is because we can add d new concepts $A_1, \ldots, A_d$, plus the axioms $A_i \sqsubseteq \neg A_j$, for $i, j \in [d]$, $i \neq j$, together with $\top \sqsubseteq A_1 \sqcup \cdots \sqcup A_d$, and for all $r \in N_R$ also $A_1 \sqsubseteq \forall r.A_2, \ldots, A_{d-1} \sqsubseteq \forall r.A_d$, $A_d \sqsubseteq \forall r.A_1$, yielding an equisatisfiable TBox.

3 Integer graphs and Embeddings into $\mathcal{Z}$

For the rest of the paper, fix a concept C and a TBox $\mathcal{T}$ in $\mathcal{ALC}^P(\mathcal{Z})$ and let C_a and $\mathcal{T}_a$ be their abstractions. Fix some ordinary tree-shaped model $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ such that $\mathcal{I} \models_{\mathcal{T}_a} C_a$, given by a blocked tree satisfying the blocking condition described in Remark 5, and let $\mathcal{G}_\mathcal{I} = (G_\mathcal{I}, \mu_\mathcal{I})$ be its constraint graph. As mentioned before, deciding the satisfiability of C w.r.t. $\mathcal{T}$ breaks down to finding an ordinary model of the abstractions C_a and $\mathcal{T}_a$, and verifying that the constraints in its constraint graph are satisfiable. The latter may be done by finding a homomorphism from $\mathcal{G}_\mathcal{I}$ to $\mathcal{Z}$.

In this section we define integer graphs, which represent such homomorphisms in a general form. As the extraction of homomorphisms from a general integer graph is somewhat cumbersome, we also introduce compacted integer graphs which simplify said extraction.

Let G be a (possibly infinite) directed graph. Define $LPath_G : V(G)^2 \to \mathbb{Z}^+ \cup \{\infty\}$ as follows. For $u, v \in V(G)$, $LPath_G(u, v)$ is undefined if v is not reachable from u, is equal to k if k is the length of the longest path from u to v, and is equal to ∞ if for every $k \in \mathbb{Z}^+$, there is a path of length at least k from u to v. Let $\equiv$ be an equivalence relation on $V(G)$. For $v \in V(G)$, denote the equivalence class of v in $\equiv$ by $[v]_\equiv$. The *quotient graph* $G/\!\equiv$ is given by $V(G/\!\equiv) = \{[v]_\equiv \mid v \in V(G)\}$, and:

$$E(G/\!\equiv) = \{([u]_\equiv, [v]_\equiv) \mid (u', v') \in E(G) \text{ for some } u' \in [u]_\equiv, v' \in [v]_\equiv\}.$$

Define $\approx_{\mathcal{I}}$ to be the equivalence relation on $V(G_{\mathcal{I}})$ induced by the reflexive symmetric transitive closure of the edges in $E(G_{\mathcal{I}})$ labeled with $l_=$. Define $G^<_{\mathcal{I}} = (V(G_{\mathcal{I}}), E^<(G_{\mathcal{I}}))$ where $E^<(G_{\mathcal{I}}) = \{e \in E(G_{\mathcal{I}}) \mid \mu_{\mathcal{I}}(e) = l_<\}$. For a path $\bar{p}$ in $\mathcal{I}$, define $\approx_{\bar{p}}$ and $G^<_{\bar{p}}$ similarly.

Definition 6 (Integer graph). *Let $\approx$ be an equivalence relation on $V(G_{\mathcal{I}})$ such that $\approx_{\mathcal{I}}$ refines $\approx$, and denote $G^<_{\mathcal{I}}/\approx$ by $H^{\approx}_{\mathcal{I}}$. If $H^{\approx}_{\mathcal{I}}$ is acyclic and if for every $a, b \in V(H^{\approx}_{\mathcal{I}})$, $LPath_{H^{\approx}_{\mathcal{I}}}(a, b) \neq \infty$, we say $H^{\approx}_{\mathcal{I}}$ is an* integer graph. *We say $H^{\approx}_{\mathcal{I}}$ is a* compacted integer graph *if, in addition, for every $a, b \in V(H^{\approx}_{\mathcal{I}})$, either a is reachable from b or vice versa. For a path $\bar{p}$ in $\mathcal{I}$ and $\approx$ such that $\approx_{\bar{p}}$ refines $\approx$, define $H^{\approx}_{\bar{p}}$ similarly.*

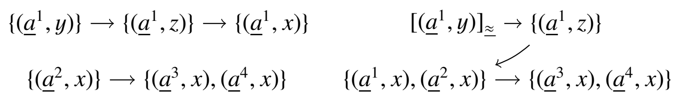
When $H^{\approx_{\mathcal{I}}}_{\mathcal{I}}$ is an integer graph, we often refer to it as *the* integer graph of $\mathcal{I}$.

Definition 7 (Vertex contraction). *Let $G = (V(G), E(G))$ and let $a, b \in V(G)$. The result of* contracting *a and b, denoted $G^{/\{a,b\}}$, is given by replacing a and b with a fresh vertex c whose neighborhood is the union of the neighborhoods of a and b. That is, $V(G^{/\{a,b\}}) = (V(G) \setminus \{a, b\}) \cup \{c\}$ and for every $a' \in V(G) \setminus \{a, b\}$, we have $(a', c) \in E(G^{/\{a,b\}})$ iff $\{(a', a), (a', b)\} \cap E(G) \neq \emptyset$ (and similarly for (c, a')). If G is quotient graph, the fresh vertex c is the union of the equivalence classes a and b.*

Notice that vertex contractions do not introduce new edges. For a quotient graph $G/\equiv$, we have that the equivalence relation $\equiv$ refines the equivalence relation obtained after contracting vertices in $G/\equiv$. Hence, we have that all the integer graphs of a given model $\mathcal{I}$ can be obtained from $H^{\approx_{\mathcal{I}}}_{\mathcal{I}}$ by contracting vertices in $H^{\approx_{\mathcal{I}}}_{\mathcal{I}}$. Moreover, we have:

Lemma 8. *If $H^{\approx_{\mathcal{I}}}_{\mathcal{I}}$ is an integer graph, $\mathcal{I}$ has a compacted integer graph.*

Example 5. The equivalence relation $\approx_{\underline{\mathcal{I}}}$ of $\underline{\mathcal{I}}$ from Example 3 only contains the pair $((\underline{a}^3, x), (\underline{a}^4, x))$. The integer graph $H^{\approx_{\underline{\mathcal{I}}}}_{\underline{\mathcal{I}}}$ is shown in Figure 3(a) with the isolated vertices omitted. A compaction of $H^{\approx_{\underline{\mathcal{I}}}}_{\underline{\mathcal{I}}}$ is given by the integer graph induced by the equivalence relation $\underline{\approx}$, where: $(\underline{a}^3, x)$ is equivalent to $(\underline{a}^4, x)$, $(\underline{a}^1, x)$ is equivalent to $(\underline{a}^2, x)$, and $(\underline{a}^1, y)$, $(\underline{a}^2, y)$, $(\underline{a}^2, z)$, $(\underline{a}^3, y)$, $(\underline{a}^3, z)$, $(\underline{a}^4, y)$, $(\underline{a}^4, z)$ are equivalent to each other, and $(\underline{a}^1, z)$ is only equivalent to itself. Note that $\approx_{\underline{\mathcal{I}}}$ indeed refines $\underline{\approx}$, $H_{\underline{\approx}}$ (shown in Figure 3(b)) is acyclic, and for any pair of vertices, one is reachable from another. The vertex $[(\underline{a}^1, y)]_{\underline{\approx}}$ denotes the equivalence class $\{(\underline{a}^1, y), (\underline{a}^2, y), (\underline{a}^2, z), (\underline{a}^3, y), (\underline{a}^3, z), (\underline{a}^4, y), (\underline{a}^4, z)\}$.



(a) $H^{\approx_{\underline{\mathcal{I}}}}_{\underline{\mathcal{I}}}$ (b) $H_{\underline{\approx}}$

Fig. 3: The integer graph $H^{\approx_{\underline{\mathcal{I}}}}_{\underline{\mathcal{I}}}$ and its compaction $H_{\underline{\approx}}$.

Note that integer graphs naturally induce a homomorphism (in fact, infinitely many) from $\mathcal{G}_I$ to $\mathcal{Z}$, as they are directed acyclic graphs:

Observation 9. Let H be an integer graph. There exists a function $num : V(H) \to \mathbb{Z}$ such that for $a, b \in V(H)$ where $a \neq b$, if b is reachable from a, then $num(a) < num(b)$.

This allows us to reduce the problem of deciding the existence of a homomorphism to checking whether there exists an integer graph for $\mathcal{I}$:

Lemma 10. *There is a homomorphism from $\mathcal{G}_{\mathcal{I}}$ to $\mathcal{Z}$ iff $\mathcal{I}$ has an integer graph.*

Combining this with the previous result of Carapelle and Turhan, we have:

Corollary 11. *C is satisfiable w.r.t. $\mathcal{T}$ if and only if there exists an ordinary model $\mathcal{I}$ of C_a w.r.t. $\mathcal{T}_a$ which has an integer graph.*

The next lemma will allow us to break the task of deciding whether $\mathcal{I}$ has an integer graph into simpler steps:

Lemma 12. *$\mathcal{I}$ has an integer graph if and only if every path in $\mathcal{I}$ which starts at the root has an integer graph.*

Proof (Sketch). One direction is immediate. For the other direction, assume every path starting at the root in $\mathcal{I}$ has an integer graph. We show the claim by induction on the number T of (blocked or maximal and finite) paths in $\mathcal{I}$. The case where $T = 1$ is immediate. Assume the claim holds for T. Let $\mathcal{I}$ have $T + 1$ such paths which all have integer graphs. Let $\bar{p}$ be one of these paths, and consider the sub-tree $\mathcal{I}'$ induced by the T paths in $\mathcal{I}$ other than $\bar{p}$. Then by the induction hypothesis, $\mathcal{I}'$ has an integer graph $H_{\mathcal{I}'}^{\approx_{\mathcal{I}'}}$. Let $H_{\bar{p}}^{\approx_{\bar{p}}}$ be the integer graph of $\bar{p}$. Let $\Delta_{\bar{p}}^{\mathcal{I}}$ denote the elements on $\bar{p}$, and let $\Delta^{\mathcal{I}'}$ denote the elements in $\mathcal{I}'$. Then the integer graph $H_{\mathcal{I}}^{\approx_{\mathcal{I}}}$ of $\mathcal{I}$ is as follows. $H_{\mathcal{I}}^{\approx_{\mathcal{I}}}$ is given by taking the disjoint union of $H_{\bar{p}}^{\approx_{\bar{p}}}$ and $H_{\mathcal{I}'}^{\approx_{\mathcal{I}'}}$ and contracting (see Definition 7) the vertices $[(u, x)]_{\approx_{\bar{p}}}$ and $[(u, x)]_{\approx_{\mathcal{I}'}}$ for every $u \in \Delta_{\bar{p}}^{\mathcal{I}} \cap \Delta^{\mathcal{I}'}$ and $x \in \mathsf{Reg}_{C,\mathcal{T}}$.

We have now established that in order to check satisfiability of C w.r.t. $\mathcal{T}$, it is enough to obtain an ordinary tree-shaped model $\mathcal{I}$ for their abstractions and to check that each path in $\mathcal{I}$ starting at the root has an integer graph.

4 Constructing Integer graphs

In this section, we describe how the existence of an integer graph for a path in $\mathcal{I}$ can be tested. First, observe that for finite paths, this task is simple:

Observation 13. Let $\bar{p}$ be a finite path in $\mathcal{I}$. It is straightforward to construct its constraint graph $\mathcal{G}_{\bar{p}}$, compute $\approx_{\bar{p}}$ and then construct the graph $H_{\bar{p}}^{\approx_{\bar{p}}}$ and test it for for being acyclic, as these are all finite structures. Together with the construction described in the proof of Lemma 12, we have that we can compute the integer graph (if it exists) of any finite tree-shaped prefix of $\mathcal{I}$.

We still need conditions for the existence of integer graphs for infinite ultimately periodic paths which can be verified in finite time.[4] Such paths $\bar{\alpha}(\bar{\beta})^{\omega}$ require caution, since it is possible for their prefixes $\bar{\alpha}(\bar{\beta})^{t}$ have an integer graph for any $t \in \mathbb{Z}^{+}$ while they themselves do not. This behavior is demonstrated in the following example:

[4] We will sometimes abuse notation and denote an ultimately periodic path by $\bar{\alpha}(\bar{\beta})^{\omega}$, where $\bar{\alpha}$ and $\bar{\beta}$ list the *labels* appearing in the non-periodic and respectively periodic parts of the path.

Example 6. Let $\underline{\theta}_5 := x = \mathrm{S}^1 x$, $\underline{\theta}_6 := y = \mathrm{S}^1 y$, and $\underline{\theta}_7 := z < \mathrm{S}^1 z$, and let $\underline{B}_{5,6,7}$ be a placeholder for $\underline{\theta}_5 \wedge \underline{\theta}_6 \wedge \underline{\theta}_7$. Let $\underline{B}_{1,2}$ be a placeholder for $\underline{\theta}_{1,2}$ (see Example 1). Consider the ultimately periodic path $\underline{b}^1 \underline{b}^2 \ldots$, where $\underline{b}^1$ is labeled with $\underline{B}_{1,2}$ and for $i \geq 2$, $\underline{b}^i$ is labeled with $\underline{B}_{5,6,7}$. Essentially, these constraints fix two integer values, held in the x and y registers, and with each repetition of $\underline{B}_{5,6,7}$, add a fresh integer value between them. Hence, for any $t \in \mathbb{Z}^+$, $\underline{b}^1 \ldots \underline{b}^{t+1}$ has an integer graph which is a simple path of length $t + 1$, since for any $t \in \mathbb{Z}^+$ there is a choice of two integer values with enough different values between them. See Figure 4 for the integer graph of $\underline{b}^1 \ldots \underline{b}^4$, with the isolated vertices omitted and where $[(\underline{b}^1, x)] = \{(\underline{b}^i, x) \mid i \in [4]\}$ and $[(\underline{b}^1, y)] = \{(\underline{b}^i, y) \mid i \in [4]\}$. However, there is no integer graph for $\underline{b}^1 \underline{b}^2 \ldots$, since no two integer values have infinitely many different integers between them.

$$[(\underline{b}^1, x)] \to \{(\underline{b}^4, z)\} \to \{(\underline{b}^3, z)\} \to \{(\underline{b}^2, z)\} \to \{(\underline{b}^1, z)\} \to [(\underline{b}^1, y)]$$

Fig. 4: Integer graph of $\underline{b}^1 \underline{b}^2 \underline{b}^3 \underline{b}^4$.

We first need to introduce the notion of integer graph extensions, which can be roughly seen as the subgraph relation adapted to quotient graphs:

Definition 14 (Integer graph extension). *Let H' and H be integer graphs. We say H'* extends *H if for every $[v] \in V(H)$, 1. there exists $[v'] \in V(H')$ such that $[v] \subseteq [v']$, and 2. if $([u], [v]) \in E(H)$ for some $[u] \in V(H)$, then $([u], [v]) \in E(H')$.*

The next easy lemma follows from the definitions:

Lemma 15. *Let $\bar{p}$ be a (finite or infinite) path in $\mathcal{I}$ and let $\bar{\alpha}$ be a prefix of $\bar{p}$. If $\bar{p}$ has an integer graph $H_{\bar{p}}$, then $\bar{\alpha}$ has an integer graph $H_{\bar{\alpha}}$ and $H_{\bar{p}}$ extends $H_{\bar{\alpha}}$.*

The behavior in Example 6 we wish to avoid, namely, having the length of the longest path between two vertices grow in subsequent extensions, can be expressed using the following:

Definition 16 (Non-disrupting extensions). *Let H'', and H', and H be integer graphs such that H'' extends H' and H' extends H. We say that H''* extends *H'* without disrupting *H if for every $u, v \in V(H)$, $LPath_{H'}(u, v) = LPath_{H''}(u, v)$.*

Through these two notions we can characterize the (finite or infinite) paths which have integer graphs, using conditions which can be verified in finite time. The following lemma is the most technically involved result of the paper. Recall that n denotes the maximal depth of constraints appearing in $\mathcal{I}$.

Lemma 17. *Let $\bar{p} = \bar{\alpha}(\bar{\beta})^\omega$ be an ultimately periodic path in $\mathcal{I}$, beginning at the root. Then $\bar{p}$ has an integer graph if and only if:*

(C) $\bar{\alpha}(\bar{\beta})^{n+1}$ has an integer graph $H_{\bar{p},n+1}$ which extends $H_{\bar{p},n}$ without disrupting $H_{\bar{p},1}$, where $H_{\bar{p},1}$ and $H_{\bar{p},n}$ are the integer graphs of $\bar{\alpha}(\bar{\beta})$ and $\bar{\alpha}(\bar{\beta})^n$, respectively.

The following example shows that the bound in Lemma 17 on the number $n + 1$ of repetitions needed is tight:

Example 7. Let $\underline{\theta}_8 := x < y$ with placeholder $\underline{B}_8$ and let $\underline{\theta}_{9,10} := (x < \mathrm{S}^3 x) \wedge (\mathrm{S}^3 y < y)$ with placeholder $\underline{B}_{9,10}$. Consider the path $\underline{u}_1 \underline{u}_2 \ldots$, where if $i = 1 \mod 4$, $\underline{u}_i$ is labeled with $\underline{B}_8$, and otherwise $\underline{u}_i$ is labeled with $\underline{B}_8 \sqcap \underline{B}_{9,10}$. It is straightforward, albeit tedious, to verify that the integer graph of $\underline{u}_1 \underline{u}_2 \underline{u}_3 \underline{u}_4$ is disrupted by its extensions up to $i = 16$.

Also note that condition (C) may be verified for a path $\bar{\alpha}(\bar{\beta})^\omega$ in finite time; by Observation 13, it is straightforward to construct the graphs $H_{\bar{p},1}$, $H_{\bar{p},n}$, $H_{\bar{p},n+1}$ and to verify they are acyclic. Then it is immediately verifiable that for every $u, v \in V(H_{\bar{p},1})$, it holds that $LPath_{H_{\bar{p},n}}(u, v) = LPath_{H_{\bar{p},n+1}}(u, v)$.

Proof (Sketch). Recall that $\mathcal{I}$, and therefore also $\bar{p}$, satisfies the blocking condition described in Remark 5. To reduce notational clutter, in this proof we omit $\bar{p}$ and denote by H_t the integer graph of $\bar{\alpha}(\bar{\beta})^t$ where relevant.

Our blocking condition allows us to define a graph operation which, in a way, appends some fixed graph induced by the periodic part of $\bar{p}$ to H_t in order to obtain H_{t+1}. This regularity in the structure of the integer graphs of the prefixes of $\bar{p}$ allows us to show that if condition (C) holds, then (*) for every $t \in \mathbb{Z}^+$, $\bar{\alpha}(\bar{\beta})^{t+n}$ has an integer graph H_{t+n} which extends H_{t+n-1} without disrupting H_t. We can further show that acyclicity is preserved when taking the limit $H_\infty = \lim_{t\to\infty} H_t$. Therefore, if H_∞ is not the integer graph of $\bar{p}$, there are $u, v \in V(H_\infty)$ such that $LPath_{H_\infty}(u, v) = \infty$. Fix such u and v. Let $t \in \mathbb{Z}^+$ be the minimal number for which $u, v \in V(H_t)$. We have that $\bar{\alpha}(\bar{\beta})^t$ is finite, so H_t is a finite directed acyclic graph, therefore $LPath_{H_t}(u, v)$ is also finite. Since by our assumption, $LPath_{H_\infty}(u, v) = \infty$, there exists some minimal $t' \geq t + n$ for which $LPath_{H_{t'}}(u, v) > LPath_{H_t}(u, v)$. Then we have that $H_{t'}$ disrupts H_t. In fact, we have that $H_{t'}$ disrupts $H_{t''}$ for every $t \leq t'' < t'$. In particular, for $t'' = t' - n$ we have that $t \leq t'' < t'$, so $H_{t'} = H_{t''+n}$ disrupts $H_{t''}$ while extending $H_{t''+n-1}$, in contradiction to (*). We conclude that there are no $u, v \in V(H_\infty)$ such that $LPath_{H_\infty}(u, v) = \infty$. Therefore H_∞ satisfies all conditions of Definition 6, making it the integer graph of $\bar{p}$.

For the other direction, assume that $\bar{p}$ has an integer graph H. First, note that $\bar{\alpha}(\bar{\beta})^t$ is a prefix of $\bar{p}$ for every $t \geq 0$. Therefore, by Lemma 15, for every $t \in \mathbb{Z}^+$, $\bar{\alpha}(\bar{\beta})^t$ has an integer graph H_t. Furthermore, for every $t < t'$, we have that $H_{t'}$ extends H_t. In particular, this is true for $t = n$ and $t' = n + 1$. Therefore, to satisfy condition (C), it remains to show that H_{n+1} does not disrupt H_1 when extending H_n.

Assume for contradiction that H_{n+1} does disrupt H_1 when extending H_n. Then, relying on the fact that the depth of the placeholders in $\bar{\beta}$ is limited by the length of $\bar{\beta}$, we can show that there are $w, w' \in V(H_1)$ such that $LPath_{H_t}(w, w') < LPath_{H_{t+1}}(w, w')$ for every $t \geq n$. Since the length of the longest path between w and w' grows with each t, we have that for every $k \in \mathbb{Z}^+$, there exists some $t_k \in \mathbb{Z}^+$ such that $LPath_{H_{t_k}}(w, w') > k$. Hence, in H, we have that $LPath_H(w, w') = \infty$, in contradiction to our assumption that H is an integer graph. We conclude that H_{n+1} does not disrupt H_1 when extending H_n, and condition (C) holds.

5 Putting it Together – the Constructive Procedure

It remains to construct an assignment to the registers. Since a homomorphism can be easily extracted from a compacted integer graph, we show in this section that such compactions are effectively computable. Let $\bar{P}_{\text{fin}}$ be the set of maximal finite paths in $\mathcal{I}$ beginning at the root, and let $\bar{P}_{\text{prd}}$ be the set of infinite ultimately periodic paths in $\mathcal{I}$ beginning at the root. For $t \in \mathbb{Z}^+$, let $U_t = \{u \in \bar{p} \mid \bar{p} \in \bar{P}_{\text{fin}}\} \cup \{u \in \bar{\alpha}(\bar{\beta})^t \mid \bar{\alpha}(\bar{\beta})^\omega \in \bar{P}_{\text{prd}}\}$. That is, all elements that appear either on maximal finite paths or by the t-th iteration of the ultimately periodic paths. Define $\mathcal{I}_t$ be the sub-tree of $\mathcal{I}$ induced by the elements in U_t. Let $u_1, u_2, \ldots$ be a lexicographical ordering of $\Delta^{\mathcal{I}} \times \mathsf{Reg}_{C,\mathcal{T}}$.

Algorithm 1 Integer graph compaction

Input: An integer graph H_t of $\mathcal{I}_t$ for $t \in \mathbb{Z}^+$
Output: A compaction $H_{t,c}$ of H_t
1: $V(H_{t,c}) = V(H_t)$
2: $E(H_{t,c}) = E(H_t)$
3: $a = [u_1]$ ▷ We refer to the equivalence classes in $H_{t,c}$
4: **for** $b, b' \in V(H_{t,c})$ **do**
5:     **if** $LPath_{H_{t,c}}(a,b) = LPath_{H_{t,c}}(a,b') \in \mathbb{Z}$ or $LPath_{H_{t,c}}(b,a) = LPath_{H_{t,c}}(b',a) \in \mathbb{Z}$ **then**
6:         contract b and b' in $V(H_{t,c})$
7: **for** $i = 2, \ldots, |V(H_t)|$ **do**
8:     **if** a is not reachable from $[u_i]$ in $H_{t,c}$ nor vice versa **then**
9:         contract a and $[u_i]$ in $V(H_{t,c})$
10:     $a = [u_i]$
11:     **for** $b, b' \in V(H_t)$ **do**
12:         **if** $LPath_{H_{t,c}}(a,b) = LPath_{H_{t,c}}(a,b') \in \mathbb{Z}$ or $LPath_{H_{t,c}}(b,a) = LPath_{H_{t,c}}(b',a) \in \mathbb{Z}$ **then**
13:             contract b and b' in $V(H_{t,c})$
14: $H_{t,c} = (V(H_{t,c}), E(H_{t,c}))$
15: **return** $H_{t,c}$

Lemma 18. *Let $t \in \mathbb{Z}^+$ and let H_t be the integer graph of $\mathcal{I}_t$. There is an effectively computable compaction $H_{t,c}$ of H_t.*

Proof (Sketch). We claim that Algorithm 1 outputs a compaction of H_t. Acyclicity is preserved by operations in Lines 6, 9, and 13. As for the reachability condition, first note that for every $b \in V(H_{t,c})$, either b is reachable from $[u_1]$ or vice versa. Let $b, b' \in V(H_{t,c})$. Assume for contradiction that b is not reachable from b' nor vice versa. Then, either both b, b' are reachable from $[u_1]$, or $[u_1]$ is reachable from both. Assume w.l.o.g. that it is the former. If $LPath_{H_t}(u_1, b) = LPath_{H_t}(u_1, b')$, then b and b' would have been contracted during the run of the algorithm. Therefore either $LPath_{H_t}(u_1, b) < LPath_{H_t}(u_1, b')$ or $LPath_{H_t}(u_1, b) > LPath_{H_t}(u_1, b')$. If $LPath_{H_t}(u_1, b) < LPath_{H_t}(u_1, b')$, there is some $b'' \in V(H_{t,c})$ such that $LPath_{H_t}(u_1, b') = LPath_{H_t}(u_1, b'') + LPath_{H_t}(b'', b')$ and such that $LPath_{H_t}(u_1, b'') = LPath_{H_t}(u_1, b)$. Hence, b and b'' would have been contracted during the run of the algorithm, producing a path from b to b', in contradiction to our assumption. The other case is analogous. Therefore, for every $b, b' \in V(H_{t,c})$, either b is reachable from b' or vice versa, making $H_{t,c}$ a compacted integer graph.

Theorem 19. *Let $\mathcal{I}$ have an integer graph. There is an effectively computable valuation function $\gamma : \Delta^{\mathcal{I}} \times \mathsf{Reg}_{C,\mathcal{T}} \to \mathcal{Z}$ such that $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \gamma)$ satisfies C w.r.t. $\mathcal{T}$.*

Proof (Sketch). We describe a procedure which, given input $(u, x) \in \mathsf{Reg}_{C,\mathcal{T}}$, outputs an integer value. It first constructs a compacted integer graph using Observation 13 and Algorithm 1, then assigned it a value based on the length of the longest path between $[(u, x)]$ and $[(\varepsilon, x_1)]$. For $t \in \mathbb{Z}^+$, denote by H_t the integer graph of $\mathcal{I}_t$. Let $t \in \mathbb{Z}^+$ be the minimal t for which $[(u, x)]$ appears in H_t, and denote $t' = t + n + 1$. Generate $H_{t'}$ using Observation 13 and then generate the compacted $H_{t',c}$ using Algorithm 1. Denote $[(u, x)] = b$ and $[(\varepsilon, x_1)] = a$ and return $\gamma(u, x) = -LPath_{H_{t',c}}(b, a)$ if $LPath_{H_{t',c}}(b, a) \in \mathbb{Z}^+$, and $\gamma(u, x) = LPath_{H_{t',c}}(a, b)$ otherwise. Recall that for every $b \in V(H_{t,c})$ we have that either b is reachable from a or vice versa, therefore γ is well-defined. Also note that our blocking condition together with Lemma 17 guarantee that the length of the longest path between two vertices is final after n iterations, therefore, if $[(u, x)]$ is reachable (and different) from $[(v, y)]$, then $\gamma(u, x) > \gamma(v, y)$. Now apply Lemma 10.

6 Conclusion

$\mathcal{ALC}^P(\mathcal{D})$ is an interesting family of DLs which enable the expression of constraints over register values of individuals along a specified path, where the values and the constraints are over a domain structure $\mathcal{D}$. In this work, we have provided a constructive satisfiability procedure for $\mathcal{ALC}^P(\mathcal{Z})$ instantiated to the integer domain $\mathcal{Z} = \langle \mathbb{Z}, =, < \rangle$. The construction is two-fold: in order to construct the $\mathcal{Z}$-model of an $\mathcal{ALC}^P(\mathcal{Z})$ concept C w.r.t. a TBox $\mathcal{T}$, first an ordinary $\mathcal{ALC}$-model $\mathcal{I}$ is constructed for the abstracted versions of $\mathcal{T}$ and C; and consecutively the so-called integer graph of $\mathcal{I}$, which provides a general representation of the relationships between the registers, is constructed and used for assigning the registers with values from $\mathbb{Z}$.

We believe that this result will enable to employ $\mathcal{ALC}^P(\mathcal{Z})$ in reasoning tasks where model construction is required, e.g., ABox abduction and query answering. Although complexity issues were not addressed in this paper, a complexity upper bound can be obtained by analyzing our procedures. This would already be an improvement, as the reductions in [4] lead to an automata model with a decidable emptiness problem of non-elementary complexity, [3]. However, since such a naïve bound is likely to not be complete, a natural sequel to this work would be to refine our approach to derive a tight complexity upper bound for this logic.

Acknowledgments. We thank Mantas Šimkus for comments on early stages of this work, and we are grateful to the anonymous reviewers for their fruitful comments. This work was supported by the FWF (project W1255).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991. pp. 452–457. Morgan Kaufmann (1991)
3. Bojańczyk, M., Toruńczyk, S.: Weak MSO+U over infinite trees. In: 29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th–March 3rd, 2012, Paris, France. pp. 648–660 (2012)
4. Carapelle, C., Turhan, A.: Description logics reasoning w.r.t. general TBoxes is decidable for concrete domains with the EHD-property. In: ECAI 2016 – 22nd European Conference on Artificial Intelligence, 29 August–2 September 2016, The Hague, The Netherlands. Frontiers in Artificial Intelligence and Applications, vol. 285, pp. 1440–1448. IOS Press (2016)
5. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. J. Web Semant. 6(4), 309–322 (2008)
6. Elsenbroich, C., Kutz, O., Sattler, U.: A case for abductive reasoning over ontologies. In: Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions, Athens, Georgia, USA, November 10–11, 2006 (2006)
7. Haarslev, V., Möller, R., Wessel, M.: The description logic $\mathcal{ALCNH}_{R+}$ extended with concrete domains: A practically motivated approach. In: Automated Reasoning, First International Joint Conference, IJCAR 2001, Siena, Italy, June 18–23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2083, pp. 29–44. Springer (2001)
8. Halland, K., Britz, K.: Abox abduction in $\mathcal{ALC}$ using a DL tableau. In: 2012 South African Institute of Computer Scientists and Information Technologists Conference, SAICSIT '12, Pretoria, South Africa, October 1–3, 2012. pp. 51–58 (2012)
9. Horrocks, I., Sattler, U.: Ontology reasoning in the $\mathcal{SHOQ}$(D) description logic. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001. pp. 199–204. Morgan Kaufmann (2001)
10. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16–22, 2011. pp. 2656–2661 (2011)
11. Lutz, C.: NExpTime-complete description logics with concrete domains. ACM Trans. Comput. Log. 5(4), 669–705 (2004)
12. Lutz, C., Miličić, M.: A tableau algorithm for description logics with concrete domains and general TBoxes. J. Autom. Reasoning 38(1–3), 227–259 (2007)
13. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic EL using a relational database system. In: IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11–17, 2009. pp. 2070–2075 (2009)
14. Ortiz, M., Calvanese, D., Eiter, T.: Data complexity of query answering in expressive description logics via tableaux. Journal of Automated Reasoning 41(1), 61–98 (2008)
15. Pukancová, J., Homola, M.: Tableau-based ABox abduction for description logics: Preliminary report. In: Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22–25, 2016. (2016)