Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Description Logics with Abstraction and Refinement: From $\mathcal{ALC}$ to $\mathcal{EL}$

Carsten Lutz[1,2] , **Lukas Schulze**[1]
[1]Department of Computer Science, Leipzig University, Germany
[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
{clu, lschulze}@informatik.uni-leipzig.de

Abstract

We study extensions of description logics from the widely used $\mathcal{EL}$ family with operators that make it possible to speak about different levels of abstraction. We analyze the computational complexity of reasoning and show that often, this complexity is significantly lower than in the corresponding extension of the more expressive description logic $\mathcal{ALC}$. By slightly varying the semantics, we also obtain a case that admits reasoning in polynomial time.

cr	ca	rr	ra	Semantics	$\mathcal{EL}$ / $\mathcal{ELH}_r$	$\mathcal{ALC}$
X				standard	coNP	Exp
X	X			standard	PSpace	2Exp
		X		standard	2Exp	2Exp
X	X	X	X	standard	2Exp	2Exp
X		X		set ensemble	PTime	Exp
X	X			set ensemble	coNP-hard	?

Figure 1: The complexity of satisfiability in abstraction DLs.

1 Introduction

Knowledge representation with ontologies often involves concepts that are situated at different levels of abstraction or, equivalently, at different levels of granularity. For example, the widely known medical ontology SNOMED CT contains the concepts Arm, Hand, Finger, Phalanx, Osteocyte, and Mitochondrion which may reasonably be viewed as all belonging to different, increasingly more fine-grained levels of abstraction. Existing ontology languages, however, do not provide any explicit support for representing and interrelating different abstraction levels.

Recently, this shortcoming has led to the proposal of a scheme for extending description logics (DLs) with operators that make it possible to explicitly speak about different abstraction levels and their interaction (Lutz and Schulze 2023). The main features of this scheme are as follows. Each of the (finitely many) abstraction levels is associated with a classical DL interpretation. A refinement function associates objects on more coarse-grained levels with an ensemble of objects on more fine-grained levels. Such an ensemble is simply a tuple of objects that the refined object decomposes into. This may for instance be in the sense of mereological parts, but the scheme is by no means restricted to mereology.

Operators based on conjunctive queries (CQs) make it possible to describe how objects relate to their refining ensembles. These operators come in two flavours. A *refinement operator* expresses that every object of a certain kind refines into a certain kind of ensemble. For example, the statement

$$L_2{:}q_A \ \underline{\mathsf{refines}}\ L_1{:}\mathsf{Aircraft},$$

where q denotes the conjunctive query

$$q_A = \mathsf{Fuselage}(x_1) \wedge \mathsf{Wings}(x_2) \wedge \mathsf{Stabilizer}(x_3) \wedge \mathsf{carries}(x_2, x_1) \wedge \mathsf{carries}(x_1, x_3),$$

expresses that every instance of Aircraft on the more coarse-grained abstraction level L_1 decomposes into an ensemble of three objects on the more fine-grained level L_2, as described by q_A. Conversely, an *abstraction operator* expresses that for every ensemble of a certain kind, there is an object that refines into it. Reusing the query q_A from above, for example, it would be reasonable to also state

$$L_1{:}\mathsf{Aircraft}\ \underline{\mathsf{abstracts}}\ L_2{:}q_A$$

expressing that every ensemble that consists of a fuselage, a set of wings, and a stabilizer, related as stated by q_A, forms an aircraft. While the operators illustrated above speak about concepts such as Aircraft that are refined or abstracted, there are analogous operators also for roles (that is, binary relations) such as carries.

The DLs with abstraction and refinement proposed in (Lutz and Schulze 2023) are based on the expressive description logics $\mathcal{ALC}$ and $\mathcal{ALCI}$. In this paper, we replace $\mathcal{ALC}$ with important members of the $\mathcal{EL}$ family of description logics, in particular with the eponymous $\mathcal{EL}$ and its extension $\mathcal{ELH}_r$ with role hierarchies and range restrictions. These DLs play an important role in practice for at least three reasons. First, they are among very few description logics that admit reasoning in polynomial time. Second, a mild extension of $\mathcal{ELH}_r$ was standardized by the W3C as the EL profile of the widely used OWL 2 ontology language (Motik *et al.* 2009). And third, many prominent large-scale ontologies such as SNOMED CT are formulated in $\mathcal{ELH}_r$ or mild extensions thereof.

Two guiding questions for our investigation are: (1) Are the resulting DLs with abstraction and refinement computationally more well-behaved than those based on $\mathcal{ALC}$? And (2) Can we even identify useful cases where reasoning is possible in polynomial time? We remark that polynomial time cannot be expected in the presence of abstraction operators because, whenever these operators are present, then there is an obvious polynomial time reduction from the homomorphism problem on directed graphs; this implies that reasoning (concept satisfiability, to be precise) is at least NP-hard. Refinement operators, however, do not preclude polynomial time reasoning up-front.

We first prove that the extension $\mathcal{ELH}_r^{\mathsf{abs}}$ of $\mathcal{ELH}_r$ with abstraction and refinement operators for both concepts and roles still enjoys the existence of universal models (defined in terms of homomorphisms). This is important because the existence of universal models makes a crucial difference when designing algorithms, and in fact universal models underlie all important polynomial time reasoning algorithms for description logics. To construct universal models, we give a non-trivial chase procedure tailored specifically to $\mathcal{ELH}_r^{\mathsf{abs}}$. The algorithms behind our upper complexity bounds then all rely on universal models.

Our findings on the complexity of satisfiability in $\mathcal{ELH}_r^{\mathsf{abs}}$ and various fragments thereof are summarized in Figure 1. There, 'cr' stands for concept refinement operators, 'ca' for concept abstraction, and likewise for 'rr' and 'ra' and roles in place of concepts. We remark that subsumption and (un)satisfiability can be reduced to one another in polynomial time in all considered logics. All stated results are completeness results with the lower bounds holding already for (the respective fragments of) $\mathcal{EL}^{\mathsf{abs}}$ and the upper bounds applying to $\mathcal{ELH}_r^{\mathsf{abs}}$. The results shown in gray are from (Lutz and Schulze 2023).

Full $\mathcal{ELH}_r^{\mathsf{abs}}$ and $\mathcal{EL}^{\mathsf{abs}}$ turn out to be computationally no more well-behaved than in the case where $\mathcal{ALC}$ is used as the base logic: satisfiability is 2EXPTIME-complete in both cases. This still holds when only role refinement is admitted. The picture changes, however, in the important case where only concept-based operators are used, but no role-based ones. With only concept refinement, the complexity reduces to coNP which we consider a significant improvement as it enables the use of SAT solvers to decide satisfiability. With both concept refinement and abstraction, satisfiability is PSPACE-complete which is still significantly lower than 2EXPTIME-completeness in the case where $\mathcal{ALC}$ is used as the base logic.

To attain polynomial time, we change the semantics: instead of tuples of objects, ensembles are now sets of objects. While this has a subtle impact on modeling (see Example 4 in the paper), it is still a very reasonable semantics. Under this semantics, we indeed achieve polynomial time reasoning when only concept and role refinement is admitted.

To comply with space restrictions, proof details are provided in the appendix, available at https://www.informatik.uni-leipzig.de/kr/research/papers.html.

Related Work. As already explained, we adopt the framework of (Lutz and Schulze 2023). It is loosely related to description logics of context (Klarman and Gutiérrez-Basulto 2016) and to other multi-dimensional DLs (Wolter and Zakharyaschev 1999). Granularity has also received attention in foundational ontologies, see e.g. (Bittner and Smith 2003). There are other approaches to combine description logic and abstraction/granularity, but from very different perspectives and in technically very different ways, see for example (Calegari and Ciucci 2010; Cima *et al.* 2022; Glimm *et al.* 2017; Lisi and Mencar 2018).

2 Preliminaries

Fix countably infinite sets $\mathbf{C}$ and $\mathbf{R}$ of *concept names* and *role names*. $\mathcal{EL}$*-concepts* C, D take the form $C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C$ where A ranges over concept names and r over role names. An $\mathcal{ELH}_r$*-ontology* is a finite set $\mathcal{O}$ of *concept inclusion (CIs)* $C \sqsubseteq D$ with C and D $\mathcal{EL}$-concepts, *role inclusions* $r \sqsubseteq s$ with $r, s \in \mathbf{R}$, and *range restrictions* $\top \sqsubseteq \forall r.C$ with $r \in \mathbf{R}$ and C an $\mathcal{EL}$-concept. We say $\mathcal{O}$ is an $\mathcal{EL}$-ontology if it contains no role inclusions and range restrictions.

An *interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ with $\Delta^\mathcal{I}$ a non-empty set (the *domain*) and $\cdot^\mathcal{I}$ an *interpretation function* that maps every concept name $A \in \mathbf{C}$ to a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and every role name $r \in \mathbf{R}$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The interpretation function is extended to compound concepts by setting $\top^\mathcal{I} = \Delta^\mathcal{I}$, $(C_1 \sqcap C_2)^\mathcal{I} = C_1^\mathcal{I} \cap C_2^\mathcal{I}$, and $(\exists r.C)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid \exists e \in C^\mathcal{I} : (d, e) \in r^\mathcal{I}\}$. An interpretation $\mathcal{I}$ *satisfies* a concept inclusion $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$ and likewise for role inclusions; it satisfies a range restriction $\top \sqsubseteq \forall r.C$ if $(d, e) \in r^\mathcal{I}$ implies $e \in C^\mathcal{I}$. We say that $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies all inclusions and range restrictions in $\mathcal{O}$. We write $\mathcal{O} \models r \sqsubseteq s$ if every model of $\mathcal{O}$ satisfies $r \sqsubseteq s$. One can decide whether $\mathcal{O} \models r \sqsubseteq s$ in polynomial time by computing the reflexive-transitive closure of the role inclusions in $\mathcal{O}$.

A *conjunctive query (CQ)* $q(\bar{x})$ takes the form $q(\bar{x}) \leftarrow \varphi(\bar{x})$ with $\bar{x}$ a tuple of variables and φ a conjunction of *concept atoms* $C(x)$ and *role atoms* $r(x, y)$ where C is a (possibly compound) $\mathcal{EL}$-concept, r is a role name, and x, y are variables from $\bar{x}$. We require that every variable from $\bar{x}$ occurs in some atom of q, but may omit this atom in writing in case it is $\top(x)$. We may write $\alpha \in q$ to indicate that α is an atom in φ. With $\mathsf{var}(q)$, we denote the variables in φ. The *arity* of q is the length of $\bar{x}$. We say that q is *connected* if the undirected graph with node set $\mathsf{var}(q)$ and edge set $\{\{v, v'\} \mid r(v, v') \in q \text{ for any } r \in \mathbf{R}\}$ is. Note that CQs as defined here do not admit quantified variables. The reason is that admitting such variables results in DLs with abstraction and refinement to become undecidable, even when based on $\mathcal{EL}$ (Lutz and Schulze 2023). In examples, we shall often write only $\varphi(\bar{x})$ in place of $q(\bar{x}) \leftarrow \varphi(\bar{x})$. We then choose a variable naming scheme such as x_1, x_2, x_3 that makes clear the order of the variables in $\bar{x}$ (and we then assume that there are no repeated variables in $\bar{x}$).

Let $q(\bar{x})$ be a CQ and $\mathcal{I}$ an interpretation. A mapping $h : \bar{x} \to \Delta^\mathcal{I}$ is a *homomorphism* from q to $\mathcal{I}$ if $C(x) \in q$ implies $h(x) \in C^\mathcal{I}$ and $r(x, y) \in q$ implies $(h(x), h(y)) \in r^\mathcal{I}$. A tuple $\bar{d} \in (\Delta^\mathcal{I})^{|\bar{x}|}$ is an *answer* to q on $\mathcal{I}$ if there is a homomorphism h from q to $\mathcal{I}$ with $h(\bar{x}) = \bar{d}$. We use $q(\mathcal{I})$ to denote the set of all answers to q on $\mathcal{I}$.

For any syntactic object O such as an ontology or a concept, we use $||O||$ to denote the *size* of O, that is, the number of symbols needed to write O using a suitable alphabet.

3 DLs with Abstraction and Refinement

We extend $\mathcal{ELH}_r$ to the DL $\mathcal{ELH}_r^{\mathsf{abs}}$ that supports abstraction and refinement, following (Lutz and Schulze 2023). Fix a countable set $\mathbf{A}$ of *abstraction levels*. An $\mathcal{ELH}_r^{\mathsf{abs}}$-ontology is a finite set of statements of the following form:

- *labeled concept inclusions* $C \sqsubseteq_L D$, role inclusions $r \sqsubseteq_L s$, and range restrictions $\top \sqsubseteq_L \forall r, C$,
- *concept refinements* $L{:}q(\bar{x})$ $\underline{\mathsf{refines}}$ $L'{:}C$,
- *concept abstractions* $L'{:}C$ $\underline{\mathsf{abstracts}}$ $L{:}q(\bar{x})$,
- *role refinements* $L{:}q(\bar{x}, \bar{y})$ $\underline{\mathsf{refines}}$ $L'{:}q_r(x, y)$,
- *role abstractions* $L'{:}r$ $\underline{\mathsf{abstracts}}$ $L{:}q(\bar{x}, \bar{y})$

where L, L' range over abstraction levels from $\mathbf{A}$, C, D over $\mathcal{EL}$-concepts, r over role names, q over conjunctive queries, and q_r over conjunctive queries of the form $C_1(x) \wedge r(x, y) \wedge C_2(y)$. In concept and role abstractions, we additionally require the CQ q to be connected.

We also consider various fragments of $\mathcal{ELH}_r^{\mathsf{abs}}$. With $\mathcal{ELH}_r^{\mathsf{abs}}$[cr,ca], for example, we mean the fragment of $\mathcal{ELH}_r^{\mathsf{abs}}$ that admits only concept refinement and concept abstraction, but neither role refinement nor role abstraction (which are identified by rr and ra). As in the base case, we drop $\mathcal{H}$ if no role inclusions are admitted and likewise for $\cdot_r$ and range restrictions.

We next define the semantics of $\mathcal{ELH}_r^{\mathsf{abs}}$, based on *A-interpretations* that include one traditional DL interpretation for each abstraction level. Formally, an A-interpretation takes the form $\mathcal{I} = (\mathbf{A}_\mathcal{I}, \prec, (\mathcal{I}_L)_{L \in \mathbf{A}_\mathcal{I}}, \rho)$, where

- $\mathbf{A}_\mathcal{I} \subseteq \mathbf{A}$ is the set of relevant abstraction levels;
- $\prec \subseteq \mathbf{A}_\mathcal{I} \times \mathbf{A}_\mathcal{I}$ is such that the directed graph $(\mathbf{A}_\mathcal{I}, \prec)$ is a tree;[1] intuitively, $L \prec L'$ means that L is less abstract than L' or, in other words, that the modeling granularity of L is finer than that of L';
- $(\mathcal{I}_L)_{L \in \mathbf{A}_\mathcal{I}}$ is a collection of interpretations $\mathcal{I}_L$, one for every $L \in \mathbf{A}_\mathcal{I}$, with pairwise disjoint domains; we use $L(d)$ to denote the unique $L \in \mathbf{A}_\mathcal{I}$ with $d \in \Delta^{\mathcal{I}_L}$;
- ρ is the *refinement function*, a partial function that associates pairs $(d, L) \in \Delta^\mathcal{I} \times \mathbf{A}_\mathcal{I}$ such that $L \prec L(d)$ with an *L-ensemble* $\rho(d, L)$, that is, with a non-empty tuple over $\Delta^{\mathcal{I}_L}$. We want every object to participate in only one ensemble and thus require that

(∗) for all $d \in \Delta^\mathcal{I}$ and $L \in \mathbf{A}_\mathcal{I}$, there is at most one $e \in \Delta^{\mathcal{I}_L}$ such that d occurs in $\rho(e, L(d))$.

For readability, we may write $\rho_L(d)$ in place of $\rho(d, L)$.

An A-interpretation $\mathcal{I} = (\mathbf{A}_\mathcal{I}, \prec, (\mathcal{I}_L)_{L \in \mathbf{A}_\mathcal{I}}, \rho)$ *satisfies*

- a labeled concept inclusion $C \sqsubseteq_L D$ if $\mathcal{I}_L$ satisfies $C \sqsubseteq D$, and likewise for role inclusions and range restrictions;
- $L{:}q(\bar{x})$ $\underline{\mathsf{refines}}$ $L'{:}C$ if $L \prec L'$ and for all $d \in C^{\mathcal{I}_{L'}}$, there is an $\bar{e} \in q(\mathcal{I}_L)$ such that $\rho_L(d) = \bar{e}$;
- $L'{:}C$ $\underline{\mathsf{abstracts}}$ $L{:}q(\bar{x})$ if $L \prec L'$ and for all $\bar{e} \in q(\mathcal{I}_L)$, there is a $d \in C^{\mathcal{I}_{L'}}$ such that $\rho_L(d) = \bar{e}$;
- $L{:}q(\bar{x}, \bar{y})$ $\underline{\mathsf{refines}}$ $L'{:}q_r(x, y)$ if $L \prec L'$ and for all $(d_1, d_2) \in q_r(\mathcal{I}_{L'})$, there is an $(\bar{e}_1, \bar{e}_2) \in q(\mathcal{I}_L)$ such that $\rho_L(d_1) = \bar{e}_1$ and $\rho_L(d_2) = \bar{e}_2$;
- $L'{:}r$ $\underline{\mathsf{abstracts}}$ $L{:}q(\bar{x}, \bar{y})$ if $L \prec L'$ and for all $(\bar{e}_1, \bar{e}_2) \in q(\mathcal{I}_L)$, there is a $(d_1, d_2) \in r^{\mathcal{I}_{L'}}$ such that $\rho_L(d_1) = \bar{e}_1$ and $\rho_L(d_2) = \bar{e}_2$.

An A-interpretation is a *model* of an $\mathcal{ELH}_r^{\mathsf{abs}}$-ontology if it satisfies all inclusions, refinements, etc in it.

Example 1. *We consider the domain of actions. Assume that there is a* $\mathsf{MealPrep}$ *action that refines into subactions:* $L_2{:}q_M$ $\underline{\mathsf{refines}}$ $L_1{:}\mathsf{MealPrep}$ *where*

$$q_M = \mathsf{Buying}(x_1) \wedge \mathsf{Cooking}(x_2) \wedge \mathsf{precedes}(x_1, x_2)$$

We might have budget-friendly meal preparation and buying actions:

$$\mathsf{BudgetMealPrep} \sqsubseteq_{L_1} \mathsf{MealPrep}$$
$$\mathsf{BudgetBuying} \sqsubseteq_{L_2} \mathsf{Buying}$$
$$\mathsf{BudgetBuying} \sqcap \exists \mathsf{bought}.\mathsf{Expensive} \sqsubseteq_{L_2} \bot$$

A budget-friendly meal preparation requires buying non-expensive ingredients: $L_2{:}q_B$ $\underline{\mathsf{refines}}$ $L_1{:}\mathsf{BudgetMealPrep}$ *where*

$$q_B(x_1, x_2) = \mathsf{BudgetBuying}(x_1).$$

We are interested in two reasoning problems: *concept satisfiability* and *concept subsumption*. Concept satisfiability means to decide, given an ontology $\mathcal{O}$, an $\mathcal{EL}$-concept C, and an abstraction level $L \in \mathbf{A}$, whether there is a model $\mathcal{I}$ of $\mathcal{O}$ such that $C^{\mathcal{I}_L} \neq \emptyset$. We then say that C is *L-satisfiable w.r.t.* $\mathcal{O}$ and call $\mathcal{I}$ an *L-model of C and $\mathcal{O}$*.

For concept subsumption, we are given an ontology $\mathcal{O}$, two concepts C and D, and an abstraction level $L \in \mathbf{A}$, and are asked to decide whether $C^{\mathcal{I}_L} \subseteq D^{\mathcal{I}_L}$ in every model $\mathcal{I}$ of $\mathcal{O}$. If this is the case we say that C *is L-subsumed by* D *w.r.t.* $\mathcal{O}$ and write $\mathcal{O} \models C \sqsubseteq_L D$.

We remark that the $\bot$-concept, interpreted as $\bot^\mathcal{I} = \emptyset$ in every interpretation $\mathcal{I}$, can be expressed in $\mathcal{EL}^{\mathsf{abs}}$[cr] at the expense of introducing fresh symbols: a CI $C \sqsubseteq_L \bot$ can be simulated by

$$L'{:}A(x) \ \underline{\mathsf{refines}}\ L{:}C \quad L'{:}r(x_1, x_2) \ \underline{\mathsf{refines}}\ L{:}C$$

where A, r, and L are a fresh concept name, role name, and abstraction level. This is because the two refinements require ensembles of different length. W.l.o.g., we thus use the $\bot$-concept whenever convenient.

Using $\bot$, (un)satisfiability and subsumption are easily interreducible in polynomial time. In fact, it is not hard to see that a concept C is L-unsatisfiable w.r.t. an ontology $\mathcal{O}$ iff C is L-subsumed by some fresh concept name A w.r.t. $\mathcal{O}$. Conversely, C is L-subsumed by D w.r.t. $\mathcal{O}$ iff C is L-unsatisfiable w.r.t. $\mathcal{O} \cup \{C \sqcap D \sqsubseteq \bot\}$. We thus state all our

[1]Dropping this restriction results in undecidability (Lutz and Schulze 2023).

results in terms of satisfiability and assume that it is understood that (up to complementation) they also apply to subsumption.

4 Upper Bounds

We prove upper complexity bounds for satisfiability in $\mathcal{ELH}_r^{\mathsf{abs}}$. A 2EXPTIME upper bound for full $\mathcal{ELH}_r^{\mathsf{abs}}$ follows from the results in (Lutz and Schulze 2023). We thus concentrate on the fragments $\mathcal{ELH}_r^{\mathsf{abs}}[\mathrm{cr}]$ and $\mathcal{ELH}_r^{\mathsf{abs}}[\mathrm{cr}, \mathrm{ca}]$.

4.1 Simplifying Assumptions

We discuss some assumptions, all w.l.o.g., made throughout Section 4. First, we assume that the input ontology $\mathcal{O}$ is in *normal form*, meaning that:

1. all the CIs in $\mathcal{O}$ are of one of the following forms, where $A, A_1, \ldots, A_n, B$ are concept *names*:

$$\top \sqsubseteq_L A \qquad A_1 \sqcap \cdots \sqcap A_n \sqsubseteq_L B$$
$$A \sqsubseteq_L \exists r.B \qquad \exists r.A \sqsubseteq_L B$$

2. all range restrictions and (concept and role) refinements and abstractions contain only concept names, but no compound concepts, also inside of CQs.

By introducing new concept and role names, any $\mathcal{ELH}_r^{\mathsf{abs}}$ ontology $\mathcal{O}$ can be converted into an ontology $\mathcal{O}'$ in normal form that is a conservative extension of $\mathcal{O}$, i.e., every model of $\mathcal{O}'$ is also a model of $\mathcal{O}$, and every model of $\mathcal{O}$ can be extended to a model of $\mathcal{O}'$ by appropriately choosing the interpretations of the concept names that have been introduced during the conversion. The conversion takes only linear time, see for example (Baader *et al.* 2005). We further assume that the concept C_0 whose satisfiability is to be decided is a concept name, thus not compound. Finally, we assume that the abstraction level L_0 for which satisfiability is to be decided is the root of the tree $G_\mathcal{O}$ that is defined by the abstractions and refinements in $\mathcal{O}$. Let us make the latter more precise.

We use $\mathbf{A}_\mathcal{O}$ to denote the set of abstraction levels mentioned in $\mathcal{O}$ and $\prec_\mathcal{O}$ for the smallest relation on $\mathbf{A}_\mathcal{O}$ such that $L \prec_\mathcal{O} L'$ if $\mathcal{O}$ contains a concept refinement $L{:}q(\bar{x})$ $\underline{\mathsf{refines}}$ $L'{:}C$ or a concept abstraction $L'{:}C$ $\underline{\mathsf{abstracts}}$ $L{:}q(\bar{x})$. The *abstraction graph of an ontology* $\mathcal{O}$ is the directed graph

$$G_\mathcal{O} = (\mathbf{A}_\mathcal{O}, \prec_\mathcal{O}^{-1}).$$

Note that by the definition of the semantics, $\mathcal{O}$ being satisfiable implies that $G_\mathcal{O}$ is a tree.

Now assume that the abstraction level L_0 for which satisfiability is to be decided is not the root of $G_\mathcal{O}$, but L_R is. Then $G_\mathcal{O}$ contains a path $L_R = \widehat{L}_1, \ldots, \widehat{L}_k = L_0$ and we can extend $\mathcal{O}$ with concept refinements $\widehat{L}_{i+1}{:}\, A(x)$ $\underline{\mathsf{refines}}$ $\widehat{L}_i{:}\, A$ and $\widehat{L}_k{:}\, C_0(x)$ $\underline{\mathsf{refines}}$ $\widehat{L}_{k-1}{:}\, A$ for $1 \leq i < k$, with A a fresh concept name, and decide L_R-satisfiability of A w.r.t. the extended ontology.

4.2 Universal Models and The Chase

A crucial property of description logics of the $\mathcal{EL}$ family is the existence of universal models, defined in terms of homomorphisms. In particular, universal models are at the basis of all polynomial time algorithms for description logic reasoning that we are aware of. A fundamental observation that underlies the design of our algorithms is that universal models also exist for $\mathcal{ELH}_r^{\mathsf{abs}}$.

Let $\mathcal{I}_i = (\mathbf{A}_{\mathcal{I}_i}, \prec_i, (\mathcal{I}_{L,i})_{L \in \mathbf{A}_{\mathcal{I}_i}}, \rho_i)$ be an A-interpretation, for $i \in \{1, 2\}$. A function $h\colon \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ is a *homomorphism* from $\mathcal{I}_1$ to $\mathcal{I}_2$ if the following conditions are satisfied, for all $d, e \in \Delta^{\mathcal{I}_1}$:

1. $L(d) = L(h(d))$;
2. $\prec_1 \subseteq \prec_2$;
3. $d \in A^{\mathcal{I}_1}$ implies $h(d) \in A^{\mathcal{I}_2}$ for all $A \in \mathbf{C}$;
4. $(d, e) \in r^{\mathcal{I}_1}$ implies $(h(d), h(e)) \in r^{\mathcal{I}_2}$ for all $r \in \mathbf{R}$;
5. $\rho_1(d, L) = \bar{e}$ implies $\rho_2(h(d), L) = h(\bar{e})$

where $h(\bar{e})$ is the tuple obtained from $\bar{e}$ by applying h component-wise. Note that this implies $\mathbf{A}_{\mathcal{I}_1} \subseteq \mathbf{A}_{\mathcal{I}_2}$.

Let C_0 be an $\mathcal{EL}$-concept, $\mathcal{O}$ an $\mathcal{ELH}_r^{\mathsf{abs}}$-ontology, and $L_0 \in \mathbf{A}$ an abstraction level. A model $\mathcal{I}$ of $\mathcal{O}$ with distinguished element $d \in C_0^\mathcal{I}$, where $L(d) = L_0$, is a *universal L_0-model of C_0 and $\mathcal{O}$* if the following holds: for every model $\mathcal{J}$ of $\mathcal{O}$ and every $e \in C_0^\mathcal{J}$ with $L(e) = L_0$, there exists a homomorphism h from $\mathcal{I}$ to $\mathcal{J}$ with $h(d) = e$. Our aim is to show the following.

Lemma 1. *Let C_0 be an $\mathcal{EL}$-concept, $\mathcal{O}$ an $\mathcal{ELH}_r^{\mathsf{abs}}$-ontology, and $L_0 \in \mathbf{A}$. If C_0 is L_0-satisfiable w.r.t. $\mathcal{O}$, then there exists a universal L_0-model of C_0 and $\mathcal{O}$.*

Lemma 1 is proved by a somewhat intricate chase procedure. For technical reasons, this chase may construct structures that do not satisfy all the conditions required of A-interpretations. The chase does thus not run directly in A-interpretations, but rather on a weakening that we call interpretation candidates.

Let $\mathbf{K}$ be a countably infinite set of *constants*. A *fact* is an expression of the form $A(a)$ or $r(a, b)$ where A is a concept name, r a role name, and a, b are constants. Homomorphisms from conjunctive queries to sets of facts are defined in the expected way. An *interpretation candidate* is a triple $I = (F, \rho, \sim)$ where

- F is a *fact assignment*, that is, a function that maps each abstraction level $L \in \mathbf{A}_\mathcal{O}$ to a set of facts $F(L)$. We use $\mathsf{dom}(F(L))$ to denote the *domain* of $F(L)$, that is, $\mathsf{dom}(F(L)) = \{a \in \mathbf{K} \mid a \text{ is used in a fact in } F(L)\}$. We demand that the $F(L)$ have pairwise disjoint domains and may write $\mathsf{dom}(F)$ to denote $\biguplus_{L \in \mathbf{A}_\mathcal{O}} \mathsf{dom}(F_L)$. We further use $L(a)$, for any $a \in \mathsf{dom}(F)$, to denote the unique $L \in \mathbf{A}_\mathcal{O}$ such that $a \in \mathsf{dom}(F(L))$;
- ρ is a *refinement function*, that is, a partial function that associates pairs $(a, L) \in \mathsf{dom}(F) \times \mathbf{A}_\mathcal{O}$ such that $L \prec L(a)$ with an *L-ensemble* $\rho(a, L)$, that is, with a non-empty tuple over $\mathsf{dom}(F(L))$;

- $\sim$ is an *equivalence relation* on the set $\mathsf{dom}(F)$. If we *set* $a_1 \sim a_2$, we mean to set $\sim := \sim \cup \{(a_1, a_2)\}$ and add the smallest number of tuples such that $\sim$ is again an equivalence relation. We use $[a]$ to denote the *equivalence class* of $a \in \mathsf{dom}(F)$ w.r.t. $\sim$.

For readability we may write F_L instead of $F(L)$ and $\rho_L(a)$ instead of $\rho(a, L)$. Note that ρ, in contrast to the refinement function in A-interpretations, allows elements to be part of multiple ensembles.

Our chase procedure starts from the *initial interpretation candidate* $I_0 = (F^0, \rho^0, \sim^0)$ for C_0, L_0, and $\mathcal{O}$ where $F^0_L = \{A_\top(a_L)\}$ for all $L \in \mathbf{A}_\mathcal{O} \setminus \{L_0\}$, $F^0_{L_0} = \{C_0(a_0)\}$, ρ^0 is empty, and $\sim^0$ is the identity. It then applies a set of rules in a fair way, that is, every rule that is applicable will eventually be applied. The chase may also abort and report unsatisfiability of the input. We start by giving the rules that treat inclusions and range restrictions:

R1 if $A_1(a) \in F_L, \dots, A_n(a) \in F_L$, and $A_1 \sqcap \dots \sqcap A_n \sqsubseteq_L B \in \mathcal{O}$, then add $B(a)$ to F_L;

R2 if $a \in \mathsf{dom}(F_L)$ and $\top \sqsubseteq_L A \in \mathcal{O}$, then add $A(a)$ to F_L;

R3 if $A(a) \in F_L$, $A \sqsubseteq_L \exists r.B \in \mathcal{O}$, then add $r(a, b)$ and $B(b)$ to F_L with b as a fresh constant;

R4 if $r(a, b) \in F_L$, $A(b) \in F_L$, and $\exists r.A \sqsubseteq_L B \in \mathcal{O}$, then add $B(a)$ to F_L.

R5 if $r(a, b) \in F_L$ and $r \sqsubseteq_L s \in \mathcal{O}$, then add $s(a, b)$ to F_L;

R6 if $r(a, b) \in F_L$ and $\top \sqsubseteq_L \forall r.C \in \mathcal{O}$, then add $C(a)$ to F_L;

Next up are the rules that pertain to concept refinements and abstractions in $\mathcal{O}$. We may use $x \in \bar{x}$ to express that variable x occurs in the tuple $\bar{x}$. For a CQ $q(\bar{x})$ and a tuple of constants $\bar{a}$ with $|\bar{x}| = |\bar{a}|$, we use $q(\bar{a})$ to denote the set of facts obtained from q by replacing in every atom the i-th variable in $\bar{x}$ by the i-th constant in $\bar{a}$, for $1 \leq i \leq |\bar{a}|$.

R7 if $A(a) \in F_L$, $L'{:}\, q(\bar{x})$ $\underline{\mathsf{refines}}$ $L{:}\, A \in \mathcal{O}$, and $\rho_{L'}(a)$ is undefined, then set $\rho_{L'}(a) = \bar{a}$ for a tuple $\bar{a}$ of fresh constants with $|\bar{a}| = |\bar{x}|$;

R8 if $A(a) \in F_L$, $L'{:}\, q(\bar{x})$ $\underline{\mathsf{refines}}$ $L{:}\, A \in \mathcal{O}$, $\rho_{L'}(a)$ is defined, and $|\bar{x}| = |\rho_{L'}(a)|$, then add $q(\rho_{L'}(a))$ to $F_{L'}$; if $|\bar{x}| \neq |\rho_{L'}(a)|$, then return 'unsatisfiable';

R9 if h is a homomorphism from q to F_L for any concept abstraction $L'{:}\, A$ $\underline{\mathsf{abstracts}}$ $L{:}\, q(\bar{x}) \in \mathcal{O}$ and there is no $a \in \mathsf{dom}(F_{L'})$ with $\rho_L(a) = h(\bar{x})$, then introduce a fresh constant a and set $\rho_L(a) = h(\bar{x})$;

R10 if h is a homomorphism from q to F_L for any concept abstraction $L'{:}\, A$ $\underline{\mathsf{abstracts}}$ $L{:}\, q(\bar{x}) \in \mathcal{O}$ and there is an $a \in \mathsf{dom}(F_{L'})$ with $\rho_L(a) = h(\bar{x})$, then add $A(a)$ to $F_{L'}$.

There are analogous rules for role refinement and role abstraction, given in the appendix. We also have rules that concern overlapping ensembles. Intuitively, overlapping ensembles require the identification of elements, but we do not want to do this in the chase itself to preserve monotonicity, that is, rule applications should always extend the interpretation candidate. We thus only record the necessary identifications in the '$\sim$' component of interpretation candidates.

R15 if $\rho_L(a_1) = \bar{e}_1$, $\rho_L(a_2) = \bar{e}_2$, there are $b_1 \in \bar{e}_1$ and $b_2 \in \bar{e}_2$ with $b_1 \sim b_2$ and $|\bar{e}_1| \neq |\bar{e}_2|$, then return 'unsatisfiable';

R16 if there are $b_1 \in \rho_L(a_1)$ and $b_2 \in \rho_L(a_2)$ with $b_1 \sim b_2$, then set $a_1 \sim a_2$;

R17 if $a_1 \sim a_2$, $\rho_L(a_1) = \bar{e}_1$, and $\rho_L(a_2) = \bar{e}_2$ with $|\bar{e}_1| = |\bar{e}_2|$, then set $\bar{e}_1[i] \sim \bar{e}_2[i]$ for $1 \leq i \leq |\bar{e}_1|$;

R18 if $a_1 \sim a_2$ and fact $f \in F_L$ contains constant a_1, then add to F_L the fact obtained from f by replacing some occurrence of a_1 with a_2.

R19 if $a_1 \sim a_2$, $\rho_L(a_1)$ is defined, and $\rho_L(a_2)$ is undefined, then add to F_L facts $A_\top(b_1), \dots, A_\top(b_n)$, with $b_1, \dots, b_n$ fresh constants and $n = |\rho_L(a_1)|$, and set $\rho_L(a_2) = (b_1, \dots, b_n)$ (where $A_\top$ is a fresh concept name).

A *chase sequence* is a sequence of interpretation candidates $I_0, I_1, \dots$ such that $I_0 = (F^0, \rho^0, \sim^0)$, I_{i+1} is obtained from I_i by applying one of the rules defined above. Every chase sequence $I_0, I_1, \dots$ gives rise to an interpretation candidate $I^* = (F^*, \rho^*, \sim^*)$ in the limit, with $F^* = \bigcup_i F_i$, $\rho^* = \bigcup_i \rho_i$, and $\sim^* = \bigcup_i \sim^i$. We also call I^* the *result* of chasing C_0 w.r.t. L_0 and $\mathcal{O}$. It can be shown that, up to isomorphism, all fair chase sequences deliver the same result.[2]

The chase is sound and complete in the following sense.

Lemma 2. *Let $\mathcal{O}$ be an $\mathcal{ELH}^{\mathsf{abs}}_r$-ontology in normal form whose abstraction graph $G_\mathcal{O}$ is a tree, C_0 a concept name, and L_0 an abstraction level. The L_0-chase on C_0 and $\mathcal{O}$ does not abort if and only if C_0 is L_0-satisfiable w.r.t. $\mathcal{O}$.*

In the proof of the 'only if' direction of Lemma 2 (soundness), we start from a non-aborting chase sequence that delivers a result $I^* = (F^*, \rho^*, \sim^*)$, and then construct from I^* an L_0-model $\mathcal{I}$ of C_0 and $\mathcal{O}$. Intuitively, we apply filtration to make the equalities recorded in $\sim^*$ real equalities. This is achieved by setting $\mathcal{I} = (\mathbf{A}_\mathcal{I}, \prec, (\mathcal{I}_L)_{L \in \mathbf{A}_\mathcal{I}}, \rho)$ where

$$\begin{aligned}
\Delta^{\mathcal{I}_L} &= \{[a] \mid a \in \mathsf{dom}(F^*_L)\}\\
A^{\mathcal{I}_L} &= \{[a] \mid A(a') \in F^*_L \text{ and } a' \in [a]\}\\
r^{\mathcal{I}_L} &= \{([a],[b]) \mid r(a', b') \in F^*_L \text{ and } a' \in [a], b' \in [b]\}\\
\rho_L &= \{([a],([b_1]\cdots[b_n])) \mid (a',(b'_1\cdots b'_n)) \in \rho^*_L \text{ with}\\
&\qquad a' \in [a], b'_i \in [b_i] \text{ for } 1 \leq i \leq n\}.
\end{aligned}$$

The remaining components $\mathbf{A}_\mathcal{I}$ and $\prec$ are defined as $\mathbf{A}_\mathcal{O}$ and $\prec_\mathcal{O}$, respectively. We show in the appendix that $\mathcal{I}$ is not only an L_0-model of C_0 and $\mathcal{O}$, but even a universal such model, thus proving Lemma 1.

4.3 $\mathcal{ELH}^{\mathsf{abs}}_r$[cr] in coNP

Our aim is to prove the following.

Theorem 1. *Satisfiability in $\mathcal{ELH}^{\mathsf{abs}}_r$[cr] is in* coNP.

[2]Note that our rule R3 is oblivious in the sense that it may always add a fresh constant b even if there is already a b' with $r(a, b')$ and $B(b')$ in F_L.

It suffices to find an NP algorithm for unsatisfiability. Assume that the concept name C_0, the $\mathcal{ELH}_r^{\mathsf{abs}}[\text{cr}]$-ontology $\mathcal{O}$, and the abstraction level L_0 are given as an input, that is, we want to decide whether C_0 is L_0-unsatisfiable w.r.t. $\mathcal{O}$. If the abstraction graph $G_\mathcal{O}$ of $\mathcal{O}$ is not a tree, we directly return 'unsatisfiable'. Otherwise, the only remaining way in which unsatisfiability may arise is that there are two refinement statements that both apply to the same element of a model, but require ensembles of different length.

Example 2. *Consider the following ontology $\mathcal{O}$:*

$$C_0 \sqsubseteq \exists r.A_1$$
$$L_1\colon A_2(x) \ \underline{\mathsf{refines}}\ L_0\colon A_1$$
$$A_2 \sqsubseteq \exists s.A_3$$
$$L_2\colon B(x) \ \underline{\mathsf{refines}}\ L_1\colon A_3$$
$$L_2\colon r(x_1, x_2) \ \underline{\mathsf{refines}}\ L_1\colon A_3$$

The reader may try to construct an L_0-model of C_0 and $\mathcal{O}$, following the sequence of existential quantifications and refinements suggested by the order of the statements in $\mathcal{O}$.

As suggested by Example 2, our algorithm guesses a sequence of existential quantifications and refinements that lead to two 'incompatible' refinements. To make this precise, we need some preliminaries.

We use u to denote the *universal role*, that is, a fixed role name that is always interpreted as $\mathsf{u}^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. An *ABox* is a finite set of facts as defined in the previous section. An interpretation $\mathcal{I}$ *satisfies* a concept assertion $C(a)$ if $a \in C^\mathcal{I}$, a role assertion $r(a, b)$ if $(a, b) \in r^\mathcal{I}$, and an ABox $\mathcal{A}$ if it satisfies all concept and role assertions in it. For an ABox $\mathcal{A}$, an $\mathcal{ELH}_r$-ontology $\mathcal{O}$, and an $\mathcal{EL}$-concept C, we write $\mathcal{A}, \mathcal{O} \models \exists \mathsf{u}.C$ if $C^\mathcal{I} \neq \emptyset$ in every model $\mathcal{I}$ of $\mathcal{O}$ that satisfies $\mathcal{A}$. It is known that given $\mathcal{A}$, $\mathcal{O}$, and C, it can be decided in polynomial time whether $\mathcal{A}, \mathcal{O} \models \exists \mathsf{u}.C$ (Krötzsch 2010). Note that a conjunctive query can be viewed as an ABox in an obvious way, by viewing variables as constants.

For a set of concept names S and $L, L' \in \mathbf{A}_\mathcal{O}$, we use $Q^{\mathsf{ref}}_{L,L'}(S)$ to denote the set of CQs $q(\bar{x})$ such that $\mathcal{O}$ contains a concept refinement L':$q(\bar{x})$ $\underline{\mathsf{refines}}$ L:C with $C \in S$. We assume that the conjunctive queries $q(\bar{x})$ in concept refinements in $\mathcal{O}$ use canonical variable names, that is, the variable with the left-most occurrence in $\bar{x}$ is x_1, the variable that occurs next is x_2, etc.

For an abstraction level $L \in \mathbf{A}_\mathcal{O}$, we use $\mathcal{O}_L$ to denote the $\mathcal{ELH}_r$-ontology that consists of all concept inclusions $C \sqsubseteq D$ such that $C \sqsubseteq_L D \in \mathcal{O}$, all role inclusions $r \sqsubseteq s$ such that $r \sqsubseteq_L s \in \mathcal{O}$, and all range restrictions $\top \sqsubseteq \forall r.C$ such that $\top \sqsubseteq_L \forall r.C \in \mathcal{O}$.

We are now ready to describe the algorithm. It guesses a sequence $S_1, L_1, \dots, S_n, L_n$ where $S_1, \dots, S_n$ are sets of concept names that occur in $\mathcal{O}$ and $L_1, \dots, L_n$ are abstraction levels, $n \leq |\mathbf{A}_\mathcal{O}|$. It accepts if the following conditions are satisfied, and rejects otherwise:

1. one of the following holds:
   - $L_1 = L_0$ and $\mathcal{O}_{L_1} \models C_0 \sqsubseteq_{L_1} \exists \mathsf{u}.(\bigsqcap S_1)$ or
   - $\mathcal{O}_{L_1} \models \top \sqsubseteq_{L_1} \exists \mathsf{u}.(\bigsqcap S_1)$;
2. for $1 \leq i < n$:
   $$\mathcal{A}_i, \mathcal{O}_{L_i} \models \exists \mathsf{u}.(\bigsqcap S_{i+1})$$
   where $\mathcal{A}_i$ is the union of all queries in $Q^{\mathsf{ref}}_{L_i,L_{i+1}}(S_i)$, viewed as ABoxes.[3]
3. There are concept refinements $L'\colon q(\bar{x})$ $\underline{\mathsf{refines}}$ $L\colon A \in \mathcal{O}$ and $L'\colon q'(\bar{x}')$ $\underline{\mathsf{refines}}$ $L\colon B \in \mathcal{O}$ such that $A, B \in S_n$ and $|\bar{x}| \neq |\bar{x}'|$.

Note that in Example 2, we have always interleaved a single existential restriction with each refinement statement. In general, however, there can be a more complex '$\mathcal{EL}$-derivation' between two subsequent refinements, and we abstract away from that by using the universal role.

Lemma 3. *The algorithm accepts iff C_0 is L_0-unsatisfiable w.r.t. $\mathcal{O}$.*

The proof of Lemma 3 crucially uses universal models as produced by the chase procedure from Section 4.2.

Note that, by what was said above, Conditions 1 to 3 can be checked in polynomial time. We have thus obtained an NP algorithm, as desired.

4.4 $\mathcal{ELH}_r^{\mathsf{abs}}[\text{cr}, \text{ca}]$ in PSPACE

We now add concept abstraction, that is, we move from $\mathcal{ELH}_r^{\mathsf{abs}}[\text{cr}]$ to $\mathcal{ELH}_r^{\mathsf{abs}}[\text{cr}, \text{ca}]$. This makes a significant difference because now we can also pass information upwards through the tree-shaped abstraction graph of the ontology, as illustrated by the following example.

Example 3. *Consider the following ontology $\mathcal{O}$:*

$$L_1{:}A_1(x)\ \underline{\mathsf{refines}}\ L_0{:}C_0 \qquad L_2{:}A_2(x)\ \underline{\mathsf{refines}}\ L_0{:}C_0$$
$$A_1 \sqsubseteq B_1 \qquad A_2 \sqsubseteq B_2$$
$$L_0{:}C_1\ \underline{\mathsf{abstracts}}\ L_1{:}B_1(x) \qquad L_0{:}C_2\ \underline{\mathsf{abstracts}}\ L_2{:}B_2(x)$$
$$C_1 \sqcap C_2 \sqsubseteq \bot$$

C_0 is L_0-unsatisfiable w.r.t. $\mathcal{O}$, but there is no (linear!) sequence of existential quantifications and refinements as in Example 2.

We want to prove the following, which is substantially more difficult than proving the coNP upper bound in the previous section. In fact, we view the following as a main result of this paper.

Theorem 2. *Satisfiability in $\mathcal{ELH}_r^{\mathsf{abs}}[\text{cr}, \text{ca}]$ is in* PSPACE.

Let a concept name C_0, an $\mathcal{ELH}_r^{\mathsf{abs}}[\text{cr}, \text{ca}]$-ontology $\mathcal{O}$, and an abstraction level $L_0 \in \mathbf{A}_\mathcal{O}$ be given as an input. If the abstraction graph of $\mathcal{O}$ is not a tree, we immediately return 'unsatisfiable'.

Our algorithm has some resemblance with the standard non-deterministic PSPACE algorithm for the satisfiability of $\mathcal{ALC}$ concepts (without ontologies) that verifies the existence of a tree model of polynomial depth by traversing it in a depth-first manner, always keeping only a single path in memory (Baader *et al.* 2017). In our case, we want to verify the existence of an A-interpretation $\mathcal{I}$ that is an L_0-model of C_0 and $\mathcal{O}$. The tree that our algorithm traverses is

[3] Here we rely on canonical variable names.

$(\mathbf{A}_\mathcal{I}, \prec^{-1})$, which we can w.l.o.g. assume to be the abstraction graph of $\mathcal{O}$ (since universal models constructed by the chase have this property).

We are, however, confronted with two challenges. First, the 'upwards' nature of abstractions makes it difficult to traverse the tree in a depth-first manner. We address this by a suitable guessing strategy. And second, the interpretations $\mathcal{I}_L$ of a universal model, which correspond to the nodes of the traversed tree, are infinite and thus cannot be guessed. While infinite but regularly-shaped models can often be substituted by 'compact' finite models of polynomial size when designing algorithms for plain $\mathcal{EL}$ (Lutz *et al.* 2009), this is no longer true in the presence of CQs. To address this, we stick with infinite models $\mathcal{I}_L$, but *represent* them by compact (finite!) interpretations that we call pseudo-models. We then use a non-standard semantics for CQs on those compact representations.

Pseudo-Models For the following, one should imagine the interpretations $\mathcal{I}_L$ to take the shape of an infinite tree whose nodes are ensembles and domain elements that do not participate in an ensemble. In a pseudo-model $\mathcal{I}$, intuitively we identify ensembles / non-ensemble elements that are isomorphic, thus obtaining finiteness but losing the tree-shape.

Let $q(\bar{x})$ be a CQ and $\mathcal{I}$ an $\mathcal{EL}$-interpretation (representing the pseudo-model). Recall that in Section 2, we had associated an undirected graph G_q with q. We assume that $\mathcal{I}$ is equipped with a set of ensembles. Let h be a homomorphism from q to $\mathcal{I}$. We aim to identify a condition on h ('tameness') that allows us to obtain from h a homomorphism into the interpretation obtained by unraveling the pseudo-model $\mathcal{I}$ into an infinite tree-like interpretation.

We associate with h an equivalence relation $\sim_h$ on $\mathsf{var}(q)$ by setting $x \sim_h y$ if G_q contains a path $x = z_1, \ldots, z_n = y$ such that $h(z_1), \ldots, h(z_n)$ are all part of the same ensemble in $\mathcal{I}$. Let $G_{h,\mathcal{I}}$ be the directed graph whose nodes are the equivalence classes of $\sim_h$ and which has an edge (c_1, c_2) if there is an $r(x_1, x_2) \in q$ with $x_1 \in c_1$ and $x_2 \in c_2$. A node c of $G_{h,\mathcal{I}}$ is an *ensemble node* if there is an (equivalently: for all) $x \in c$ such that $h(x)$ is part of an ensemble in $\mathcal{I}$. We recommend to the reader to verify that all nodes that are not ensemble nodes are singleton classes. We say that h is *tame* if the following conditions are satisfied:

1. $G_{h,\mathcal{I}}$ is a tree, possibly with self-loops on ensemble nodes;
2. for all edges (c_1, c_2) in $G_{h,\mathcal{I}}$, there are $d_1, d_2 \in \Delta^\mathcal{I}$ such that for all $r(x_1, x_2) \in q$ with $x_1 \in c_1$ and $x_2 \in c_2$, we have $h(x_1) = d_1$ and $h(x_2) = d_2$.

Condition 2 reflects the fact that the tree-like interpretations $\mathcal{I}_L$ satisfy the following property: if d_i is an element in ensemble e_i, for $i \in \{1, 2\}$, and there is a role edge $(d_1, d_2) \in r^{\mathcal{I}_L}$, then d_1, d_2 are *unique* with this property.

An answer $\bar{d} \in q(\mathcal{I})$ is *tame* if there is a tame homomorphism h from $q(\bar{x})$ to $\mathcal{I}_L$ with $h(\bar{x}) = \bar{d}$. An A-interpretation $\mathcal{I}$ being a *pseudo-model* of $\mathcal{O}$ is defined in the same way as being a model of $\mathcal{O}$ except that in the semantics of concept abstractions, answers to a CQ q on an interpretation $\mathcal{I}_L$ are replaced with tame answers.

A central observation underlying the subsequent algorithm is that we can always find pseudo-models in which each maximal connected component has size polynomial in $||\mathcal{O}||$. Formally, a *maximal connected component (MCC)* of an A-interpretation $\mathcal{I}$ is an $\mathcal{EL}$-interpretation that can be obtained as follows: choose an abstraction level L, then choose a maximal subset $\Delta \subseteq \Delta^{\mathcal{I}_L}$ such that the following undirected graph is connected:

$$(\Delta, \{\{d, e\} \mid (d, e) \in r^{\mathcal{I}_L} \text{ for some role name } r \text{ or } d, e \text{ in } \bar{e} \text{ for some } L\text{-ensemble } \bar{e}\});$$

and finally take the restriction of $\Delta^{\mathcal{I}_L}$ to domain Δ. In the appendix, we prove the following.

Lemma 4. *If C_0 is L_0-satisfiable w.r.t. $\mathcal{O}$, then there is an L_0-pseudo-model $\mathcal{I}$ of C_0 and $\mathcal{O}$ such that each MCC of $\mathcal{I}$ has at most $2 \cdot (||\mathcal{O}||^2 + ||\mathcal{O}||)$ elements.*

Our proof of Lemma 4 is rather laborious. The reason is that the structure of the universal models as constructed in Section 4.2 turns out to be surprisingly hard (and, to us, actually infeasible) to analyze. This is mainly due to the application of the filtration construction after chase termination. To avoid such an analysis, we first introduce another, more semantic construction of universal models. In this construction, we start from the universal models from Section 4.2 and 'combine small pieces of them' in a uniform, tree-like way. The structure of the resulting universal models, which we call *uniform*, is clear by definition. In particular, each $\mathcal{EL}$-interpretation $\mathcal{I}_L$ is a tree of ensembles / non-ensemble nodes, as described above. Starting from uniform universal models, we can then carefully craft pseudo-models by selecting ensembles and non-ensemble elements and 'rerouting' role edges.

The Algorithm The aim of our algorithm is to verify the existence of a pseudo-model of C_0 and $\mathcal{O}$, as per Lemma 4. To represent MCCs of that pseudo-model, we use mosaics. A *mosaic* is a tuple $M = (\mathcal{I}, L, E, \bar{e})$ that consists of

1. a model $\mathcal{I}$ of $\mathcal{O}_L$ such that $|\Delta^\mathcal{I}| \leq 2 \cdot (||\mathcal{O}||^2 + ||\mathcal{O}||)$,
2. an abstraction level $L \in \mathbf{A}_\mathcal{O}$,
3. a set E of non-overlapping ensembles, that is, non-empty tuples over $\Delta^\mathcal{I}$ that do not share elements, and
4. a tuple $\bar{e}$ over $\Delta^\mathcal{I}$ with $\bar{e} \in E$ or $\bar{e} = ()$.

We may write $\mathcal{I}^M$ for $\mathcal{I}$, and likewise for L^M, E^M, and $\bar{e}^M$. We further define a function

- $Q^{\mathsf{ref}}_{M,L'}$ that maps each $d \in \Delta^\mathcal{I}$ to the set of CQs $Q^{\mathsf{ref}}_{M,L'}(d) = \{q \mid L'{:}q(\bar{x}) \text{ \underline{refines} } L{:}A \in \mathcal{O} \text{ and } d \in A^\mathcal{I}\}$;
- $T^{\mathsf{abs}}_{M,L'}$ that maps each $\bar{d} \in E$ to the set of concept names $T^{\mathsf{abs}}_{M,L'}(\bar{d}) = \{A \mid L'{:}A \text{ \underline{abstracts} } L{:}q(\bar{x}) \in \mathcal{O} \text{ and } \bar{d} \in q(\mathcal{I}) \text{ is tame}\}$.

Note that, as mosaics are equipped with an explicit set E of ensembles, it is clear what we mean by a tame answer.

Our algorithm is now listed as Algorithm 1. In Line 3, we guess a set X_L of sets of concept names. This is related to the first challenge mentioned above and the idea is that for

Algorithm 1 Algorithm for satisfiability in $\mathcal{EL}^{\mathsf{abs}}$[cr,ca]

procedure $\mathcal{EL}$[cr,ca]-SAT(C_0, L_0)
  for all $L \in \mathbf{A}_\mathcal{O}$ **do**
    Guess a set $X_L \in 2^{2^{\mathbf{C}}}$ of sets concept names such that $|X_L| \leq ||\mathcal{O}||^3 + ||\mathcal{O}||^2$
  Guess a mosaic M such that $L^M = L_0$, $C_0^{\mathcal{I}^M} \neq \emptyset$, and $e^M = ()$
  $R \leftarrow$ RECURSE(M)
  for all $L \in \mathbf{A}_\mathcal{O}$ and $T \in X_L$ **do**
    Guess a mosaic M such that $L^M = L$, $(\bigsqcap T)^{\mathcal{I}^M} \neq \emptyset$, and $e^M = ()$
    $R \leftarrow R \wedge$ RECURSE(M)
  return R
procedure RECURSE$(M = (\mathcal{I}, L, E, \bar{e}))$
  for all $d \in \Delta^\mathcal{I}$ and $L' \in \mathbf{A}_\mathcal{O}$ s.t. $Q^{\mathsf{ref}}_{M,L'}(d) \neq \emptyset$ **do**
    Guess a mosaic $M' = (\mathcal{I}', L', E', \bar{e}')$
    if $\bar{e}' \notin q(\mathcal{I}')$ for some $q \in Q^{\mathsf{ref}}_{M,L'}(d)$ or $d \notin A^\mathcal{I}$ for some $A \in T^{\mathsf{abs}}_L(\bar{e}')$ **then return** false
    RECURSE(M')
  for all $L'{:}A$ abstracts $L{:}q(\bar{x}) \in \mathcal{O}$ and all tame answers $\bar{d} \in q(\mathcal{I})$ with $\bar{d} \neq \bar{e}$ **do**
    if $\bar{d} \notin E$ **then return** false
    Guess a set $T' \in X_{L'}$
    if $T^{\mathsf{abs}}_{M,L'}(\bar{d}) \not\subseteq T'$ or $\bar{d} \notin q(\mathcal{I})$ for some $L'{:}q(\bar{x})$ refines $L{:}A \in \mathcal{O}$ with $A \in T'$ **then return** false
  return true

every set $S \in X_L$, there must be an element on level L that satisfies all concept names in S, and that (copies of) these elements can be used to satisfy all abstractions that ever require a witness during the run of the algorithm. Intuitively, the algorithm repeatedly guesses mosaics and makes recursive calls to satisfy refinement statements from the ontology. More precisely, it is the tuple $\bar{e}^M$ in the fourth component of a mosaic M that, if not empty, is the ensemble which satisfies the refinement.

Lemma 5. *The algorithm accepts iff C_0 is L_0-satisfiable w.r.t. $\mathcal{O}$.*

It is easy to see that the recursion depth of the algorithm is bounded by $|\mathbf{A}_\mathcal{O}|$ and that only a polynomial amount of space is consumed. Needless to say, we can eliminate non-determinism by applying Savitch's theorem.

5 Lower Bounds

We prove lower complexity bounds that match the upper bounds presented in Section 4. We start with the following.

Theorem 3. *Satisfiability is*

1. coNP-*hard in* $\mathcal{EL}^{\mathsf{abs}}$[cr] *and*
2. PSPACE-*hard in* $\mathcal{EL}^{\mathsf{abs}}$[cr, ca].

The proofs of Points 1 and 2 of Theorem 3 are closely related. We start with Point 1, which is proved by reduction from unsatisfiability in propositional logic.

Let φ be a propositional formula that uses variables $p_1, \dots, p_n$ and only the junctors $\neg$ and $\wedge$. Let $\mathsf{sub}(\varphi)$ be the set of all subformulas of φ. We construct an $\mathcal{EL}^{\mathsf{abs}}$[cr]-ontology $\mathcal{O}$ that uses the following concept and role names:

- T_ψ and F_ψ, for each $\psi \in \mathsf{sub}(\varphi)$, to represent that ψ evaluates to true or false;
- P_i and $\overline{P}_i$, for $i \in \{1, \dots, n\}$, to represent assigning true or false to p_i.

The ontology $\mathcal{O}$ uses the abstraction levels $L_0 \succ \cdots \succ L_n$. When refining from L_i to L_{i+1}, we introduce two domain elements that represent the two possible truth assignments for variable p_{i+1}. This is achieved by including in $\mathcal{O}$ the following for $1 \leq i \leq n$:

$$L_i{:}\, P_i(x_1) \wedge \overline{P}_i(x_2) \text{ } \underline{\mathsf{refines}} \text{ } L_{i-1}{:}\, \top. \tag{1}$$

If desired, it is easy to make the query connected. To preserve the truth assignments to variables on finer levels, we add for $1 \leq i < n$ and $i < j < n$:

$$L_{j+1}{:}P_i(x_1) \wedge P_i(x_2) \text{ } \underline{\mathsf{refines}} \text{ } L_j{:}P_i \tag{2}$$

$$L_{j+1}{:}\overline{P}_i(x_1) \wedge \overline{P}_i(x_2) \text{ } \underline{\mathsf{refines}} \text{ } L_j{:}\overline{P}_i \tag{3}$$

This generates a binary tree of refinements of depth n, representing all possible truth assignments at the leaves, that is, by domain elements on level L_n. We evaluate φ on all these truth assignments and generate an inconsistency if φ ever evaluates to true:

$$P_i \sqsubseteq_{L_n} T_{p_i} \text{ and } \overline{P}_i \sqsubseteq_{L_n} F_{p_i} \text{ for } 1 \leq i \leq n \tag{4}$$

$$T_\psi \sqsubseteq_{L_n} F_{\neg\psi} \text{ and } F_\psi \sqsubseteq_{L_n} T_{\neg\psi} \text{ for all } \neg\psi \in \mathsf{sub}(\varphi) \tag{5}$$

and for all $\psi = \psi_1 \wedge \psi_2 \in \mathsf{sub}(\varphi)$:

$$T_{\psi_1} \sqcap T_{\psi_2} \sqsubseteq_{L_n} T_\psi \quad F_{\psi_1} \sqsubseteq_{L_n} F_\psi \quad F_{\psi_2} \sqsubseteq_{L_n} F_\psi \tag{6}$$

and finally:

$$T_\varphi \sqsubseteq_{L_n} \bot. \tag{7}$$

Lemma 6. *φ is unsatisfiable iff $\top$ is L_0-satisfiable w.r.t. $\mathcal{O}$.*

Point 2 of Theorem 3 is proved by reduction from the validity of quantified Boolean formulas (QBFs) of the form $\varphi_0 = Q_1 p_1 \cdots Q_n p_n \varphi$ with $Q_i \in \{\exists, \forall\}$ and φ a propositional formula that uses only the variables p_1 to p_n and the junctors $\neg$ and $\wedge$ (Arora and Barak 2009). We construct an $\mathcal{EL}^{\mathsf{abs}}$[cr, ca]-ontology $\mathcal{O}$ such that φ_0 is valid if and only if $\top$ is L_0-satisfiable w.r.t. $\mathcal{O}$.

The construction of $\mathcal{O}$ may be viewed as an extension of the construction from the previous reduction. In particular, we use the same concept and role names, plus a concept name F and a role name s. We next give details. To construct $\mathcal{O}$, we reuse statements (1) to (6) from the previous reduction, adding an $s(x_1, x_2)$-atom to the query in refinements (1) to (3). As in the previous reduction, this generates a full binary tree of refinements of depth n that represents all truth assignments as domain elements on level L_n. We next implement a bottom-up pass on this tree that evaluates the quantifiers in φ_0 using the concept name F. We first add to $\mathcal{O}$:

$$F_\varphi \sqsubseteq_{L_n} F. \tag{8}$$

For each $i \in \{1, \dots, n\}$ with $Q_i = \forall$ and $j \in \{1, 2\}$, we further add the following concept abstraction:

$$L_{i-1}\colon F \ \underline{\mathsf{abstracts}}\ L_i\colon q(x_1, x_2) \text{ where} \quad (9)$$
$$q(x_1, x_2) = F(x_j) \wedge s(x_1, x_2)$$

and for each $i \in \{1, \dots, n\}$ with $Q_i = \exists$, we add the concept abstraction

$$L_{i-1}\colon F \ \underline{\mathsf{abstracts}}\ L_i\colon q(x_1, x_2) \text{ where} \quad (10)$$
$$q(x_1, x_2) = F(x_1) \wedge F(x_2) \wedge s(x_1, x_2)$$

Note that, as required and due to our use of the role name s, the queries in these abstraction statements are connected. This is in fact the only reason why s was introduced. Finally, we add the following CI, representing our wish that φ_0 is valid:

$$F \sqsubseteq_{L_0} \bot. \quad (11)$$

Lemma 7. *φ_0 is valid iff $\top$ is L_0-satisfiable w.r.t. $\mathcal{O}$.*

Finally, we prove the following.

Theorem 4. *Satisfiability in $\mathcal{EL}^{\mathsf{abs}}$[rr] is 2EXPTIME-hard.*

This is achieved by reducing the word problem for exponentially space-bounded alternating Turing machines. More precisely, we adapt a reduction from (Lutz and Schulze 2023) used there to show that satisfiability in $\mathcal{ALC}^{\mathsf{abs}}$[rr] is 2EXPTIME-hard.

6 Getting To Polynomial Time

We now consider a semantic variation of $\mathcal{ELH}_r^{\mathsf{abs}}$[cr] that reduces the complexity of satisfiability from coNP to PTime. This variation is obtained by letting L-ensembles be *sets* rather than tuples, that is, dropping the order of elements in the ensemble. Moreover, refinements are now interpreted as a *partial* description of an ensemble, that is, the variables in the CQ used in the refinement describe elements of the ensemble that must exist, but other elements may exist as well.

In more detail, the refinement function ρ is now a partial function that associates every pair $(d, L) \in \Delta^{\mathcal{I}} \times \mathbf{A}_{\mathcal{I}}$ such that $L \prec L(d)$ with a non-empty subset of $\Delta^{\mathcal{I}_L}$ called an *L-ensemble*. We still require that every object participates in at most one ensemble, that is, Property $(*)$ from the original definition of the semantics is still required to be satisfied. The semantics of refinement statements is then as follows. An A-interpretation $\mathcal{I} = (\mathbf{A}_{\mathcal{I}}, \prec, (\mathcal{I}_L)_{L \in \mathbf{A}_{\mathcal{I}}}, \rho)$ *satisfies* a

- concept refinement $L{:}q(\bar{x})$ $\underline{\mathsf{refines}}$ $L'{:}C$ if $L, L' \in \mathbf{A}_{\mathcal{I}}$ and for all $d \in C^{\mathcal{I}_{L'}}$, there is an $\bar{e} \in q(\mathcal{I}_L)$ s.t. all elements of $\bar{e}$ are in $\rho_L(d)$;
- role refinement $L{:}q(\bar{x}, \bar{y})$ $\underline{\mathsf{refines}}$ $L'{:}q_r(x, y)$ if $L, L' \in \mathbf{A}_{\mathcal{I}}$ and for all $(d_1, d_2) \in q_r(\mathcal{I}_{L'})$, there is an $(\bar{e}_1, \bar{e}_2) \in q(\mathcal{I}_L)$ s.t. all elements of $\bar{e}_i$ are in $\rho_L(d_i)$, for $i \in \{1, 2\}$.

We call this semantics the *set ensemble semantics*. Note that under this semantics, we can no longer simulate $\bot$ by concept refinement. We instead assume that $\bot$ is explicitly available as a concept constructor (to ensure that satisfiability and subsumption are mutually reducible).

The following example illustrates the impact of switching to set ensemble semantics which, we believe, is fairly mild if the modeling discipline is adjusted in a suitable way.

Example 4. *Consider the following ontology $\mathcal{O}$:*

$$\mathsf{SportsCar} \sqsubseteq \mathsf{Car}$$
$$L_1{:}\mathsf{Engine}(x_1) \wedge \mathsf{Body}(x_2)\ \underline{\mathsf{refines}}\ L_2{:}\mathsf{Car}$$
$$L_1{:}\mathsf{TurboEngine}(x_1) \wedge \mathsf{Body}(x_2)\ \underline{\mathsf{refines}}\ L_2{:}\mathsf{SportsCar}$$

Under the standard semantics, every sports car refines into an ensemble of exactly two elements, the first one both an engine and a turbo engine, and the second one a body. Under set ensemble semantics, a sports car may refine into an ensemble of three elements: an engine, a turbo engine, and a body. If we add the natural concept inclusion

$$\mathsf{TurboEngine} \sqsubseteq \mathsf{Engine},$$

then the turbo engine is also an engine and, arguably, the difference between the two semantics becomes negligible.

We aim to prove the following.

Theorem 5. *Under the set ensemble semantics, satisfiability in $\mathcal{EL}^{\mathsf{abs}}$[cr, rr] is in PTime.*

We prove Theorem 5 by providing a polynomial time reduction from L-satisfiability in $\mathcal{ELH}_r^{\mathsf{abs}}$[cr] to satisfiability in $\mathcal{ELHO}_{r,\bot}$, the extension of $\mathcal{ELH}_r$ with nominals and $\bot$. More precisely, we assume a countably infinite set $\mathbf{I}$ of individuals and admit expressions $\{a\}$, with $a \in \mathbf{I}$, as concepts. The semantics is given by $\{a\}^{\mathcal{I}} = a$ for all interpretations $\mathcal{I}$. It is known that satisfiability in $\mathcal{ELHO}_{r,\bot}$ is in PTime (Krötzsch 2010).

Let C_0 be an $\mathcal{EL}$-concept, $\mathcal{O}$ an $\mathcal{ELH}_r^{\mathsf{abs}}$[cr]-ontology in normal form, and $L_0 \in \mathbf{A}_{\mathcal{O}}$, given as input. We assume w.l.o.g. that no two CQs in (refinements in) $\mathcal{O}$ share a variable. If $G_{\mathcal{O}}$ is not a tree, we may directly return 'unsatisfiable'. Otherwise, we construct in polynomial time an $\mathcal{ELHO}_{r,\bot}$-ontology $\mathcal{O}'$. Introduce a fresh role name r_L for each role name r in $\mathcal{O}$ and each abstraction level L in $\mathcal{O}$, and an additional fresh role name u. We include in $\mathcal{O}'$ the following concept inclusions:

1. $\top \sqsubseteq \exists u.L$ for all $L \in \mathbf{A}_{\mathcal{O}}$;
2. $L \sqcap A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$ for all $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq_L B$ in $\mathcal{O}$ (with $A_1 \sqcap \cdots \sqcap A_n = \top$ if $n = 0$);
3. $\exists r_L.A \sqsubseteq B$ for all $\exists r.A \sqsubseteq_L B$ in $\mathcal{O}$;
4. $L \sqcap A \sqsubseteq \exists r_L.(L \sqcap B)$ for all $A \sqsubseteq_L \exists r.B$ in $\mathcal{O}$;
5. for all $L{:}q(\bar{x})$ $\underline{\mathsf{refines}}$ $L'{:}A$ in $\mathcal{O}$:
   - $L' \sqcap A \sqsubseteq \exists u.(L \sqcap B \sqcap \{a_x\})$ whenever $B(x)$ is an atom in q;
   - $L' \sqcap A \sqsubseteq \exists u.(L \sqcap \{a_x\} \sqcap \exists r_L.(L \sqcap \{a_y\}))$ whenever $r(x, y)$ is an atom in q;
6. for all $L{:}q(\bar{x}, \bar{y})$ $\underline{\mathsf{refines}}$ $L'{:}q_r(x, y)$ in $\mathcal{O}$ with $q_r = A_1(x) \wedge r(x, y) \wedge A_2(y)$:
   - $A_1 \sqcap \exists r_{L'}.A_2 \sqsubseteq \exists u.(L \sqcap B \sqcap \{a_x\})$ whenever $B(x)$ is an atom in q;
   - $A_1 \sqcap \exists r_{L'}.A_2 \sqsubseteq \exists u.(L \sqcap \{a_x\} \sqcap \exists s_L.(L \sqcap \{a_y\}))$ whenever $s(x, y)$ is an atom in q;

Moreover, $\mathcal{O}'$ contains the following:

7. the role inclusion $r_L \sqsubseteq s_L$ for every role inclusion $r \sqsubseteq_L s$ in $\mathcal{O}$;

8. the range restriction $\top \sqsubseteq \forall r_L.C$ for every range restriction $\top \sqsubseteq_L \forall r.C$ in $\mathcal{O}$.

Lemma 8. *C_0 is L_0-satisfiable w.r.t. $\mathcal{O}$ under set ensemble semantics iff $C_0 \sqcap L_0$ is satisfiable w.r.t. $\mathcal{O}'$.*

This proves Theorem 5. An extension to the case that includes concept or role abstractions is not easily possible. In fact, it is straightforward to prove the following by a reduction from the homomorphism problems on directed graphs (the semantics of concept abstractions is defined as expected).

Theorem 6. *Under the set ensemble semantics, satisfiability is coNP-hard in $\mathcal{EL}^{\mathsf{abs}}[\mathrm{cr}, \mathrm{ca}]$.*

We remark that exactly the same reduction as given in this section also serves to reduce satisfiability in $\mathcal{ALC}^{\mathsf{abs}}[\mathrm{cr}, \mathrm{rr}]$ under set ensemble semantics to satisfiability in $\mathcal{ALCO}$, the extension of $\mathcal{ALC}$ with nominals. The latter problem is ExpTime-complete (Tobies 2001), which explains the entry for $\mathcal{ALC}^{\mathsf{abs}}$ under set ensemble semantics in Figure 1.

7 Conclusion

We have studied description logics of refinement and abstraction based on members of the $\mathcal{EL}$ family. While, compared to the $\mathcal{ALC}$ version, the computational complexity does not drop for the full logic, we have identified natural fragments where it does. We leave the complexity of other (less natural) fragments such as $\mathcal{ELH}_r^{\mathsf{abs}}[\mathrm{ca}]$ and $\mathcal{ELH}_r^{\mathsf{abs}}[\mathrm{ca}, \mathrm{ra}]$ as future work.

It would be interesting to consider DLs of abstraction and refinement based on the extension $\mathcal{ELI}$ of $\mathcal{EL}$ with inverse roles. Then already reasoning in the base logic is ExpTime-hard so we cannot expect any lower complexities. One might also define ontology languages with abstraction and refinement based on existential rules, see e.g. (Baget *et al.* 2011; Calì *et al.* 2010). It is then natural to extend the arity of all relations by one position that represents the abstraction level. Note, however, that since every object is required to refine only into a single ensemble, it does not seem possible to encode abstraction and refinement into existing (decidable) existential rule formalisms in a simple way.

Acknowledgments

The research reported in this paper has been supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 Project-ID 329551904 “EASE - Everyday Activity Science and Engineering”, University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject “P02 - Ontologies with Abstraction”.

This work is partly supported by BMBF (Federal Ministry of Education and Research) in DAAD project 57616814 (SECAI, School of Embedded Composite AI) as part of the program Konrad Zuse Schools of Excellence in Artificial Intelligence.

References

Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI*, pages 364–369. Professional Book Center, 2005.

Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In *Proc. of IJCAI*, pages 712–717. IJCAI/AAAI, 2011.

Thomas Bittner and Barry Smith. A theory of granular partitions. In *Foundations of Geographic Information Science*, pages 117–149. Taylor & Francis, 2003.

Silvia Calegari and Davide Ciucci. Granular computing applied to ontologies. *Int. J. Approx. Reason.*, 51(4):391–409, 2010.

Andrea Calì, Georg Gottlob, Thomas Lukasiewicz, Bruno Marnette, and Andreas Pieris. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *Proc. of LICS*, pages 228–242. IEEE Computer Society, 2010.

Ashok K Chandra, Dexter C Kozen, and Larry J Stockmeyer. Alternation. *Journal of the ACM (JACM)*, 28(1):114–133, 1981.

Gianluca Cima, Marco Console, Maurizio Lenzerini, and Antonella Poggi. Monotone abstractions in ontology-based data management. In *Proc. of AAAI*, pages 5556–5563. AAAI Press, 2022.

Birte Glimm, Yevgeny Kazakov, and Trung-Kien Tran. Ontology materialization by abstraction refinement in horn SHOIF. In *Proc. of AAAI*, pages 1114–1120. AAAI Press, 2017.

Szymon Klarman and Víctor Gutiérrez-Basulto. Description logics of context. *J. Log. Comput.*, 26(3):817–854, 2016.

Markus Krötzsch. Efficient inferencing for OWL EL. In *Proc. of JELIA*, volume 6341 of *LNCS*, pages 234–246. Springer, 2010.

Francesca A. Lisi and Corrado Mencar. A granular computing method for OWL ontologies. *Fundam. Informaticae*, 159(1-2):147–174, 2018.

Carsten Lutz and Lukas Schulze. Description logics with abstraction and refinement. In *Proc. of KR*, pages 492–501, 2023.

Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. of IJCAI*. AAAI Press, 2009.

Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz, editors. *OWL 2 Web Ontology Language Profiles (Second Edition)*. W3C Recommendation, 2009. Available at https://www.w3.org/TR/owl2-profiles/.

Christos H Papadimitriou. Computational complexity. In *Encyclopedia of computer science*, pages 260–265. 2003.

Stephan Tobies. *Complexity results and practical algorithms for logics in knowledge representation*. PhD thesis, RWTH Aachen University, Germany, 2001.

Frank Wolter and Michael Zakharyaschev. Multi-dimensional description logics. In *Proc. of IJCAI*, pages 104–109. Morgan Kaufmann, 1999.