Mining $\mathcal{EL}^{\perp}$ Bases with Adaptable Role Depth

Ricardo Guimarães,[1] Ana Ozaki,[1] Cosimo Persia,[1] Baris Sertkaya[2]
[1] Department of Informatics, University of Bergen, Norway
[2] Frankfurt University of Applied Sciences, Germany
Ricardo.Guimaraes@uib.no, Ana.Ozaki@uib.no, Cosimo.Persia@uib.no, Sertkaya@fb2.fra-uas.de

Abstract

In Formal Concept Analysis, a base for a finite structure is a set of implications that characterizes all valid implications of the structure. This notion can be adapted to the context of Description Logic, where the base consists of a set of concept inclusions instead of implications. In this setting, concept expressions can be arbitrarily large. Thus, it is not clear whether a finite base exists and, if so, how large concept expressions may need to be. We first revisit results in the literature for mining $\mathcal{EL}^{\perp}$ bases from finite interpretations. Those mainly focus on finding a finite base or on fixing the role depth but potentially losing some of the valid concept inclusions with higher role depth. We then present a new strategy for mining $\mathcal{EL}^{\perp}$ bases which is adaptable in the sense that it can bound the role depth of concepts depending on the local structure of the interpretation. Our strategy guarantees to capture *all* $\mathcal{EL}^{\perp}$ concept inclusions holding in the interpretation, not only the ones up to a fixed role depth.

1 Introduction

Among its many applications in artificial intelligence, logic is used to formally represent knowledge. Such knowledge, often in the form of facts and rules, enables machines to process complex relational data, deduce new knowledge from it, and extract hidden relationships in a specific domain. A well-studied formalism for knowledge representation is given by a family of logics known as description logics (DLs) (Baader et al. 2017). DL is the logical formalism behind the design of many knowledge-based applications. However, it is often difficult and time-consuming to manually model in a formal language rules and constraints that hold in a domain of knowledge.

In this work, we consider an automatic method to extract rules (concept inclusions (CIs)) formulated in DL from data. This data can be, for instance, a collection of facts in a database or a knowledge graph. For instance, in the DBpedia knowledge graph (Lehmann et al. 2015), one can represent the relationship between a city 'a' and the region 'b' it belongs to with the facts $\mathsf{city}(\mathsf{a})$, $\mathsf{region}(\mathsf{b})$, $\mathsf{partof}(\mathsf{a}, \mathsf{b})$, and $\mathsf{capital}(\mathsf{b}, \mathsf{a})$. From this data, one can mine a CI expressing that a capital is a city that is part of a region.

To mine CIs that hold in a dataset, we combine notions of Formal Concept Analysis (FCA) (Ganter and Wille 1999) and DLs. FCA is a subfield of lattice theory that provides methods for analysing datasets and identifying the dependencies in them. In FCA a dataset, also called a *formal context*, is a table showing which objects have which attributes. Given a formal context, FCA methods are used to extract the dependencies between the attributes, also called implications (Figure 1). A *base* is a set of implications that entails every valid implication of the dataset and only those (soundness and completeness). It can be used for detecting erroneous or missing items in the dataset (Baader et al. 2007). In the DL setting, a base is a set of CIs (an ontology) which can serve as a starting point for ontology engineers to build an ontology in a domain of interest.

However, for some DLs and datasets, it may happen that no finite base exists. Cyclic relationships are common in knowledge graphs and they are the main challenge for finding a finite DL base. With only one cyclic relationship, we already have that infinitely many concepts hold in the dataset. Strategies for limiting the size of concepts in the presence of cyclic dependencies have already been investigated in the literature. Baader and Distel (2008) and Distel (2011) propose a way of mining DL finite bases expressible in the DL $\mathcal{EL}^{\perp}_{gfp}$ which is the addition of greatest fixpoint semantics to the DL language $\mathcal{EL}^{\perp}$. The semantics offered by $\mathcal{EL}^{\perp}_{gfp}$ elegantly solves the difficulty of mining CIs from cyclic relationships in the data. However, this semantics comes with two drawbacks. Firstly, $\mathcal{EL}^{\perp}_{gfp}$ concepts may be difficult to understand, and learned CIs may be too complex to validate by domain experts. Secondly, there is no efficient implementation of a reasoner for $\mathcal{EL}^{\perp}_{gfp}$, even though the reasoning complexity is tractable, like for $\mathcal{EL}^{\perp}$. The authors also show how to transform an $\mathcal{EL}^{\perp}_{gfp}$ base into an $\mathcal{EL}$ base. However, it is far from being trivial to avoid the step of creating an $\mathcal{EL}^{\perp}_{gfp}$ base in their approach.

A simplification of the mentioned work has been proposed by Borchmann, Distel, and Kriegel (2016) where they show how to mine $\mathcal{EL}^{\perp}$ finite bases with a predefined and fixed role depth for concept expressions. As a consequence, the base is sound and complete only w.r.t. CIs containing concepts with bounded role depth. Their approach avoids

	City	Region	$\exists$partof.$\top$	Settlement
a	×		×	×
b	×		×	×
c		×	×	

Figure 1: (a) A dataset with 4 attributes and 3 objects. (b) The implications City → $\exists$partof.$\top$ and City → Settlement hold in the dataset but City → Region does not hold in it.

the step of creating an $\mathcal{EL}^{\bot}_{gfp}$ base but also avoids the main challenge in creating a finite base for $\mathcal{EL}^{\bot}$, which is the fact that the role depth of concepts can be arbitrarily large.

Our work brings together the best of the approaches by Distel (2011) and Borchmann, Distel, and Kriegel (2016): we *directly* compute a finite $\mathcal{EL}^{\bot}$ base that captures the *whole* language (not only up to a certain role depth). In particular, we present a new approach for computing the role depth of concepts which *adapts* depending on the objects considered during the computation of CIs.

Related work. Several authors have worked on combining FCA and DLs or on applying methods from one field to the other (Ozaki 2020). Baader (1995) uses FCA to compute the subsumption hierarchy of the conjunction of predefined concepts. Rudolph (2004; 2006) uses the DL $\mathcal{FLE}$ for the definition of FCA attributes and FCA techniques for generating a knowledge base. Baader et al. (2007) uses FCA for completing missing knowledge in a DL knowledge base. Baader et al. (2000) proposes a method for building DL ontologies through the interaction of domain experts. Sertkaya (2010) presents a survey on applications of FCA methods in DLs. Borchmann and Distel (2011) provide a practical application of the theory developed by Distel (2011) on knowledge graphs. Borchmann (2014) shows how a base of confident $\mathcal{EL}^{\bot}_{gfp}$ concept inclusions can be extracted from a DL interpretation. Monnin et al. (2017) compare, using FCA techniques, data present in DBpedia with the constraints of a given ontology to check if the data is compliant with it. Kriegel (2019a) among other contributions extends the results by Borchmann, Distel, and Kriegel (2016) to a logic that is more expressive than $\mathcal{EL}^{\bot}$. He also investigates the same problem for probabilistic DLs (Kriegel 2019b).

In the next section, we present basic definitions and notation. In Section 3, we present the problem of mining $\mathcal{EL}^{\bot}$ CIs and establish lower bounds for this problem. In Section 4, we present our main result for mining $\mathcal{EL}^{\bot}$ bases with adaptable role depth. Our result uses a notion that relates each vertex in a graph to a set of vertices, called *maximum vertices from* (MVF). In Section 5, we show that the MVF of a vertex in a graph can be computed in linear time in the size of the graph. Missing proofs can be found in the long version (Guimarães et al. 2021).

2 Preliminaries

We introduce the syntax and semantics of $\mathcal{EL}^{\bot}$ and basic definitions related to description graphs used in the paper.

The Description Logic $\mathcal{EL}^{\bot}$

$\mathcal{EL}^{\bot}$ (Baader, Brandt, and Lutz 2005) is a lightweight DL, which only allows for expressing conjunctions and existential restrictions. Despite this rather low expressive power, slight extensions of it have turned out to be highly successful in practical applications, especially in the medical domain (Spackman, Campbell, and Cote 1997).

We use two *finite* and *disjoint* sets, $\mathsf{N_C}$ and $\mathsf{N_R}$, of *concept* and *role* names to define the syntax and semantics of $\mathcal{EL}^{\bot}$. $\mathcal{EL}^{\bot}$ *concept expressions* are built according to the grammar rule $C, D ::= A \mid \top \mid \bot \mid C \sqcap D \mid \exists r.C$ with $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. We write $\exists r^{n+1}.C$ as a shorthand for $\exists r.(\exists r^{n}.C)$ where $\exists r^{1}.C := \exists r.C$. An $\mathcal{EL}^{\bot}$ *TBox* is a finite set of *concept inclusions* (CIs) $C \sqsubseteq D$, where C, D are $\mathcal{EL}^{\bot}$ concept expressions. We may omit '$\mathcal{EL}^{\bot}$' when we speak of concept expressions, CIs, and TBoxes, if this is clear from the context. We may write $C \equiv D$ (an equivalence) as a short hand for when we have both $C \sqsubseteq D$ and $D \sqsubseteq C$. The *signature* of a concept expression, a CI, or a TBox is the set of concept and role names occurring in it.

The semantics of $\mathcal{EL}^{\bot}$ is based on *interpretations*. An interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set, called the *domain of* $\mathcal{I}$, and $\cdot^{\mathcal{I}}$ is a function mapping each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each $r \in \mathsf{N_R}$ to a subset $r^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ extends to arbitrary $\mathcal{EL}^{\bot}$ concept expressions as usual:

$$(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}} \quad (\top)^{\mathcal{I}} := \Delta^{\mathcal{I}} \quad (\bot)^{\mathcal{I}} := \varnothing$$
$$(\exists r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$$

An interpretation $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$, in symbols $\mathcal{I} \models C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. It satisfies a TBox $\mathcal{T}$ if it satisfies all CIs in $\mathcal{T}$. A TBox $\mathcal{T}$ *entails* a CI $C \sqsubseteq D$, written $\mathcal{T} \models C \sqsubseteq D$, iff all interpretations satisfying $\mathcal{T}$ also satisfy $C \sqsubseteq D$. We write $\Sigma_{\mathcal{I}}$ for the set of concept or role names X such that $X^{\mathcal{I}} \neq \varnothing$. A *finite interpretation* is an interpretation with a finite domain.

Description Graphs, Products, and Unravellings

We also use the notion of description graphs (Baader 2003). The *description graph* $\mathcal{G}(\mathcal{I}) = (V_{\mathcal{I}}, E_{\mathcal{I}}, L_{\mathcal{I}})$ of an interpretation $\mathcal{I}$ is defined as (e.g. Figure 4):

1. $V_{\mathcal{I}} = \Delta^{\mathcal{I}}$;
2. $E_{\mathcal{I}} = \{(x, r, y) \mid r \in \mathsf{N_R} \text{ and } (x, y) \in r^{\mathcal{I}}\}$;
3. $L_{\mathcal{I}}(x) = \{A \in \mathsf{N_C} \mid x \in A^{\mathcal{I}}\}$.

The *description tree* of an $\mathcal{EL}^{\bot}$ concept expression C over the signature Σ is the finite directed tree $\mathcal{G}(C) = (V_C, E_C, L_C)$ where V_C is the set of nodes, $E_C \subseteq V_C \times \mathsf{N_R} \times V_C$ is the set of edges, and $L_C : V \to 2^{\mathsf{N_C}}$ is the labelling function. $\mathcal{G}(C)$ is defined inductively:

1. for $C = \top$, $V_C = \{\rho_C\}$ and $L_C(\rho_C) = \varnothing$ where ρ_C is the root node of the tree;
2. for $C = A \in \mathsf{N_C}$, $V_C = \{\rho_C\}$ and $L_C(\rho_C) = A$;
3. for $C = D_1 \sqcap D_2$, $\mathcal{G}(C)$ is obtained by merging the roots ρ_{D_1}, ρ_{D_2} in one ρ_C with $L_C(\rho_C) = L_{D_1}(\rho_{D_1}) \cup L_{D_2}(\rho_{D_2})$;

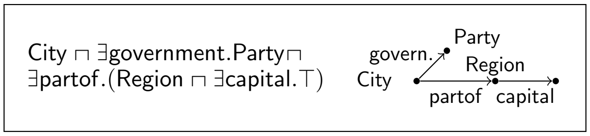

Figure 2: A concept expression and its description graph.

4. for $C = \exists r.D$, $\mathcal{G}(C)$ is built from $\mathcal{G}(D)$ by adding a new node (root) ρ_C to V_D and an edge (ρ_C, r, ρ_D) to E_D.

The *concept expression* (unique up to logical equivalence) $\mathbf{C}(\mathcal{G}_v)$ of a tree shaped graph $\mathcal{G}_v = (V, E, L)$ rooted in v is

$$\prod_{i=1}^{k} P_i \sqcap \prod_{j=1}^{l} \exists r_j.\mathbf{C}(\mathcal{G}_{w_j}),$$

where $L(v) = \{P_i \mid 1 \leqslant i \leqslant k\}$, $(v, r_j, w_j) \in E$ (and there are l such edges) and $\mathbf{C}(\mathcal{G}_{w_j})$ is inductively defined, with $\mathcal{G}_{w_j}$ being the subgraph of $\mathcal{G}$ rooted in w_j.

A *walk* in a description graph $\mathcal{G} = (V, E, L)$ between two nodes $u, v \in V$ is a word $\mathbf{w} = v_0 r_0 v_1 r_1 \ldots r_{n-1} v_n$ where $v_0 = u$, $v_n = v$, $v_i \in V$, $r_i \in \mathsf{N_R}$ and $(v_i, r_i, v_{i+1}) \in E$ for all $i \in \{0, \ldots, n-1\}$. The length of $\mathbf{w}$ in this case is n, in symbols, $|\mathbf{w}| = n$. Walks with length $n = 0$ are possible, it means that the walk has just one vertex (no edges). Vertices and edges may occur multiple times in a walk. Let $\mathcal{G} = (V, E, L)$ be an $\mathcal{EL}^{\bot}$ description graph with $x \in V$ and $d \in \mathbb{N}$. Denote by $\delta(\mathbf{w})$ the last vertex in the walk $\mathbf{w}$. The *unravelling* of $\mathcal{G}$ up to depth d is the description graph $\mathcal{G}^x_d = (V_d, E_d, L_d)$ starting at node x defined as follows:

1. V_d is the set of all directed walks in $\mathcal{G}$ that start at x and have length at most d;
2. $E_d = \{(\mathbf{w}, r, \mathbf{w}rv) \mid v \in V, r \in \mathsf{N_R}, \mathbf{w}, \mathbf{w}rv \in V_d\}$;
3. $L_d(\mathbf{w}) = L(\delta(\mathbf{w}))$.

A *path* is a walk where vertices do not repeat.

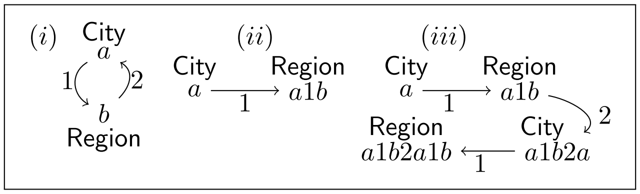

Figure 3: Unravelling of the description graph of the interpretation $\mathcal{I}$ in (i). For readability, partof has been replaced with symbol 1 and capital with symbol 2. (ii) depicts $\mathcal{G}(\mathcal{I})^a_1$ and (iii) depicts $\mathcal{G}(\mathcal{I})^a_3$.

Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be n description graphs such that $\mathcal{G}_i = (V_i, E_i, L_i)$. Then the *product* of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is the description graph (V, E, L) defined as:

1. $V = \times_{i=1}^{n} V_i$;
2. $E = \{((v_1, \ldots, v_n), r, (w_1, \ldots, w_n)) \mid r \in \mathsf{N_R}, (v_i, r, w_i) \in E_i, \text{ for all } 1 \leqslant i \leqslant n\}$;
3. $L(v_1, \ldots, v_n) = \bigcap_{i=1}^{n} L_i(v_i)$.

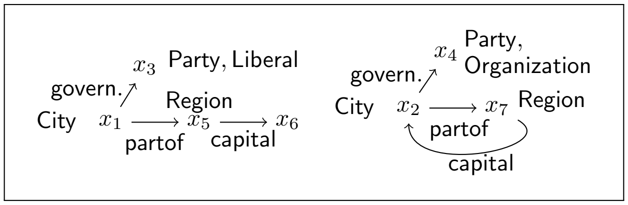

Figure 4: Description graph of the interpretation $\mathcal{I} = \{\{x_1, \cdots, x_7\}, \cdot^{\mathcal{I}}\}$ where $\{x_1, x_2\} = \mathsf{City}^{\mathcal{I}}$, $\{x_3, x_4\} = \mathsf{Party}^{\mathcal{I}}$, $\{(x_1, x_5), (x_2, x_7)\} = \mathsf{partof}^{\mathcal{I}}$, etc.

If each $\mathcal{G}_i$ is a tree with root v_i then we denote by $\prod_{i=1}^{n} \mathcal{G}_i$ the tree rooted in $(v_1, \ldots, v_n)$ contained in the product graph of $\mathcal{G}_1, \ldots, \mathcal{G}_n$.

3 Mining $\mathcal{EL}^{\bot}$ Bases

The set of all $\mathcal{EL}^{\bot}$ CIs that are satisfied by an interpretation $\mathcal{I}$ is in general infinite because whenever $\mathcal{I} \models C \sqsubseteq D$, $\mathcal{I} \models \exists r.C \sqsubseteq \exists r.D$ as well. Therefore one is interested in a finite and small set of CIs that entails the whole set of valid CIs. For mining such a set of CIs from a given interpretation we employ ideas from FCA and recall literature results.

Definition 1. A TBox $\mathcal{T}$ is a *base* for a finite interpretation $\mathcal{I}$ and a DL language L, if for every CI $C \sqsubseteq D$, formulated within L and $\Sigma_{\mathcal{I}}$: $\mathcal{I} \models C \sqsubseteq D$ iff $\mathcal{T} \models C \sqsubseteq D$.

We say that a DL has the *finite base property* (FBP) if, for all finite interpretations $\mathcal{I}$, there is a finite base with CIs formulated within the DL language and $\Sigma_{\mathcal{I}}$. Not all DLs have the finite base property. Consider for instance the fragments $\mathcal{EL}^{\bot}_{rhs}$ (and $\mathcal{EL}^{\bot}_{lhs}$) of $\mathcal{EL}^{\bot}$ that allows only concept names on the left-hand (right-hand) side but *complex* $\mathcal{EL}^{\bot}$ concept expressions on the right-hand (left-hand) side of CIs.

Proposition 1. $\mathcal{EL}^{\bot}_{rhs}$ and $\mathcal{EL}^{\bot}_{lhs}$ do not have the FBP.

Proof Sketch. No finite base $\mathcal{EL}^{\bot}_{rhs}$ exists for the interpretation in Figure 5 (i). For every $n \geqslant 1$, the $\mathcal{EL}^{\bot}_{rhs}$ base should entail the CI $A \sqsubseteq \exists r^n.\top$. Similarly, no finite $\mathcal{EL}^{\bot}_{lhs}$ base exists for the interpretation in Figure 5 (ii). For every $n \geqslant 1$, the $\mathcal{EL}^{\bot}_{lhs}$ base should entail the CI $\exists s.\exists r^n.B \sqsubseteq A$. □

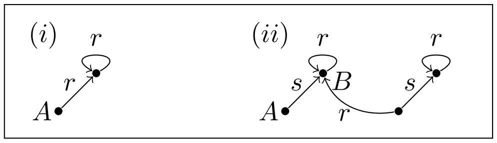

Figure 5: Lack of the FBP for $\mathcal{EL}^{\bot}_{rhs}$ (i) and $\mathcal{EL}^{\bot}_{lhs}$ (ii).

The main difficulty in creating an $\mathcal{EL}^{\bot}$ base is knowing how to define the role depth of concept expressions in the base. In a finite interpretation, an arbitrarily large role depth means the presence of a cyclic structure in the interpretation. However, $\mathcal{EL}^{\bot}$ concept expressions cannot express cycles. The difficulty can be overcomed by extending $\mathcal{EL}^{\bot}$ with greatest fix-point semantics. It is known that the resulting DL, called $\mathcal{EL}^{\bot}_{gfp}$, has the FBP (Baader and Distel 2008; Distel 2011). The authors then show how to transform an $\mathcal{EL}^{\bot}_{gfp}$ base into an $\mathcal{EL}^{\bot}$ base, thus, establishing that $\mathcal{EL}^{\bot}$ also enjoys the FBP.

In the following, we show that, although finite, the role depth of a base for $\mathcal{EL}^{\bot}$ and a (finite) interpretation $\mathcal{I}$ can be exponential in the size of $\mathcal{I}$.

Example 1. Consider $\mathcal{I}$ represented in the shaded area in Figure 6. For $p_1 = 2, p_2 = 3, p_3 = 5$ and for all $k \in \mathbb{N}^+$, we have that $x_i \in (\exists r^{k \cdot p_i - 1}.A)^{\mathcal{I}}$, where $1 \leqslant i \leqslant 3$. We know that $30 = min(\bigcap_{i=1}^{3}\{k \cdot p_i \mid k \in \mathbb{N}^+\}) = \prod_{i=1}^{n} p_i$ (which is the least common multiple). We also know that for any $n, p \in \mathbb{N}^+$, $n + 1$ is a multiple of p iff n is a multiple of p minus 1. Therefore, the number

$$d = min(\bigcap_{i=1}^{3}\{k \cdot p_i - 1 \mid k \in \mathbb{N}^+\}),$$

such that $\{x_1, x_2, x_3\} = B^{\mathcal{I}} = (\exists r^d.A)^{\mathcal{I}}$, is $\prod_{i=1}^{3} p_i - 1 = 29$. A base for $\mathcal{I}$ should have the CI with role depth at least d because it has to entail the CI $B \sqsubseteq \exists r^d.A$.

Theorem 1. There is a finite interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that any $\mathcal{EL}^{\bot}$ base for $\mathcal{I}$ has a concept expression with role depth exponential in the size of $\mathcal{I}$.

Proof Sketch. We can generalise Example 1 to the case where we have an interpretation $\mathcal{J}$ that for an arbitrary $n > 1$, and for every $i \in \{1, \cdots, n\}$ and $k \in \mathbb{N}^+$, there is an $x \in \Delta^{\mathcal{J}}$ that satisfies $x \in (\exists r^{k \cdot p_i - 1}.A)^{\mathcal{J}}$ where p_i is the i-th prime number. In this case, the minimal role depth of concepts in any base for $\mathcal{J}$ must be $d \geqslant \prod_{i=1}^{n} p_i - 1 \geqslant 2^n$. □

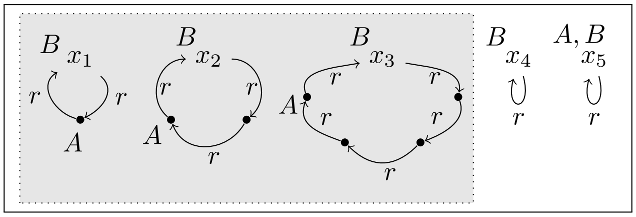

Figure 6: Description graph of an interpretation $\mathcal{I}$. Let $X = \{x_1, x_2, x_3\}$. For all $d < 29$ we have $x_4 \in \mathbf{C}\,(\prod_{x \in X} \mathcal{G}(\mathcal{I})_d^x)^{\mathcal{I}} = (B \sqcap \exists r^d.\top)^{\mathcal{I}}$. However, for all $k \geqslant 29$, $x_4 \notin \mathbf{C}\,(\prod_{x \in X} \mathcal{G}(\mathcal{I})_k^x)^{\mathcal{I}}$ since $x_4 \notin (\exists r^{29}.A)^{\mathcal{I}}$.

In addition to the role depth of the concept expressions in the base, the size of the base itself can also be exponential in the size of the data given as input, which is a well-known result in classical FCA (Kuznetsov 2004). The DL setting is more challenging than classical FCA, and so, this lower bound also holds in the problem we consider. In Section 4, we present our definition of an $\mathcal{EL}^{\bot}$ base for a finite interpretation $\mathcal{I}$ and highlight cases in which the role depth is polynomial in the size of $\mathcal{I}$.

4 Adaptable Role Depth

We present in this section our main result which is our strategy to construct $\mathcal{EL}^{\bot}$ bases with adaptable role depth. To define an $\mathcal{EL}^{\bot}$ base, we use the notion of a model-based most specific concept (MMSC) up to a certain role depth. The MMSC plays a key rôle in the computation of a base from a given finite interpretation.

Definition 2. An $\mathcal{EL}^{\bot}$ concept expression C is a model-based most specific concept of $X \subseteq \Delta^{\mathcal{I}}$ with role depth $d \geqslant 0$ iff (1) $X \subseteq C^{\mathcal{I}}$, (2) C has role depth at most d, and (3) for all $\mathcal{EL}^{\bot}$ concept expressions D with role depth at most d, if $X \subseteq D^{\mathcal{I}}$ then $\varnothing \models C \sqsubseteq D$.

For a given $X \subseteq C^{\mathcal{I}}$ and a role depth d there may be multiple MMSCs (always at least one (Borchmann, Distel, and Kriegel 2016)) but they are logically equivalent. So we write '*the*' MMSC of X with role depth d (in symbols $\mathsf{mmsc}\,(X, \mathcal{I}, d)$), meaning a representative of such class of concepts. As a consequence of Definition 2, if $X = \varnothing$ then $\mathsf{mmsc}\,(X, \mathcal{I}, d) \equiv \bot$ for any interpretation $\mathcal{I}$ and $d \in \mathbb{N}$.

Example 2. Consider the interpretation $\mathcal{I}$ in Figure 4 and let $X = \{x_1, x_2\}$. We have that $\mathsf{mmsc}\,(X, \mathcal{I}, 1)$ is equivalent to

$$\mathsf{City} \sqcap \exists \mathsf{government}.\mathsf{Party} \sqcap \exists \mathsf{partof}.\mathsf{Region}.$$

With an increasing k, the concept expression $\mathsf{mmsc}\,(X, \mathcal{I}, k)$ can become more and more specific. Indeed, $\mathsf{mmsc}\,(X, \mathcal{I}, 2)$ is equivalent to

$$\mathsf{mmsc}\,(X, \mathcal{I}, 1) \sqcap \exists \mathsf{partof}.(\mathsf{Region} \sqcap \exists \mathsf{capital}.\top)$$

which is more specific than $\mathsf{mmsc}\,(X, \mathcal{I}, 1)$. However, for any $k \geqslant 2$, $\mathsf{mmsc}\,(X, \mathcal{I}, 2) \equiv \mathsf{mmsc}\,(X, \mathcal{I}, k)$.

For a fixed d, a straightforward (and inefficient) way of computing $\mathsf{mmsc}\,(\{X\}, \mathcal{I}, d)$ would be conjoining every $\mathcal{EL}^{\bot}$ concept expression C (over $\mathsf{N_C} \cup \mathsf{N_R}$) such that $X \subseteq C^{\mathcal{I}}$ and the depth of C is bounded by d. A more elegant method for computing MMSCs is based on the product of description graphs and unravelling cyclic concept expressions up to a sufficient role depth.

The MMSC can be written as the concept expression obtained from the product of description graphs of an interpretation (Borchmann, Distel, and Kriegel 2016). Formally, if $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a finite interpretation, $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$ and a $d \geqslant 0$, then

$$\mathsf{mmsc}\,(\{X\}, \mathcal{I}, d) \equiv \mathbf{C}(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_d^{x_i}).$$

The interesting challenge is how to identify the smallest d that satisfies the property: if $x \in \mathsf{mmsc}\,(X, \mathcal{I}, d)^{\mathcal{I}}$, then $x \in \mathsf{mmsc}\,(X, \mathcal{I}, k)^{\mathcal{I}}$ for every $k > d$. In the following, we develop a method for computing MMSCs with a role depth that is suitable for building an $\mathcal{EL}^{\bot}$ base of the given interpretation. This method is based on the already mentioned MVF notion, defined as follows.

Definition 3. Given a description graph $\mathcal{G} = (V, E)$ with $u \in V$, we define the *maximum vertices from (or MVF) u in $\mathcal{G}$*, denoted $\mathsf{mvf}(\mathcal{G}, u)$, as:

$$\max\{\mathsf{v_{num}}(\mathbf{w}) \mid \mathbf{w} \text{ is a walk in } \mathcal{G} \text{ starting at } u\}$$

where $\mathsf{v}_{\mathsf{num}}(\mathbf{w})$ is the number of distinct vertices occurring in $\mathbf{w}$. Additionally, we define the function mmvf as follows:

$$\mathsf{mmvf}(\mathcal{G}) := \max_{u \in V} \mathsf{mvf}(\mathcal{G}, u).$$

In other words, MVF measures the maximum number of distinct vertices that a walk with a fixed starting point can visit in the graph.

Example 3. Consider the interpretation $\mathcal{I}$ in Figure 4. Any walk in the description graph of $\mathcal{I}$ starting at x_1 will visit at most three distinct vertices (including x_1). Although there are four vertices reachable from x_1, we have that $\mathsf{mvf}(\mathcal{G}(\mathcal{I}), x_1) = 3$. For the vertex x_2, there are walks of any finite length, but we visit at most three distinct vertices, namely, x_2, x_4, x_7, and $\mathsf{mvf}(\mathcal{G}(\mathcal{I}), x_2) = 3$.

For computing the MMSC up to a sufficient role depth based on MVF we use the following notion of simulation.

Definition 4. Let $\mathcal{G}_1 = (V_1, E_1, L_1)$, $\mathcal{G}_2 = (V_2, E_2, L_2)$ be $\mathcal{EL}^\perp$ description graphs and $(v_1, v_2) \in V_1 \times V_2$. A relation $Z \subseteq V_1 \times V_2$ is a *simulation* from $(\mathcal{G}_1, v_1)$ to $(\mathcal{G}_2, v_2)$, if (1) $(v_1, v_2) \in Z$, (2) $(w_1, w_2) \in Z$ implies $L_1(w_1) \subseteq L_2(w_2)$, and (3) $(w_1, w_2) \in Z$ and $(w_1, r, w_1') \in E_1$ imply there is $w_2' \in V_2$ such that $(w_2, r, w_2') \in E_2$ and $(w_1', w_2') \in Z$.

Simulations can be used to decide whether an individual from an interpretation domain belongs to the extension of a given concept expression.

Lemma 1 ((Borchmann, Distel, and Kriegel 2016))**.** Let $\mathcal{I}$ be an interpretation, let C be an $\mathcal{EL}^\perp$ concept expression, and let $\mathcal{G}(C) = (V_C, E_C, L_C)$ be the $\mathcal{EL}^\perp$ description graph of C with root ρ_C. For every $x \in \Delta^{\mathcal{I}}$, there is a simulation from $(\mathcal{G}(C), \rho_C)$ to $(\mathcal{G}(\mathcal{I}), x)$ iff $x \in C^{\mathcal{I}}$.

Lemma 1 together with other previous results is used below to prove Lemma 2, which is crucial for defining the adaptable role depth. It shows the upper bound on the required role depth of the MMSC.

Lemma 2. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation and take an arbitrary $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$, $x' \in \Delta^{\mathcal{I}}$, and $k \in \mathbb{N}$. Let

$$d = \mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)\right) \cdot \mathsf{mvf}(\mathcal{G}(\mathcal{I}), x').$$

If $x' \in \mathbf{C}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_d^{x_i}\right)^{\mathcal{I}}$ then $x' \in \mathbf{C}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_k^{x_i}\right)^{\mathcal{I}}$.

Proof Sketch. We show in the long version (Guimarães et al. 2021) the following claim.

Claim 1. For all description graphs $\mathcal{G} = (V, E, L)$ and $\mathcal{G}' = (V', E', L')$, all vertices $v \in V$ and $v' \in V'$, and

$$d = \mathsf{mvf}(\mathcal{G}, v) \cdot \mathsf{mvf}(\mathcal{G}', v')$$

if there is a simulation $Z_d : (\mathcal{G}_d^v, v) \mapsto (\mathcal{G}', v')$, then there is a simulation $Z_k : (\mathcal{G}_k^v, v) \mapsto (\mathcal{G}'v')$ for all $k \in \mathbb{N}$.

If $k \leqslant d$, one can restrict Z_d to the vertices of $\mathcal{G}_k^v$, which would be a subgraph of $\mathcal{G}_d^v$. Otherwise, the intuition behind this claim is that the pairs in Z_d define a walk in $\mathcal{G}'$ for each walk in $\mathcal{G}$ that has length at most $d-1$. And if a walk in $\mathcal{G}$ has length at least $d-1$, then there is a vertex w that this walk visits twice while the image of this walk in $\mathcal{G}'$ also repeats a vertex at the same time. This paired repetition can be used to find a matching vertex in V' for each vertex of $\mathcal{G}_k^v$ by recursively shortening the walk that this vertex corresponds to if it has length d or larger.

Lemma 1 and $x' \in \mathbf{C}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_d^{x_i}\right)^{\mathcal{I}}$ imply that there is a simulation Z_d from $\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_d^{x_i}, (x_1, \ldots, x_n)\right)$ to $(\mathcal{G}(\mathcal{I}), x')$. Then, by Claim 1 there is a simulation Z_k : $\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_k^{x_i}, (x_1, \ldots, x_n)\right) \mapsto (\mathcal{G}(\mathcal{I}), x')$ (we just need to take $\mathcal{G} = \prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$, $\mathcal{G}' = \mathcal{G}(\mathcal{I})$, $v = (x_1, \ldots, x_n)$ and $v' = x'$). Therefore, Lemma 1 implies that $x' \in \mathbf{C}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_k^{x_i}\right)^{\mathcal{I}}$. □

Lemma 2 shows that even for vertices that are parts of cycles, there is a certain depth of unravellings, which we call a fixpoint, that is guaranteed to be an upper bound.

Proposition 2 gives an intuition about how large the MVF of a vertex in a product graph can be when compared to the MVF of the corresponding vertices in the product's factors.

Proposition 2. Let $\{\mathcal{G}_i \mid 1 \leqslant i \leqslant n\}$ be n description graphs such that $\mathcal{G}_i = (V_i, E_i, L_i)$. Also let $v_i \in V_i$. Then:

$$\mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}_i, (v_1, \ldots, v_n)\right) \leqslant \prod_{i=1}^{n} \mathsf{mvf}(\mathcal{G}_i, v_i).$$

Proof. Let $\mathbf{w}$ be an arbitrary walk in $\prod_{i=1}^{n} \mathcal{G}_i, (v_i)_{1 \leqslant i \leqslant n}$ that starts in $(v_1, \ldots, v_n)$ and let $(w_1, \ldots, w_n)$ be a vertex in this walk. It follows from the definition of product that each w_i belongs to a walk in $\mathcal{G}_i$ that begins in v_i. Therefore, there are only $\mathsf{mvf}(\mathcal{G}_i, v_i)$ options for each w_i. Hence, there are at most $\prod_{i=1}^{n} \mathsf{mvf}(\mathcal{G}_i, v_i)$ possible options for $(w_1, \ldots, w_n)$. In other words, $\mathsf{v}_{\mathsf{num}}(\mathbf{w}) \leqslant \prod_{i=1}^{n} \mathsf{mvf}(\mathcal{G}_i, v_i)$. Since $\mathbf{w}$ is arbitrary, we can conclude that $\mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}_i, (v_1, \ldots, v_n)\right) \leqslant \prod_{i=1}^{n} \mathsf{mvf}(\mathcal{G}_i, v_i)$. □

Although the MVF of a product can be exponential in $|\Delta^{\mathcal{I}}|$, there are many cases in which it is linear in $|\Delta^{\mathcal{I}}|$. Example 4 illustrates one such case.

Example 4. Consider the interpretation of Figure 4. The elements x_1, x_3, x_4, x_5 and x_6 never reach cycles, therefore, each of them can only have walks up to a finite length. Take $X = \{x_1, x_2\}$. Since every walk in $\mathcal{G}(\mathcal{I})$ starting from x_1 has length at most 2, the longest walk possible in $\prod_{i \in \{1,2\}} \mathcal{G}(\mathcal{I})$ starting at node (x_1, x_2) is: (x_1, x_2), partof, (x_5, x_7), $\mathsf{capital}$, (x_6, x_2). Thus $\mathsf{mvf}\left(\prod_{i \in \{1,2\}} \mathcal{G}(\mathcal{I}), (x_1, x_2)\right) = 2$. Take $X = \{x_1, x_7\}$, then $\mathsf{mvf}\left(\prod_{i \in \{1,7\}} \mathcal{G}(\mathcal{I}), (x_1, x_7)\right) = 1$, since x_1 and x_7 do not share labels in their outgoing edges.

The observations about the MVF in Example 4 are generalised in Lemma 3 which shows a sufficient condition for polynomial (linear) role depth.

Lemma 3. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation and $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$. If for some $1 \leqslant i \leqslant n$ it holds that every walk in $\mathcal{G}(\mathcal{I})$ starting at x_i has length at most m for some $m \in \mathbb{N}$, then $\mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)\right) \leqslant \mathsf{mvf}\left(\mathcal{G}(\mathcal{I}), x_i\right)$.

Proof Sketch. As it happens in Example 4, it can be proven that whenever there is a vertex x_i for which every walk starting at it has length at most m, then m also bounds the lengths of the walks starting at $(x_1, \ldots x_n)$ in $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$. □

Combining the bounds for the fixpoint and MVF given by Lemmas 2 and 3, we can define a function that returns an upper approximation of the fixpoint, for any subset of the domain of an interpretation, as follows.

Definition 5. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation and $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$. Also let

$$X_{lim} = \{x \in X \mid \exists m \in \mathbb{N} : \text{every walk starting at } x \text{ in } \mathcal{G}(\mathcal{I}) \text{ has length} \leqslant m\}.$$

The function $d_{\mathcal{I}} : \mathcal{P}(\Delta^{\mathcal{I}}) \mapsto \mathbb{N}$ is defined as follows:

$$d_{\mathcal{I}}(X) = \begin{cases} d - 1 & \text{if } X_{lim} \neq \varnothing \\ d \cdot \mathsf{mmvf}(\mathcal{G}(\mathcal{I})) & \text{otherwise,} \end{cases}$$

where $d = \mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)\right)$.

Next, we prove that function $d_{\mathcal{I}}$ is indeed an upper bound for the fixpoint of an MMSC. The idea sustaining Lemma 4 is that if $x \in X \subseteq \Delta^{\mathcal{I}}$ and every walk in $\mathcal{G}(\mathcal{I})$ starting at x has length at most m, then m can be used as a fixpoint depth for the MMSC of X in $\mathcal{I}$. Lemma 2 covers the cases where vertices are the starting point of walks of any length.

Lemma 4. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation and $X \subseteq \Delta^{\mathcal{I}}$. Then, for any $k \in \mathbb{N}$, it holds that:

$$\mathsf{mmsc}(X, \mathcal{I}, d_{\mathcal{I}}(X))^{\mathcal{I}} \subseteq \mathsf{mmsc}(X, \mathcal{I}, k)^{\mathcal{I}}.$$

Proof Sketch. Let $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$. If $k \leqslant d_{\mathcal{I}}(X)$, the lemma holds trivially. For $k > d_{\mathcal{I}}(X)$ we divide the proof in two cases. First, if there is a $x_i \in X$ such that every walk in $\mathcal{G}(\mathcal{I})$ starting at x_i has length at most m for some $m \in \mathbb{N}$, then as stated in Lemma 3, every walk in $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$ starting at $(x_1, \ldots, x_n)$ has length at most $\mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)\right) - 1$.

In other words, even when $k > d_{\mathcal{I}}(X)$, we have: $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_k^x = \prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_{d_{\mathcal{I}}(X)}^x$, and therefore, we can apply Lemma 1 to conclude that: $\mathsf{mmsc}(X, \mathcal{I}, d_{\mathcal{I}}(X))^{\mathcal{I}} \subseteq \mathsf{mmsc}(X, \mathcal{I}, k)^{\mathcal{I}}$. Otherwise, if $X_{lim} \neq \varnothing$, the lemma is a direct consequence of Definition 5 and Lemma 2. □

In the remaining of this paper, we write $\mathsf{mmsc}(X, \mathcal{I})$ as a shorthand for $\mathsf{mmsc}(X, \mathcal{I}, d_{\mathcal{I}}(X))$. An important consequence of Lemma 4 and the definition of MMSC is that, for any $\mathcal{EL}^{\perp}$ concept expression C and finite interpretation $\mathcal{I}$, it holds that $C^{\mathcal{I}} = \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}}$.

Lemma 5. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation. Then, for all $\mathcal{EL}^{\perp}$ concept expression C it holds that: $\mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = C^{\mathcal{I}}$.

Proof. Direct consequence of Lemma 4.4 (vi) of (Borchmann, Distel, and Kriegel 2016) and Lemma 4. □

We use this result below to define a finite set of concept expressions $M_{\mathcal{I}}$ for building a base of the CIs valid in $\mathcal{I}$.

Definition 6. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation. The set $M_{\mathcal{I}}$ is the union of $\{\perp\} \cup \mathsf{N_C}$ and

$$\{\exists r.\mathsf{mmsc}(X, \mathcal{I}) \mid r \in \mathsf{N_R} \text{ and } X \subseteq \Delta^{\mathcal{I}}, X \neq \varnothing\}$$

We also define $\Lambda_{\mathcal{I}} = \{\bigsqcap U \mid U \subseteq M_{\mathcal{I}}\}$.

Building the base mostly relies on the fact that, given a finite interpretation $\mathcal{I}$, for any $\mathcal{EL}^{\perp}$ concept expression C, there is a concept expression $D \in \Lambda_{\mathcal{I}}$ such that $C^{\mathcal{I}} = D^{\mathcal{I}}$.

Theorem 2. Let $\mathcal{I}$ be a finite interpretation and let $\Lambda_{\mathcal{I}}$ be defined as above. Then,

$$\begin{aligned} \mathcal{B}(\mathcal{I}) = &\{C \equiv \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right) \mid C \in \Lambda_{\mathcal{I}}\} \cup \\ &\{C \sqsubseteq D \mid C, D \in \Lambda_{\mathcal{I}} \text{ and } \mathcal{I} \models C \sqsubseteq D\} \end{aligned}$$

is a finite $\mathcal{EL}^{\perp}$ base for $\mathcal{I}$.

Proof Sketch. As $\Lambda_{\mathcal{I}}$ is finite, so is $\mathcal{B}(\mathcal{I})$. The CIs are clearly sound and the soundness of the equivalences is due to Lemma 5. For completeness, assume that $\mathcal{I} \models C \sqsubseteq D$. Using an adaptation of Lemma 5.8 from (Distel 2011) and Lemma 5 above, we can prove, by induction on the structure of the concept expressions C and D, that there are concept expressions $E, F \in \Lambda_{\mathcal{I}}$ such that $\mathcal{B}(\mathcal{I}) \models E \equiv \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)$, $\mathcal{B}(\mathcal{I}) \models F \equiv \mathsf{mmsc}\left(D^{\mathcal{I}}, \mathcal{I}\right)$, $\mathcal{B}(\mathcal{I}) \models C \equiv \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)$, and $\mathcal{B}(\mathcal{I}) \models D \equiv \mathsf{mmsc}\left(D^{\mathcal{I}}, \mathcal{I}\right)$. By construction, as $E \sqsubseteq F \in \mathcal{B}(\mathcal{I})$, we can prove that whenever $\mathcal{I} \models C \sqsubseteq D$, so does $\mathcal{B}(\mathcal{I})$. □

Recall the interpretation $\mathcal{I}$ in Figure 6. In order to compute a base for $\mathcal{I}$, we should compute an MMSC with role depth at least 29. An important benefit of our approach is that the role depth of the other MMSCs, which are part of the mined CIs in the base may be smaller. For instance, the role depth of $\mathsf{mmsc}(\{x_1\}, \mathcal{I})$ is 10. In the next section, we show that one can compute the MVF of a vertex in a graph in linear time in the size of the graph.

5 Computing the MVF

As discussed in Section 4, the MVF is the key to provide an upper bound for the fixpoint for each MMSC. An easy way to estimate the MVF would consist in computing the number of vertices reachable from v in the description graph $\mathcal{G}$. Let $\mathsf{reach}(\mathcal{G}, v)$ be such a function. By definition it holds that $\mathsf{mvf}(\mathcal{G}, v) \leqslant \mathsf{reach}(\mathcal{G}, v)$. Although $\mathsf{reach}(\mathcal{G}, v)$ can be computed in polynomial time, the difference between these two metrics can be quite large. For instance, consider that v is the root of a description graph $\mathcal{G}$ that is a binary tree with 2^n nodes. Then $\mathsf{mvf}(\mathcal{G}, v) = n$, while $\mathsf{reach}(\mathcal{G}, v) = 2^n$.

In this section, we present an algorithm to compute $\mathsf{mvf}(\mathcal{G}, v)$ that takes linear time in the size of $\mathcal{G}$, but first we need to recall some fundamental concepts from Graph Theory, one of them is the notion of strongly connected components (Definition 7).

Definition 7. Let $\mathcal{G} = (V, E, L)$ be a description graph. The *strongly connected components* (SCCs) of $\mathcal{G}$, in symbols $\mathsf{SCC}(\mathcal{G})$, are the partitions $V_1, \ldots, V_n$ of V such that for all $1 \leqslant i \leqslant n$: if $u, v \in V_i$ then there is a path from u to v and a path from v to u in $\mathcal{G}$. Additionally, we define a function $\mathsf{scc}(\mathcal{G}, v)$, which returns the SCC of $\mathcal{G}$ that contains v.

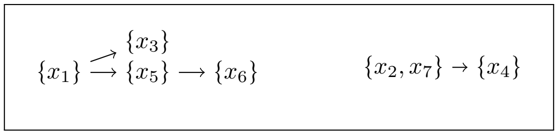

Figure 7: Condensation of the description graph in Figure 4. Every vertex is an SCC of the original graph and the edges indicate accessibility between the SCCs. Also, the condensation has no labels.

A compact way of representing a description graph $\mathcal{G}$ consists in regarding each SCC in $\mathcal{G}$ as a single vertex. This compact graph is a directed acyclic graph (DAG), also called condensation of $\mathcal{G}$ (Harary, Norman, and Cartwright 1965), and it is formalised in Definition 8.

Definition 8. Let $\mathcal{G} = (V, E, L)$ be a description graph. The *condensation of* $\mathcal{G}$ is the directed acyclic graph $\mathcal{G}^* = (V^*, E^*)$ where $V^* = \{\mathsf{scc}(\mathcal{G}, u) \mid u \in V\}$ and $E^* = \{(\mathsf{scc}(\mathcal{G}, u), \mathsf{scc}(\mathcal{G}, v)) \mid (u, r, v) \in E \text{ and } \mathsf{scc}(\mathcal{G}, u) \neq \mathsf{scc}(\mathcal{G}, v)\}$. Also, if $\mathbf{w}^*$ is path in $\mathcal{G}^*$, the *weight of* $\mathbf{w}^*$, in symbols $\mathsf{weight}(\mathcal{G}^*)$, is the sum of the sizes of the SCCs that appear as vertices of $\mathbf{w}^*$.

We use these notions to link the MVF (Definition 3) to the paths in the condensation graph in Lemma 6.

Lemma 6. Let $\mathcal{G} = (V, E, L)$ be a description graph, let $\mathcal{G}^* = (V^*, E^*)$ be the condensation of $\mathcal{G}$, and $v \in V$. Then:

$$\mathsf{mvf}(\mathcal{G}, v) = \max\{\mathsf{weight}(\mathbf{w}^*) \mid \mathbf{w}^* \text{ is a path in } \mathcal{G}^* \text{ starting at } \mathsf{scc}(\mathcal{G}, v)\}.$$

Proof Sketch. First we prove that every path $\mathbf{w}^* = V_1, \ldots, V_m$ in $\mathcal{G}^*$ starting at $\mathsf{scc}(\mathcal{G}, v)$ induces a walk $\mathbf{w}$ in $\mathcal{G}$ starting at v with $\mathsf{v_{num}}(\mathbf{w}) = \mathsf{weight}(\mathbf{w}^*)$. Then, we show that if $\mathbf{w}^*$ has maximal weight, then no walk in $\mathcal{G}$ starting at v can visit more than $\mathsf{weight}(\mathbf{w}^*)$ vertices. □

By Lemma 6, we only need to compute the maximum weight of a path in $\mathcal{G}^*$ that starts at $\mathsf{scc}(\mathcal{G}^*, v)$ to obtain the MVF of a vertex v in a description graph $\mathcal{G}$. Algorithm 1 relies on this result and proceeds as follows: first, it computes the SCCs of the description graph and the condensation graph. Then, the algorithm transverses the condensation graph, using an adaptation of depth-first search to determine the maximum path weight for the initial SCC.

Algorithm 1 assumes that the SCCs and condensation are computed correctly. Besides keeping the computed values, the array wgt prevents recursive calls on SCCs that have already been processed. According to Lemma 6, to prove that Algorithm 1 is correct we just need to prove that the function maxWeight in fact returns the maximum weight of a path in the condensation given a starting vertex (which corresponds to an SCC in the original graph).

Lemma 7. Given $\mathcal{G} = (V, E, L)$ and $v \in V$ as input, Algorithm 1 returns the maximum weight of a path in the condensation of $\mathcal{G}$ starting at $\mathsf{scc}(\mathcal{G}, v)$.

Proof Sketch. Let $\mathcal{G}^* = (V^*, E^*)$ be the condensation of $\mathcal{G}$. If $\mathsf{scc}(\mathcal{G}, v)$ has no successor in $\mathcal{G}^*$, then the output of maxWeight is correct. If $\mathsf{scc}(\mathcal{G}, v)$ has successors, then the maximum weight of a path staring at $\mathsf{scc}(\mathcal{G}, v)$ in $\mathcal{G}^*$ is given by $|\mathsf{scc}(\mathcal{G}, v)|$ plus the maximum value computed among its successors. This equation holds because $\mathcal{G}^*$ is a DAG. □

Algorithm 1: Computing MVF via Lemma 6

```
Input: A description graph $\mathcal{G} = (V, E, L)$ and a
       vertex $v \in V$
Output: The MVF of $v$ in $\mathcal{G}$, i.e., $\mathsf{mvf}(G, v)$
$V^* \leftarrow \mathsf{SCC}(\mathcal{G})$
$E^* \leftarrow \mathsf{condense}(\mathcal{G}, V^*)$
$\mathcal{G}^* \leftarrow (V^*, E^*)$
for $V' \in V^*$ do
    $wgt[V'] \leftarrow$ null
return $\mathsf{maxWeight}(\mathcal{G}^*, \mathsf{scc}(\mathcal{G}, v), wgt)$
// Auxiliary function
Function $\mathsf{maxWeight}(\mathcal{G}^*, V', wgt)$:
    $current \leftarrow 0$
    for $W' \in \{U' \in V^* \mid (V', U') \in E^*\}$ do
        if $wgt[W'] =$ null then
            $current \leftarrow$
              $\max(current, \mathsf{maxWeight}(\mathcal{G}^*, W', wgt))$
        else
            $current \leftarrow wgt[W']$
        $wgt[V'] \leftarrow current + |V'|$
    return $wgt[V']$
```

Lemmas 6 and 7 imply that Algorithm 1 computes the MVF of v in $\mathcal{G}$ correctly. Moreover, the computation of SCCs can be done in time $O(|V| + |E|)$ (Tarjan 1972), the condensation in time $O(|E|)$ (Martello and Toth 1982) and the depth-first transversal via maxWeight in time $O(|V| + |E|)$. Hence, it is possible to compute the MVF of a vertex in a graph in linear time in the size of the description graph even if it consists solely of cycles. Yet, given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ the graph given as input to Algorithm 1 might be a product graph with an exponential number of vertices in $|\Delta^{\mathcal{I}}|$. Also, Algorithm 1 can be modified to compute the MVF for all vertices by starting the function maxWeight from an unvisited SCC until all vertices are visited in polynomial time in the size of the graph.

6 Conclusion

In this work, we introduce a way of computing $\mathcal{EL}^{\bot}$ bases from finite interpretations that adapts the role depth of concepts according to the the structure of interpretations. Our definition relies on a notion that relates vertices in a graph to sets of vertices, called MVF. We have also shown that the MVF computation can be performed in polynomial time in the size of the underlying graph structure. Our $\mathcal{EL}^{\bot}$ base, however, is not minimal. As future work, we plan to build on previous results combining FCA and DLs to define a base with minimal cardinality. We will also investigate the problem of mining CIs in the presence of noise in the dataset. We plan to use the support and confidence measures from association rule mining to deal with noisy data and implement our approach using knowledge graphs as datasets.

Acknowledgements

Parts of this work have been done in the context of CEDAS (Center for Data Science, University of Bergen, Norway). This work was supported by the Free University of Bozen-Bolzano, Italy, under the project PACO.

References

Baader, F. 1995. Computing a Minimal Representation of the Subsumption Lattice of All Conjunctions of Concepts Defined in a Terminology. In *Proc. Intl. KRUSE Symposium*, 168–178.

Baader, F. 2003. Terminological Cycles in a Description Logic with Existential Restrictions. In *IJCAI*, 325–330. Morgan Kaufmann.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ Envelope. In Kaelbling, L. P.; and Saffiotti, A., eds., *IJCAI*, 364–369. Professional Book Center.

Baader, F.; and Distel, F. 2008. A Finite Basis for the Set of $\mathcal{EL}$-Implications Holding in a Finite Model. In Medina, R.; and Obiedkov, S., eds., *ICFCA 2008*, 46–61. Springer-Verlag.

Baader, F.; Ganter, B.; Sertkaya, B.; and Sattler, U. 2007. Completing Description Logic Knowledge Bases using Formal Concept Analysis. In Veloso, M. M., ed., *IJCAI*, 230–235. AAAI Press.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Baader, F.; and Molitor, R. 2000. Building and Structuring Description Logic Knowledge Bases Using Least Common Subsumers and Concept Analysis. In *ICCS*, 292–305. Springer.

Borchmann, D. 2014. *Learning Terminological Knowledge with High Confidence from Erroneous Data*. Ph.D. thesis, Dresden University of Technology.

Borchmann, D.; and Distel, F. 2011. Mining of EL-GCIs. In Spiliopoulou, M.; Wang, H.; Cook, D. J.; Pei, J.; Wang, W.; Zaïane, O. R.; and Wu, X., eds., *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, 1083–1090. IEEE Computer Society.

Borchmann, D.; Distel, F.; and Kriegel, F. 2016. Axiomatisation of general concept inclusions from finite interpretations. *Journal of Applied Non-Classical Logics* 26(1): 1–46.

Distel, F. 2011. *Learning description logic knowledge bases from data using methods from formal concept analysis*. Ph.D. thesis, Dresden University of Technology.

Ganter, B.; and Wille, R. 1999. *Formal Concept Analysis: Mathematical Foundations*. Berlin/Heidelberg: Springer.

Guimarães, R.; Ozaki, A.; Persia, C.; and Sertkaya, B. 2021. Mining $\mathcal{EL}^{\perp}$ Bases with Adaptable Role Depth. *arXiv e-prints* arXiv:2102.10689.

Harary, F.; Norman, R. Z.; and Cartwright, D. 1965. *Structural models: an introduction to the theory of directed graphs*. New York: John Wiley & Sons. ISBN 9780471351306.

Kriegel, F. 2019a. *Constructing and Extending Description Logic Ontologies using Methods of Formal Concept Analysis*. Ph.D. thesis, Technische Universität Dresden, Dresden, Germany.

Kriegel, F. 2019b. Learning Description Logic Axioms from Discrete Probability Distributions over Description Graphs. In Calimeri, F.; Leone, N.; and Manna, M., eds., *JELIA 2019*, 399–417. Springer.

Kuznetsov, S. O. 2004. On the Intractability of Computing the Duquenne-Guigues Base. *J. UCS* 10(8): 927–933.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; and Bizer, C. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6(2): 167–195.

Martello, S.; and Toth, P. 1982. Finding a minimum equivalent graph of a digraph. *Networks* 12(2): 89–100.

Monnin, P.; Lezoche, M.; Napoli, A.; and Coulet, A. 2017. Using Formal Concept Analysis for Checking the Structure of an Ontology in LOD: The Example of DBpedia. In *ISMIS*, 674–683. Springer.

Ozaki, A. 2020. Learning Description Logic Ontologies: Five Approaches. Where Do They Stand? *KI - Künstliche Intelligenz* .

Rudolph, S. 2004. Exploring Relational Structures Via $\mathcal{FLE}$. In Wolff, K. E.; Pfeiffer, H. D.; and Delugach, H. S., eds., *ICCS*, 196–212. Springer-Verlag.

Rudolph, S. 2006. *Relational exploration: Combining Description Logics and Formal Concept Analysis for knowledge specification*. Ph.D. thesis, Fakultät Mathematik und Naturwissenschaften, TU Dresden, Germany.

Sertkaya, B. 2010. A Survey on how Description Logic Ontologies Benefit from Formal Concept Analysis. In Kryszkiewicz, M.; and Obiedkov, S., eds., *CLA*, volume 672 of *CEUR Workshop Proceedings*, 2–21.

Spackman, K.; Campbell, K.; and Cote, R. 1997. SNOMED RT: A reference terminology for health care. *J. American Medical Informatics Association* 640–644. Fall Symposium Supplement.

Tarjan, R. 1972. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing* 1(2): 146–160.