Fixed Parameter Tractable Reasoning in DLs via Decomposition

František Simančík, Boris Motik, and Markus Krötzsch

Department of Computer Science, University of Oxford, UK

1 Introduction

DL reasoning is of high computational complexity even for basic DLs such as $\mathcal{ALCI}$ [3, Chapter 3]. Intuitively, due to disjunctions (*or-branching*) and/or existential quantifiers (*and-branching*), a DL reasoner may need to investigate (at least) exponentially many combinations of concepts. A range of highly-tuned optimizations, such as absorption, dependency-directed backtracking, blocking, and caching [3, Chapter 9], can be used to tame these sources of complexity. None of these techniques, however, provide formal tractability guarantees. Such guarantees can be obtained by restricting the language expressivity, as done in the $\mathcal{EL}$ [2], DL-Lite [4,1], and DLP [8] families of DLs. Tractable DLs typically do not support disjunctions, which eliminates or-branching, and they either significantly restrict universal quantification (as in $\mathcal{EL}$ and DL-Lite) or disallow existential quantification (as in DLP), which eliminates or reduces and-branching.

Obtaining tractability guarantees for hard computational problems has been extensively studied in *parameterized complexity* [5]. The general idea is to measure the "hardness" of a problem instance of size n using a nonnegative integer *parameter* k, and the goal is to solve the problem in time that becomes polynomial in n whenever k is fixed. A particular goal is to identify *fixed parameter tractable* (FPT) problems, which can be solved in time $f(k) \cdot n^c$, where c is a constant and f is an arbitrary computable function that depends *only* on k. Note that not every problem that becomes tractable if k is fixed is in FPT. For example, checking whether a graph of size n contains a clique of size k can clearly be performed in time $O(n^k)$, which is polynomial if k is a constant; however, since k is in the exponent of n, this does not prove membership in FPT.

Note that every problem is FPT if the parameter is the problem's size, so a useful parameterization should allow increasing the size arbitrarily while keeping the parameter bounded. Various problems in AI were successfully parameterized by exploiting the graph-theoretic notions of *tree decompositions* and *treewidth* [6,7,10], which we recapitulate next. A *hypergraph* is a pair $G = \langle V, H \rangle$ where V is a set of vertices and $H \subseteq 2^V$ is a set of *hyperedges*. A *tree decomposition* of G is a pair $\langle T, L \rangle$ where T is an undirected tree whose sets of vertices (also called *bags*) and edges are denoted with $\mathsf{B}(T)$ and $\mathsf{E}(T)$, and $L : \mathsf{B}(T) \to 2^V$ is a labeling of $\mathsf{B}(T)$ by subsets of V such that

(T1) for each $v \in V$, the set $\{b \in \mathsf{B}(T) \mid v \in L(b)\}$ induces a connected subtree of T, and
(T2) for each $e \in H$, there exists a bag $b \in \mathsf{B}(T)$ such that $e \subseteq L(b)$.

The *width* of $\langle T, L \rangle$ is defined as $\max_{b \in \mathsf{B}(T)} L(b) - 1$. Finally, the *treewidth* of G is the minimum width among all possible tree decompositions of G. Consider now an instance N of the SAT problem, where N is a finite set of clauses (i.e., disjunctions of possibly negated propositional variables). The notions of tree decompositions and treewidth of N are defined w.r.t. the hypergraph $G_N = \langle V_N, H_N \rangle$ where V_N is the set of propositional variables occurring in N, and H_N contains the hyperedge $\{p_1, \ldots, p_k\}$ for each clause $(\neg)p_1 \vee \ldots \vee (\neg)p_k \in N$. When parameterized by treewidth, SAT is FPT [10]. Intuitively, the treewidth of N shows how many propositional variables must be considered simultaneously in order to check the satisfiability of N; thus, bounding the treewidth has the effect of bounding or-branching.

Inspired by these results, we present a novel DL reasoning algorithm that ensures fixed parameter tractability. To this end, in Section 3 we introduce a notion of a *decomposition* $\mathcal{D}$ of a signature Σ. Intuitively, $\mathcal{D}$ is a graph that restricts the propagation information between the atomic concepts in Σ. A decomposition of Σ can be seen as one or more tree decompositions, each reflecting the propagation of information due to or-branching, interconnected to reflect the propagation of information due to and-branching. We identify a parameter of $\mathcal{D}$ called *width*; intuitively, this parameter determines an upper bound on the number of concepts that must be considered simultaneously to solve a reasoning problem. Let $\mathcal{O}$ be an $\mathcal{ALCI}$ ontology *normalized* to contain only axioms of the form $\bigsqcap_i A_i \sqsubseteq \bigsqcup_j B_j$, $A \sqsubseteq \exists R.B$, and $A \sqsubseteq \forall R.B$, where $A_{(i)}$ and $B_{(j)}$ are atomic concepts, and R is a (possibly inverse) role. We present a resolution-based reasoning calculus that runs in time $O(f(d) \cdot |\mathcal{D}| \cdot |\mathcal{O}|)$, where d is the width of $\mathcal{D}$, $|\mathcal{D}|$ is the size of $\mathcal{D}$, and $|\mathcal{O}|$ is the number of axioms in $\mathcal{O}$. Our calculus is not complete for all $\mathcal{D}$: it is not guaranteed to derive all consequences that might be of interest. To remedy that, we introduce a notion of $\mathcal{D}$ being *admissible* for $\mathcal{O}$ and the relevant consequences, and we show that admissibility guarantees completeness.

Ideally, given $\mathcal{O}$ and the relevant consequences, one would identify an admissible decomposition $\mathcal{D}$ of smallest width and then run our calculus in order to obtain an FPT algorithm. In Section 4, however, we show that, for certain $\mathcal{O}$, all admissible decompositions of smallest width have exponentially many vertices. This is in contrast to tree decompositions (e.g., for each instance of SAT, a tree decomposition of minimal width exists in which the number of vertices is linear in the size of the instance) and is due to the fact that, in addition to or-branching, our decompositions analyze information flow due to and-branching as well. We therefore further restrict the notion of admissible decompositions in several ways. For each of the resulting notions, one can compute a decomposition of width at most d (if one exists) in time $f(d) \cdot |\mathcal{O}|^c$ with f a computable function and c an integer constant; together with our resolution-based calculus, we thus obtain an FPT calculus for reasoning with normalized $\mathcal{ALCI}$ ontologies.

In Section 5 we show that the minimum decomposition width of several commonly used ontologies is much smaller than the respective ontology's size. This suggests that decomposition width provides a "reasonable" measure of ontology complexity, and that our approach might even provide practical tractability guarantees.

Our results can be applied to $\mathcal{SHI}$ ontologies by transforming away role hierarchies and transitivity and normalizing the ontology in a preprocessing step. Such transformations, however, are don't-care nondeterministic, and the minimum decomposition width of the normalization result might depend on the nondeterministic choices. In this paper we thus restrict our attention to normalized $\mathcal{ALCI}$ ontologies, and we leave an investigation of how normalization affects the minimum width for future work.

$$\mathsf{R_1}\ \frac{}{A \sqsubseteq A} \qquad\qquad \mathsf{R_2}\ \frac{K_1 \sqsubseteq M_1 \sqcup A \quad A \sqcap K_2 \sqsubseteq M_2}{K_1 \sqcap K_2 \sqsubseteq M_1 \sqcup M_2}$$

$$\mathsf{R_3}\ \frac{}{K \sqsubseteq M} : K \sqsubseteq M \in \mathcal{O} \qquad \mathsf{R_4}\ \frac{\begin{array}{c} B \sqcap \bigsqcap_i D_i \sqsubseteq \bigsqcup_j E_j \\ \text{or} \quad \bigsqcap_i D_i \sqsubseteq \bigsqcup_j E_j \end{array}}{A \sqcap \bigsqcap_i C_i \sqsubseteq \bigsqcup_j F_j} : \begin{array}{l} A \sqsubseteq \exists R.B \in \mathcal{O} \\ C_i \sqsubseteq \forall R.D_i \in \mathcal{O} \\ E_j \sqsubseteq \forall R^{-}.F_j \in \mathcal{O} \end{array}$$

Fig. 1. A simple resolution calculus

The proofs of all results presented in this paper are available in the technical report at http://www.comlab.ox.ac.uk/boris.motik/pubs/smk11dl-decomposition.pdf.

2 Source of Complexity in DL Reasoning

In order to motivate the results presented in the following sections, in this section we present a very simple calculus that is not FPT, and we discuss the rough idea for making the calculus FPT. The calculus is based on resolution, and is similar to the calculus presented in [9]. Resolution can often provide worst-case optimal calculi whose best case complexity is significantly lower than the worst case complexity; indeed, the calculus from [9] has demonstrated excellent practical performance.

The calculus manipulates *clauses*—expressions of the form $K \sqsubseteq M$, where K is a finite conjunction of atomic concepts, and M is a finite disjunction of atomic concepts. With $\mathsf{sig}(K)$, $\mathsf{sig}(M)$, and $\mathsf{sig}(K \sqsubseteq M)$ we denote the sets of atomic concepts occurring in K, M, and $K \sqsubseteq M$, respectively. We consider two disjunctions (resp. conjunctions) to be the same whenever they mention the same atoms; that is, we disregard the order and the multiplicity of atoms. We write empty K and M as $\top$ and $\bot$, respectively. Furthermore, we say that a clause $K' \sqsubseteq M'$ is a *strengthening* of a clause $K \sqsubseteq M$ if $\mathsf{sig}(K') \subseteq \mathsf{sig}(K)$ and $\mathsf{sig}(M') \subseteq \mathsf{sig}(M)$. We write $K \sqsubseteq M \,\hat{\in}\, \mathcal{N}$ if the set of clauses $\mathcal{N}$ contains at least one strengthening of the clause $K \sqsubseteq M$.

Given a normalized ontology $\mathcal{O}$, our calculus constructs a *derivation*—a sequence $\mathcal{S}_0, \mathcal{S}_1, \ldots$ of sets of clauses such that $\mathcal{S}_0 = \emptyset$, and for each $i > 0$, set $\mathcal{S}_i$ is obtained from $\mathcal{S}_{i-1}$ by applying a rule from Fig. 1. Rules $\mathsf{R_1}$ and $\mathsf{R_2}$ implement propositional resolution, and rule $\mathsf{R_3}$ ensures that each clause in $\mathcal{O}$ is taken into account. Rule $\mathsf{R_4}$ handles role restrictions; letter R stands for a role (i.e., R need not be atomic), and $\mathsf{inv}(R)$ is the inverse role of R; finally, note that the atom B in the premise of the rule is optional. Intuitively, the rule says that, if B, D_i, and $\neg E_j$ jointly imply a contradiction, but $A \sqsubseteq \exists R.B$, $C_i \sqsubseteq \forall R.D_i$, and $\neg F_j \sqsubseteq \forall R.\neg E_j$ hold, then A, C_i, and $\neg F_j$ jointly imply a contradiction too. Reasoning with the second premise is analogous.

A *saturation* is defined as $\mathcal{S} := \bigcup_i \mathcal{S}_i$. The calculus *infers* a clause $K \sqsubseteq M$, written $\mathcal{O} \vdash K \sqsubseteq M$, if $K \sqsubseteq M \,\hat{\in}\, \mathcal{S}$. It is straightforward to see that the calculus is *sound*: if $\mathcal{O} \vdash K \sqsubseteq M$, then $\mathcal{O} \models K \sqsubseteq M$. Typically, resolution is used as a refutation-complete calculus; however, it is possible to show that the variant of resolution presented here is *complete* in the following stronger sense: if $\mathcal{O} \models K \sqsubseteq M$, then $\mathcal{O} \vdash K \sqsubseteq M$; note that this means that the calculus infers *at least one strengthening* of each clause entailed by

$\mathcal{O}$. This stronger notion of completeness can be useful in practice; for example, $\mathcal{O}$ can be classified using a single run of the calculus, which is not the case for calculi (such as tableau) that are only refutationally complete.

Let d be the number of atomic concepts in $\mathcal{O}$. Since each clause is uniquely identified by the atomic concepts that occur in K and/or M, the calculus can derive at most 4^d clauses, which is exponential in $|\mathcal{O}|$. The high complexity of DL reasoning arises because one may have to consider exponentially many combinations of concepts, and this fact fundamentally underpins all DL reasoning algorithms. Clearly, a tractable algorithm should consider only polynomially many combinations. For example, reasoning algorithms for $\mathcal{EL}$ exploit the fact that only polynomially many combinations are "relevant" and that all of them can be constructed deterministically. In the following sections, we ensure tractability of reasoning in a radically different way. Instead of restricting the ontology language, we show that by restricting the structure of the ontology with a suitable parameter one can limit the number of concepts that must be simultaneously considered, which effectively limits the exponent in the above calculation. Since the base of the exponent not depend on $|\mathcal{O}|$, we will thus obtain an FPT reasoning calculus.

3 Reasoning with Decompositions

In this section we develop the notions of decomposition, decomposition admissibility, and the resolution calculus. We start by introducing the notion of decomposition.

Definition 1. *Let $\Sigma = \langle \Sigma_A, \Sigma_R \rangle$ be a DL signature, where Σ_A is a finite set of atomic concepts and Σ_R is a finite set of atomic roles; let $\Sigma_{R^-} = \{R^- \mid R \in \Sigma_R\}$ be the set of inverse roles of Σ_R; and let ϵ be a symbol not contained in $\Sigma_A \cup \Sigma_R \cup \Sigma_{R^-}$.*

A decomposition *of Σ is a labeled graph $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig} \rangle$, where $\mathcal{V}$ is a finite set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times (\Sigma_R \cup \Sigma_R^- \cup \{\epsilon\})$ is a set of directed edges labeled by a role or by ϵ, and $\mathsf{sig} : \mathcal{V} \to 2^{\Sigma_A}$ is a labeling of each vertex with a set of atomic concepts. The* width *of $\mathcal{D}$ is defined as $\mathsf{wd}(\mathcal{D}) := \max_{v \in \mathcal{V}} |\mathsf{sig}(v)|$.*

Note that $\mathcal{D}$ is not defined w.r.t. an ontology, but w.r.t. a signature Σ, and we will establish a link between $\mathcal{D}$ and $\mathcal{O}$ shortly in our notion of admissibility. This is mainly so as to gather all conditions that guarantee completeness in one place. We discuss the intuition behind this definition after presenting the resolution-based calculus.

Definition 2. *Let Σ be a DL signature, let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig} \rangle$ be a decomposition of Σ, and let $\mathcal{O}$ be a normalized $\mathcal{ALCI}$ ontology over Σ. The* resolution calculus *for $\mathcal{D}$ and $\mathcal{O}$ is defined as follows.*

A clause system *for $\mathcal{D}$ is a function $\mathcal{S}$ that assigns to each vertex $v \in \mathcal{V}$ a set of clauses $\mathcal{S}(v)$. A* derivation *of the calculus is a sequence of clause systems $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots$ such that $\mathcal{S}_0(v) = \emptyset$ for each $v \in \mathcal{V}$ and, for each $i > 0$, $\mathcal{S}_i$ is obtained from $\mathcal{S}_{i-1}$ by an application of a derivation rule from Fig. 2; we assume that each derivation is* fair *in the usual sense. The* saturation *is the clause system $\mathcal{S}$ defined by $\mathcal{S}(v) := \bigcup_i \mathcal{S}_i(v)$ for each $v \in \mathcal{V}$. The calculus* infers *a clause $K \sqsubseteq M$ at vertex v, written $\mathcal{O}, v \vdash_{\mathcal{D}} K \sqsubseteq M$, if $K \sqsubseteq M \mathrel{\hat{\in}} \mathcal{S}(v)$; furthermore, the calculus* infers *a clause $K \sqsubseteq M$, written $\mathcal{O} \vdash_{\mathcal{D}} K \sqsubseteq M$, if a vertex $v \in \mathcal{V}$ exists such that $\mathcal{O}, v \vdash_{\mathcal{D}} K \sqsubseteq M$.*

$$\mathbf{R_1}\ \frac{}{\text{add } A \sqsubseteq A \text{ to } \mathcal{S}(v)} : A \in \mathsf{sig}(v)$$

$$\mathbf{R_2}\ \frac{K_1 \sqsubseteq M_1 \sqcup A \in \mathcal{S}(v) \quad A \sqcap K_2 \sqsubseteq M_2 \in \mathcal{S}(v)}{\text{add } K_1 \sqcap K_2 \sqsubseteq M_1 \sqcup M_2 \text{ to } \mathcal{S}(v)}$$

$$\mathbf{R_3}\ \frac{}{\text{add } K \sqsubseteq M \text{ to } \mathcal{S}(v)} : \begin{array}{l} K \sqsubseteq M \in \mathcal{O} \\ \mathsf{sig}(K \sqsubseteq M) \subseteq \mathsf{sig}(v) \end{array}$$

$$\mathbf{R_4}\ \frac{\begin{array}{rr} & B \sqcap \bigsqcap_i D_i \sqsubseteq \bigsqcup_j E_j \in \mathcal{S}(u) \\ \text{or} & \bigsqcap_i D_i \sqsubseteq \bigsqcup_j E_j \in \mathcal{S}(u) \end{array}}{\text{add } A \sqcap \bigsqcap_i C_i \sqsubseteq \bigsqcup_j F_j \text{ to } \mathcal{S}(v)} : \begin{array}{l} A \sqsubseteq \exists R.B \in \mathcal{O} \\ C_i \sqsubseteq \forall R.D_i \in \mathcal{O} \\ E_j \sqsubseteq \forall \mathsf{inv}(R).F_j \in \mathcal{O} \\ \langle u, v, R \rangle \in \mathcal{E} \\ \mathsf{sig}(A \sqcap \bigsqcap_i C_i \sqsubseteq \bigsqcup_j F_j) \subseteq \mathsf{sig}(v) \end{array}$$

$$\mathbf{R_5}\ \frac{K \sqsubseteq M \in \mathcal{S}(u)}{\text{add } K \sqsubseteq M \text{ to } \mathcal{S}(v)} : \begin{array}{l} \langle u, v, \epsilon \rangle \in \mathcal{E} \\ \mathsf{sig}(K \sqsubseteq M) \subseteq \mathsf{sig}(v) \end{array}$$

Fig. 2. The decomposition calculus

The calculus is complete *(*sound*) if $\mathcal{O} \models K \sqsubseteq M$ implies (is implied by) $\mathcal{O} \vdash_{\mathcal{D}} K \sqsubseteq M$ for each clause $K \sqsubseteq M$ over Σ. Given a set of clauses $\mathcal{C}$ over Σ, the calculus is $\mathcal{C}$-*complete *if $\mathcal{O} \models K \sqsubseteq M$ implies $\mathcal{O} \vdash_{\mathcal{D}} K \sqsubseteq M$ for each $K \sqsubseteq M \in \mathcal{C}$.*

While the simple calculus from Section 2 saturates a single set of clauses, the resolution calculus for $\mathcal{D}$ and $\mathcal{O}$ saturates one set of clauses per decomposition vertex. In particular, for a vertex $v \in \mathcal{V}$, set $\mathcal{S}(v)$ contains only clauses whose propositional atoms are all contained in $\mathsf{sig}(v)$, so v identifies a propositional subproblem of $\mathcal{O}$. Rules $\mathbf{R_1}$–$\mathbf{R_3}$ implement propositional resolution "within" each vertex v. Rule $\mathbf{R_5}$ propagates propositional consequences from vertex u to vertex v connected by an ϵ-labeled edge; thus, the ϵ-labeled edges of $\mathcal{D}$ "connect" the subproblems of $\mathcal{O}$ in accordance with or-branching. Finally, rule $\mathbf{R_4}$ propagates modal consequences from a vertex u to a vertex v connected by an R-labeled edge; thus, the R-labeled edges of $\mathcal{D}$ "connect" the subproblems of $\mathcal{O}$ in accordance with and-branching. A clause is inferred if at least one saturated set $\mathcal{S}(v)$ contains a strengthening of the clause.

Note that rules $\mathbf{R_1}$–$\mathbf{R_3}$ consider only one vertex at a time, whereas rules $\mathbf{R_4}$ and $\mathbf{R_5}$ involve two vertices. Thus, although this was not our initial motivation, the calculus seems to exhibit significant parallelization potential. We leave a thorough investigation of the reasoning problem in terms of parallel complexity classes for future work.

The notion of $\mathcal{C}$-completeness takes into account that one might be interested not only in refutational completeness, but in the derivation of all clauses from some set $\mathcal{C}$. For example, if one is interested in the classification of $\mathcal{O}$, then $\mathcal{C}$ would contain all clauses of the form $A \sqsubseteq B$ with A and B atomic concepts occurring in $\mathcal{O}$.

The following proposition determines the complexity of the calculus in terms of the sizes of $\mathcal{D}$ and the number $|\mathcal{O}|$ of axioms in $\mathcal{O}$. It essentially observes two key facts: first, since the clauses in each $\mathcal{S}(v)$ are restricted to atomic concepts in $\mathsf{sig}(v)$, the maximum number of clauses in $\mathcal{S}(v)$ is determined solely by $\mathsf{wd}(\mathcal{D})$; and second, given a node or a pair of nodes, all rules can be applied in time that also depends solely on $\mathsf{wd}(\mathcal{D})$. Once we limit the size of $\mathcal{D}$, this proposition will provide us with an FPT algorithm.

Proposition 1. *Let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig}\rangle$ and O be as in Definition 2. The saturation of the resolution calculus for $\mathcal{D}$ and O can be computed in time $O(f(\mathsf{wd}(\mathcal{D})) \cdot (|\mathcal{V}| + |\mathcal{E}|) \cdot |O|)$, where f is some computable function.*

The rules of our calculus are clearly sound for arbitrary decompositions $\mathcal{D}$ and ontologies O; however, the converse is not true. As a trivial example, note that the decomposition with the empty vertex and edge sets satisfies Definition 1, and that our calculus does not infer any clause using such $\mathcal{D}$. Therefore, we next introduce the notion of *admissibility*, which we later show to be sufficient for completeness.

Definition 3. *Let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig}\rangle$ be a decomposition of a DL signature $\Sigma = \langle \Sigma_A, \Sigma_R\rangle$.*

Let $\mathcal{W} \subseteq \mathcal{V}$ be an arbitrary set of vertices. The signature *of $\mathcal{W}$ is defined as $\mathsf{sig}(\mathcal{W}) := \bigcup_{w\in\mathcal{W}} \mathsf{sig}(w)$. The* ϵ-projection *of $\mathcal{D}$ w.r.t. $\mathcal{W}$ is the undirected graph $\mathcal{D}_{\mathcal{W}}$ that contains the undirected edge $\{u, v\}$ for each $\langle u, v, \epsilon\rangle \in \mathcal{E}$ with $u, v \in \mathcal{W}$. Set $\mathcal{W}$ is* ϵ-connected *if, for all $u, v \in W$, vertices $\{w_0, w_1, \ldots, w_n\} \subseteq \mathcal{W}$ exist such that $w_0 = u$, $w_n = v$, and $\langle w_{i-1}, w_i, \epsilon\rangle \in \mathcal{E}$ for each $1 \leq i \leq n$; furthermore, $\mathcal{W}$ is an* ϵ-component *of $\mathcal{D}$ if $\mathcal{W}$ is ϵ-connected, and each $\mathcal{W}'$ such that $\mathcal{W} \subsetneq \mathcal{W}' \subseteq \mathcal{V}$ is not ϵ-connected.*

Decomposition $\mathcal{D}$ is admissible *for an ontology O if $\langle u, v, \epsilon\rangle \in \mathcal{E}$ implies $\langle v, u, \epsilon\rangle \in \mathcal{E}$ for all $u, v \in \mathcal{V}$, and if each ϵ-component $\mathcal{W}$ of $\mathcal{D}$ satisfies the following properties:*

(i) *$\mathcal{D}_{\mathcal{W}}$ is an undirected tree;*
(ii) *for each atomic concept $A \in \mathsf{sig}(\mathcal{W})$, the set $\{w \in \mathcal{W} \mid A \in \mathsf{sig}(w)\}$ is ϵ-connected;*
(iii) *for each clause $K \sqsubseteq M \in O$ such that $\mathsf{sig}(K) \subseteq \mathsf{sig}(\mathcal{W})$, a vertex $w \in \mathcal{W}$ exists such that $\mathsf{sig}(K \sqsubseteq M) \subseteq \mathsf{sig}(w)$;*
(iv) *for each axiom $A \sqsubseteq \exists R.B \in O$ such that $A \in \mathsf{sig}(\mathcal{W})$, an ϵ-component $\mathcal{U}$ of $\mathcal{D}$ and vertices $w \in \mathcal{W}$ and $u \in \mathcal{U}$ exist such that*
  - *$\langle u, w, R\rangle \in \mathcal{E}$,*
  - *$A \in \mathsf{sig}(w)$,*
  - *$B \in \mathsf{sig}(u)$,*
  - *for each $C \sqsubseteq \forall R.D \in O$, if $C \in \mathsf{sig}(\mathcal{W})$ then $C \in \mathsf{sig}(w)$ and $D \in \mathsf{sig}(u)$, and*
  - *for each $E \sqsubseteq \forall \mathsf{inv}(R).F \in O$, if $E \in \mathsf{sig}(\mathcal{U})$ then $E \in \mathsf{sig}(u)$ and $F \in \mathsf{sig}(w)$.*

A clause $K \sqsubseteq M$ is covered *by $\mathcal{D}$ if an ϵ-component $\mathcal{W}$ of $\mathcal{D}$ and a vertex $w \in \mathcal{W}$ exist such that $\mathsf{sig}(K) \cup [\mathsf{sig}(M) \cap \mathsf{sig}(\mathcal{W})] \subseteq \mathsf{sig}(w)$. Decomposition $\mathcal{D}$ is* admissible *for C if each clause in C is covered by $\mathcal{D}$.*

Definition 3 incorporates two largely orthogonal ideas. First, each ϵ-component $\mathcal{W}$ of $\mathcal{D}$ reflects the propositional constraints on domain elements of a particular type in a model of O. To deal with or-branching, each $\mathcal{W}$ is a tree decomposition formed by undirected ϵ-labeled edges. Conditions (i)–(iii) are analogous to (T1) and (T2) in Section 1, but (iii) is more general: instead of requiring $\mathsf{sig}(K \sqsubseteq M) \subseteq \mathsf{sig}(w)$ for each $K \sqsubseteq M \in O$ and some $w \in \mathcal{W}$, Condition (iii) takes into account that, if $\mathsf{sig}(K) \nsubseteq \mathsf{sig}(\mathcal{W})$, then $K \sqsubseteq M$ can be satisfied by making the atomic concepts in $\mathsf{sig}(K) \setminus \mathsf{sig}(\mathcal{W})$ false on the appropriate domain element; thus, $\mathsf{sig}(K \sqsubseteq M) \subseteq \mathsf{sig}(w)$ must hold for some $w \in \mathcal{W}$ only if $\mathsf{sig}(K) \subseteq \mathsf{sig}(\mathcal{W})$. Admissibility for C uses an analogous idea.

Second, to deal with and-branching, the ϵ-components of $\mathcal{D}$ are interconnected via role-labeled edges. If a concept A occurs in an ϵ-component $\mathcal{W}$ and in an axiom of $\mathcal{O}$ of the form $A \sqsubseteq \exists R.B$, then a domain element corresponding to $\mathcal{W}$ might need to have an R-successor; to reflect that, $\mathcal{D}$ must contain an ϵ-component $\mathcal{U}$, and vertices $w \in \mathcal{W}$ and $u \in \mathcal{U}$ connected by an R-labeled edge must exist such that $A \in \mathsf{sig}(w)$ and $B \in \mathsf{sig}(u)$. Furthermore, in order to address the universal quantifiers over R, if $C \sqsubseteq \forall R.D \in \mathcal{O}$ and $C \in \mathsf{sig}(\mathcal{W})$, then $C \in \mathsf{sig}(w)$ and $D \in \mathsf{sig}(u)$ must hold, and analogously for universals over $\mathsf{inv}(R)$. These conditions ensure that w and u contain all atomic concepts that might be relevant for modal reasoning, which in turn allows our calculus to infer all relevant constrains on atomic concepts.

The following theorem shows that admissibility indeed ensures completeness.

Theorem 1. *Let $\mathcal{O}$ be an ontology, let $\mathcal{C}$ be a set of clauses, and let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig} \rangle$ be a decomposition that is admissible for $\mathcal{O}$ and $\mathcal{C}$. Then, the resolution calculus for $\mathcal{D}$ and $\mathcal{O}$ is $\mathcal{C}$-complete.*

Ideally, given an ontology $\mathcal{O}$ and a set of clauses $\mathcal{C}$, one would identify a decomposition $\mathcal{D}$ of smallest width and then apply the resolution calculus for $\mathcal{D}$ and $\mathcal{O}$ to obtain an FPT algorithm. The following theorem shows, however, that this idea does not work, since it is not the case that, for each ontology $\mathcal{O}$, there exists a decomposition of minimal width that is admissible for $\mathcal{O}$ and whose size is polynomial in $|\mathcal{O}|$. In order to address this problem, in Section 4 we further restrict the notion of admissibility.

Theorem 2. *A family of $\mathcal{ALCI}$ ontologies $\{\mathcal{O}_n\}$ exists such that each decomposition admissible for $\mathcal{O}_n$ and $\mathcal{C} = \{C \sqsubseteq \bot\}$ of minimal width has size exponential in $|\mathcal{O}_n|$.*

4 Constructing Decompositions of Polynomial Size

In Section 4.3 we present a general method for computing admissible decompositions of polynomial size, for which we obtain the desired FPT result. This method embodies two largely orthogonal ideas, each of which we present separately for didactic purposes. In particular, in Section 4.1 we present an approach for analyzing and-branching, and in Section 4.2 we present an approach for analyzing or-branching.

4.1 Analyzing And-Branching via Deductive Overestimation

In this section we present an approach for analyzing and-branching, which is inspired by the reasoning algorithm for $\mathcal{EL}$ [2]. The approach uses an overestimation of the subsumption relation to construct the decomposition. It manipulates expressions of the form $K \rightsquigarrow A$, where K is a conjunction of atomic concepts, and A is an atomic concept. Given an $\mathcal{ALCI}$ ontology $\mathcal{O}$ and a set of clauses $\mathcal{C}$, the *deductive overestimation* $\rightsquigarrow$ for $\mathcal{O}$ and $\mathcal{C}$ is the relation obtained by exhaustive application of the rules shown in Fig. 3.

Intuitively, $K \rightsquigarrow A$ states that an object whose existence is required to satisfy K can become an instance of A. On $\mathcal{EL}$ ontologies $\rightsquigarrow$ coincides with the subsumption relation, but on more expressive ontologies $\rightsquigarrow$ overestimates the subsumption relation. In order to check whether a clause $K \sqsubseteq M \in \mathcal{C}$ is entailed by $\mathcal{O}$, rule $\mathbf{E_1}$ introduces an instance

$$\mathsf{E_1}\ \frac{}{K \rightsquigarrow A_1 \quad \ldots \quad K \rightsquigarrow A_n} : \begin{array}{l} A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B_1 \sqcup \ldots \sqcup B_m \in C \\ K = A_1 \sqcap \ldots \sqcap A_n \end{array}$$

$$\mathsf{E_2}\ \frac{K \rightsquigarrow A_1 \quad \ldots \quad K \rightsquigarrow A_n}{K \rightsquigarrow B_1 \quad \ldots \quad K \rightsquigarrow B_m} : A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B_1 \sqcup \ldots \sqcup B_m \in O$$

$$\mathsf{E_3}\ \frac{K \rightsquigarrow A}{B \rightsquigarrow B} : A \sqsubseteq \exists R.B \in O$$

$$\mathsf{E_4}\ \frac{K \rightsquigarrow A \quad K \rightsquigarrow C}{B \rightsquigarrow D} : \begin{array}{l} A \sqsubseteq \exists R.B \in O \\ C \sqsubseteq \forall R.D \in O \end{array} \qquad \mathsf{E_5}\ \frac{K \rightsquigarrow A \quad B \rightsquigarrow E}{K \rightsquigarrow F} : \begin{array}{l} A \sqsubseteq \exists R.B \in O \\ E \sqsubseteq \forall R^{-}.F \in O \end{array}$$

Fig. 3. Computing the deductive overestimation for O and C

of all atomic concepts in K. Rule $\mathsf{E_2}$ addresses the fact that, if some object α is an instance of $A_1, \ldots, A_n$ and O contains a clause $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B_1 \sqcup \ldots \sqcup B_m$, then the object must be an instance of some B_i. Since a polynomial overestimation method that reasons by case is unlikely to exist, rule $\mathsf{E_2}$ overestimates the subsumption relation by saying that α can be an instance of all $B_1, \ldots, B_m$. Rule $\mathsf{E_3}$ takes into account that, given $A \sqsubseteq \exists R.B \in O$, each instance of A needs an R-successor that is an instance of B. Analogously to the $\mathcal{EL}$ reasoning calculus, in order to obtain a polynomial overestimation method, rule $\mathsf{E_3}$ "reuses" the same successor to satisfy multiple existential restrictions to the same concept B. Finally, rules $\mathsf{E_4}$ and $\mathsf{E_5}$ implement modal reasoning.

Having computed $\rightsquigarrow$, we construct the decomposition $\mathcal{D}_{\mathsf{E}} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig} \rangle$ of the symbols occurring in O and C as shown below. Note that $\mathcal{D}_{\mathsf{E}}$ contains no ϵ-labeled edges, as this decomposition method does not analyze or-branching. By Theorems 1 and 3, the resolution calculus for $\mathcal{D}_{\mathsf{E}}$ and O is C-complete.

$$\begin{array}{ll} \mathcal{V} := \{v_K \mid K \rightsquigarrow A \text{ for some } A\} & \mathsf{sig}(v_K) := \{A \mid K \rightsquigarrow A\} \\ \mathcal{E} := \{\langle v_B, v_K, R \rangle \mid K \rightsquigarrow A \text{ and } A \sqsubseteq \exists R.B \in O\} & \end{array}$$

Theorem 3. *Decomposition $\mathcal{D}_{\mathsf{E}}$ is admissible for O and C.*

4.2 Analyzing Or-Branching via Tree Decomposition

We now present an approach for computing admissible decompositions that analyzes or-branching. The approach handles the clauses in O as explained in Section 1 for SAT, and it imposes additional constraints in order to satisfy condition (iv) of Definition 3.

Given a normalized ontology O and a set of clauses C, we define the hypergraph $G_{O,C} = \langle V, H \rangle$ such that V and H are the smallest sets satisfying the following properties. For each atomic concept A occurring in O or C, we have $A \in V$. For each clause $K \sqsubseteq M \in O$, we have $\mathsf{sig}(K \sqsubseteq M) \in H$. For each $A \sqsubseteq \exists R.B \in O$, set H contains hyperedges $\mathsf{dom}_{A \sqsubseteq \exists R.B}$ and $\mathsf{ran}_{A \sqsubseteq \exists R.B}$ defined as shown below, where $C_i \sqsubseteq \forall R.D_i$, $1 \leq i \leq n$ and $E_j \sqsubseteq \forall \mathsf{inv}(R).F_j$, $1 \leq j \leq m$ are all axioms in O of the respective forms:

$$\begin{aligned} \mathsf{dom}_{A \sqsubseteq \exists R.B} &:= \{A, C_1, \ldots, C_n, F_1, \ldots, F_m\}, \\ \mathsf{ran}_{A \sqsubseteq \exists R.B} &:= \{B, D_1, \ldots, D_n, E_1, \ldots, E_m\}. \end{aligned}$$

Finally, $\mathsf{sig}(K \sqsubseteq M) \in H$ for each $K \sqsubseteq M \in C$.

Given a tree decomposition $\langle T, L\rangle$ of $G_{O,C}$, we construct (don't-care nondeterministically) a decomposition $\mathcal{D}_\mathsf{T} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig}\rangle$ as follows. The vertices of $\mathcal{D}_\mathsf{T}$ are the bags of T—that is, $\mathcal{V} := \mathsf{B}(T)$. The signatures of $\mathcal{D}_\mathsf{T}$ are the labels of T—that is, $\mathsf{sig} := L$. The ϵ-edges of $\mathcal{D}_\mathsf{T}$ are the edges of T—that is, for each $\{u, v\} \in \mathsf{E}(T)$, we have $\langle u, v, \epsilon\rangle \in \mathcal{E}$. Finally, for each $A \sqsubseteq \exists R.B \in O$, choose vertices $u, v \in \mathcal{V}$ such that $\mathsf{ran}_{A\sqsubseteq\exists R.B} \subseteq L(u)$ and $\mathsf{dom}_{A\sqsubseteq\exists R.B} \subseteq L(v)$ and set $\langle u, v, R\rangle \in \mathcal{E}$; such u and v exist due to property (T2) of the definition of tree decompositions in Section 1.

Theorem 4. *Every decomposition $\mathcal{D}_\mathsf{T}$ is admissible for O and C.*

4.3 Analyzing And- and Or-Branching Simultaneously

We now show how to combine the approaches for analyzing and- and or-branching to obtain a C-*decomposition* of a normalized $\mathcal{ALCI}$ ontology O and a set of clauses C.

The procedure consists of three steps. First, we compute the relation $\rightsquigarrow$ as described in Section 4.1. This step analyzes the and-branching inherent in O and C.

Second, for all K such that $K \rightsquigarrow A$ for some A, we simultaneously define hypergraphs $G_K = \langle V_K, H_K\rangle$ where $V_K := \{A \mid K \rightsquigarrow A\}$, and H_K are the smallest sets satisfying the following conditions. For each clause $K' \sqsubseteq M' \in O$ with $\mathsf{sig}(K' \sqsubseteq M') \subseteq V_K$, we have $\mathsf{sig}(K' \sqsubseteq M') \in H_K$. For each axiom $A \sqsubseteq \exists R.B \in O$ such that $A \in V_K$, set H_K contains hyperedge $\mathsf{dom}_{K,A\sqsubseteq\exists R.B}$ and set H_B contains hyperedge $\mathsf{ran}_{K,A\sqsubseteq\exists R.B}$ defined below, where $C_i \sqsubseteq \forall R.D_i$, $1 \leq i \leq n$ and $E_j \sqsubseteq \forall \mathsf{inv}(R).F_j$, $1 \leq j \leq m$ are all axioms in O of the respective forms such that $C_i \in V_K$ and $E_j \in V_B$:

$$\mathsf{dom}_{K,A\sqsubseteq\exists R.B} := \{A, C_1, \ldots, C_n, F_1, \ldots, F_m\},$$
$$\mathsf{ran}_{K,A\sqsubseteq\exists R.B} := \{B, D_1, \ldots, D_n, E_1, \ldots, E_m\}.$$

Finally, $[\mathsf{sig}(K \sqsubseteq M) \cap V_K] \in H_K$ for each $K \sqsubseteq M \in C$.

Third, we compute a tree decomposition $\langle T_K, L_K\rangle$ for each hypergraph G_K; without loss of generality we assume that all sets $\mathsf{B}(T_K)$ are disjoint. We then construct the decomposition $\mathcal{D}_\mathsf{C} = \langle \mathcal{V}, \mathcal{E}, \mathsf{sig}\rangle$ as follows. The vertices of $\mathcal{D}_\mathsf{C}$ are the bags of the tree decompositions—that is, $\mathcal{V} := \bigcup_K \mathsf{B}(T_K)$. The signatures of $\mathcal{D}_\mathsf{C}$ are the labels of the tree decompositions—that is, $\mathsf{sig} := \bigcup_K L_K$. The ϵ-edges of $\mathcal{D}_\mathsf{C}$ are the edges of the tree decompositions—that is, $\langle u, v, \epsilon\rangle \in \mathcal{E}$ for each $\{u, v\} \in \mathsf{E}(T_K)$. Finally, for each axiom $A \sqsubseteq \exists R.B \in O$ and each K such that $A \in V_K$, choose $u \in \mathsf{B}(V_B)$ and $v \in \mathsf{B}(V_K)$ such that $\mathsf{ran}_{K,A\sqsubseteq\exists R.B} \subseteq L(u)$ and $\mathsf{dom}_{K,A\sqsubseteq\exists R.B} \subseteq L(v)$ and set $\langle u, v, R\rangle \in \mathcal{E}$; such u and v exist due to property (T2) of the definition of tree decompositions in Section 1.

The class of all C-*decompositions* of O and C consists of all decompositions obtained in the way specified above. Note that the first step (computation of $\rightsquigarrow$) is deterministic, but the second step is not as each G_K may admit several tree decompositions. The C-*width* of O and C is the minimal width of any C-decomposition of O and C.

Theorem 5. *Every decomposition $\mathcal{D}_\mathsf{C}$ is admissible for O and C.*

To show that DL reasoning is FPT if the C-width is bounded, we next estimate the effort required for computing a C-decomposition of O and C. With $||O||$ and $||C||$ we denote the sizes of (i.e. the numbers of symbols required to encode) O and C, respectively.

Table 1. Upper bounds on C-width for classification

Ontology	$\|\Sigma_A\|$	$\|\Sigma_A^{norm}\|$	$\mathsf{wd}(\mathcal{D}_\mathsf{E})$	$\mathsf{wd}(\mathcal{D}_\mathsf{C})$
SNOMED CT (http://ihtsdo.org/snomed-ct/)	315,489	516,703	349	100
SNOMED CT-SEP (see [9] for reference)	54,973	149,839	1,196	168
FMA (http://fma.biostr.washington.edu/)	41,700	81,685	1,166	35
GALEN (http://opengalen.org/)	23,136	49,245	646	54
OBI (http://obi-ontology.org/)	2,955	4,296	304	45

Proposition 2. *An algorithm exists that takes as input a positive integer d, a normalized $\mathcal{ALCI}$ ontology $\mathcal{O}$, and a set of clauses C, that runs in time $O(g(d) \cdot (\|\mathcal{O}\| + \|C\|)^5)$ for g a computable function, and that computes a C-decomposition of $\mathcal{O}$ and C of width at most d whenever at least one such decomposition exists.*

We can now formulate the main FPT result for C-decompositions.

Theorem 6. *Let d be a positive integer, let $\mathcal{O}$ be a normalized $\mathcal{ALCI}$ ontology, and let $K \sqsubseteq M$ be a clause. The problem of deciding whether a C-decomposition of $\mathcal{O}$ and $C = \{K \sqsubseteq M\}$ of width at most d exists, and if so, whether $\mathcal{O} \models K \sqsubseteq M$, is FPT.*

5 Experimental Results

It can be argued that FPT is interesting only if the parameter can be substantially smaller than the input size. In order to judge the "usefulness" of C-width as a complexity measure, we measured the C-width of several ontologies (listed in Table 1) that are often used for evaluating DL reasoners. We weakened all ontologies to $\mathcal{ALCHI}$ by discarding all unsupported features, we applied the structural transformation from [9], and we eliminated role inclusion axioms by unfolding the role hierarchy into universal restrictions to obtain normalized $\mathcal{ALCI}$ ontologies. Note that there are several different ways of formulating and optimizing structural transformation, and each could produce an ontology of a different C-width, so our results are not necessarily optimal.

After normalization, we next computed the deductive overestimation $\rightsquigarrow$ and the decomposition $\mathcal{D}_\mathsf{E}$ as described in Section 4.1, we constructed the hypergraphs G_K as described in Section 4.3, and we fed all of them into TreeD[1]—a library for computing tree decompositions—to construct a C-decomposition $\mathcal{D}_\mathsf{C}$. For each ontology we considered two sets of goal clauses: $C_1 = \{A \sqsubseteq \bot \mid A \in \Sigma_A\}$, which corresponds to checking satisfiability of all atomic concepts, and $C_2 = \{A \sqsubseteq B \mid A, B \in \Sigma_A\}$, which corresponds to classification. In theory, the C-width of $\mathcal{O}$ and C_1 can be smaller than the C-width of $\mathcal{O}$ and C_2; however, we have not observed a difference between the two in practice, so we present here only the results for classification. Also, please note that TreeD was able only to produce approximate, rather than exact tree decompositions; hence, our results provide only an upper bound on the C-width.

The results of our experiments are shown in Table 1. For each ontology we list the number of atomic concepts in the original ontology ($|\Sigma_A|$), the number of atomic

[1] http://www.itu.dk/people/sathi/treed/

concepts after normalization ($|\Sigma_A^{norm}|$), and the widths of the two decompositions that we constructed. Notice that although some of the tested ontologies contain tens or even hundreds of thousands of concepts, the width of $\mathcal{D}_{\mathbf{C}}$ rarely exceeds one hundred, and it is always by several orders of magnitude smaller than the total number of concepts in the ontology. This suggests that our notion of a decomposition might even prove to be useful in practice, provided that our resolution algorithm is suitably optimized.

6 Conclusion

We presented a DL reasoning algorithm that is fixed parameter tractable for a suitable notion of the input width. We see two main challenges for our future work. On the theoretical side, our approach should be extended to more complex ontology languages; handling counting seems particularly challenging. On the practical side, our algorithm should be optimized for practical use. A particular challenge is to combine the construction of a decomposition with actual reasoning and thus save preprocessing time.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite Family and Relations. Journal of Artificial Intelligence Research 36, 1–69 (2009)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In: Kaelbling, L.P., Saffiotti, A. (eds.) Proc. of the 19th Int. Joint Conference on Artificial Intelligence (IJCAI 2005). pp. 364–369. Morgan Kaufmann Publishers, Edinburgh, UK (July 30–August 5 2005)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2nd edn. (August 2007)
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. Journal of Automated Reasoning 9, 385–429 (2007)
5. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (1999)
6. Gottlob, G., Pichler, R., Wei, F.: Bounded Treewidth as a Key to Tractability of Knowledge Representation and Reasoning. In: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006). pp. 250–256. AAAI Press, Boston, MA, USA (2006)
7. Gottlob, G., Scarcello, F., Sideri, M.: Fixed-parameter complexity in AI and nonmonotonic reasoning. Artificial Intelligence 138(1–2), 55–86 (2002)
8. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic. In: Proc. of the 12th Int. World Wide Web Conference (WWW 2003). pp. 48–57. ACM Press, Budapest, Hungary (May 20–24 2003)
9. Simančík, F., Kazakov, Y., Horrocks, I.: Consequence-Based Reasoning beyond Horn Ontologies. In: Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011) (July 16–22 2011), to appear
10. Szeider, S.: On Fixed-Parameter Tractable Parameterizations of SAT. In: Giunchiglia, E., Tacchella, A. (eds.) Proc. of the 6th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2003), Selected Revised Papers. LNCS, vol. 2919, pp. 188–202. Springer, Santa Margherita Ligure, Italy (May 5–8 2003)