Journal of Artificial Intelligence Research 76 (2023) 883-924 Submitted 03/2022; published 04/2023

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

Ricardo Guimarães Ricardo.Guimaraes@uib.no
Ana Ozaki Ana.Ozaki@uib.no
Cosimo Persia Cosimo.Persia@uib.no
Department of Informatics, University of Bergen

Barış Sertkaya sertkaya@fb2.fra-uas.de
Frankfurt University of Applied Sciences

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 has been 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. We also consider the case of *confident* $\mathcal{EL}^\perp$ bases, which requires that some proportion of the domain of the interpretation satisfies the base, instead of the whole domain. This case is useful to cope with noisy data.

1. Introduction

In artificial intelligence (AI), logic plays a key rôle in representing the knowledge of an application domain in a structured and formally well-understood way. It allows us to model the relevant notions of the application domain as classes of individuals sharing some commonalities, describe individuals and their relationships to each other and to the classes. This enables AI applications to reason about complex relational data, deduce new facts and extract hidden relationships.

Among the logic-based knowledge representation formalisms, Description Logics (DLs) (Baader, Horrocks, Lutz, & Sattler, 2017) is a well-established family of logics that is used for representing conceptual knowledge and reasoning about such knowledge. DLs have proven successful in various application domains such as natural language processing, configuration, databases, and bio-medical ontologies, but their most notable success is due to the fact that DLs provide the logical underpinning of OWL, the standard ontology language for the semantic web (Horrocks, Patel-Schneider, & van Harmelen, 2003). As a consequence of this standardization, tools that support knowledge engineers in building knowledge bases (KBs) written in OWL and maintaining their quality gained more importance. Once the KB is built, there are several mature tools that use DL reasoning for inferring new consequences from the KB and for detecting inconsistencies in modelling. There are also tools that support the knowledge engineer in pinpointing the reasons for inconsistencies and help her to resolve them or to remove unwanted consequences (Horridge, 2011; Peñaloza, 2009). These tools adress the maintenance aspect of an existing knowledge base and not the aspect of building a knowledge base from the scratch.

In real world applications, the amount of knowledge to be modelled can be large and the knowledge engineer is not an expert in modelling with a logical language. This makes the manual construction of knowledge bases with concept inclusions (CIs) formulated in a DL a rather complex and time-consuming task. Given a data set one might be interested in knowing which CIs hold in it, in particular, if there a finite representation for them and how concise this representation can be. In the present work we study these questions in light of Formal Concept Analysis (FCA) (Ganter & Wille, 1999) and DLs (Baader et al., 2017). The goal is to extract CIs from a data set, which can be a collection of facts in a database, a set of statements, or a knowledge graph.

Example 1. *Consider the DBpedia knowledge graph (Lehmann, Isele, Jakob, Jentzsch, Kontokostas, Mendes, Hellmann, Morsey, van Kleef, Auer, & Bizer, 2015), where one can represent a city '*a*', which is the capital of a region '*b*', 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 small data set, one can mine a CI expressing that a capital is a city that is part of a region.*

FCA is an application of lattice theory that provides techniques for discovering clusters in a data set, building a hierarchy of these clusters and identifying dependencies in the data set. In FCA, data is represented as a *formal context*, which is a table showing which objects have which attributes. For analysing a given formal context, one can extract the dependencies between sets of attributes, also called *implications*. The set of all implications that hold in a formal context can be large. Therefore, a compact representation called a *base*, that entails every valid implication of the data set, has been considered in the literature (see Section 2). In the case of plain FCA, implications correspond to propositional formulae and well-known algorithms exist for extracting such a base from the data set. However, if we want to extract a base of CIs, which are formulated in a DL, the situation gets more complicated. For some DLs, it may even happen that no such finite base exists. The reason is simply existence of cyclic relationships expressible in DLs. With only one cyclic relationship in the data set, we can potentially express infinitely many different CIs.

An approach to deal with cyclic relationships has been proposed by Baader and Distel (2008). The authors propose to use the DL $\mathcal{EL}^{\bot}_{gfp}$ for capturing the semantics of cyclic relationships. $\mathcal{EL}^{\bot}_{gfp}$ is the DL enriching $\mathcal{EL}^{\bot}$ with greatest fix-point semantics. The semantics offered by $\mathcal{EL}^{\bot}_{gfp}$ elegantly solves the difficulty of mining CIs from cyclic relationships in the data. However, this semantics comes with two drawbacks. First, $\mathcal{EL}^{\bot}_{gfp}$ concepts may be more difficult to understand than simply $\mathcal{EL}^{\bot}$ concepts since $\mathcal{EL}^{\bot}$ is already widely used and accepted as an ontology language. Second, there is no efficient implementation of a reasoner for $\mathcal{EL}^{\bot}_{gfp}$, even though the reasoning complexity is tractable (Baader, 2003), like for $\mathcal{EL}^{\bot}$. In the same work the authors also present a way of transforming an $\mathcal{EL}^{\bot}_{gfp}$ base into an $\mathcal{EL}$ base. However, it is far from being trivial to avoid the step of creating an $\mathcal{EL}^{\bot}_{gfp}$ base in their approach. In a later work (Borchmann, Distel, & Kriegel, 2016), the authors propose an approach for mining finite $\mathcal{EL}^{\bot}$ bases with a predefined and fixed role depth of concept expressions. As a consequence, the base is sound and complete only w.r.t. CIs containing concepts with a bounded role depth. Their approach elegantly avoids the complicated step of creating an $\mathcal{EL}^{\bot}_{gfp}$ base but has the drawback of being incomplete for $\mathcal{EL}^{\bot}$ CIs with concept descriptions of arbitrary depth.

The present work aims to bring together the best sides of the two approaches: we propose a way of directly computing a finite $\mathcal{EL}^{\bot}$ base without the intermediate step of computing an $\mathcal{EL}^{\bot}_{gfp}$ base. Our approach is able to extract a base of cyclic relations of arbitrary role depth, not only the ones up to a certain role depth. In particular, we present a new approach for computing the role depth of concepts which is *"adaptable"* based on the objects considered during the computation of the base. The number of CIs in our approach is in the worst case double-exponential in the size of the finite interpretation given as input. We point out that the number of CIs in the work by Baader and Distel (2008) is minimal and it is double-exponential in the worst case, just as in our approach. However, in general, the number of CIs in our work may not be minimal. This work extends our conference paper (Guimarães, Ozaki, Persia, & Sertkaya, 2021) with full proofs of the results presented there and with a section on compact representation of the product graph. In addition to these we extend our results to the case of mining a base from noisy data as suggested in (Borchmann, 2014). To this purpose, we show that our approach of adaptable role depth easily extends to CIs with a certain *confidence* threshold. This is especially useful in applications where CIs are satisfied by only a subset of the data due to noise, which is a common problem in datasets for real world applications.

The paper is organised as follows: in the next section, we present a short overview of previous work on extracting CIs in the context of DLs. In Section 3, we introduce the basic definitions and notions of DLs and description graphs. In Section 4, we present the problem of mining $\mathcal{EL}^{\bot}$ CIs and establish lower bounds for this problem. In Section 5, 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 6, we show that the MVF of a vertex in a graph can be computed in linear time in the size of the graph. In Section 7 we extend our results for mining confident bases with adaptable role depth. Full proofs of our results can be found in Appendix A.

2. Related Work

The notion of an implication for expressing dependencies between properties of objects has been considered in the literature in several different contexts. An implication $X \rightarrow Y$ has the meaning that objects that have all attributes in X also have the attributes in Y. In data mining such an implication, called a *strong association rule*, expresses that if the items in the set X occur together in a transaction, then the items in Y are also likely to occur in this transaction. An algorithmic approach for discovering association rules in large data sets was first formulated in (Agrawal, Imielinski, & Swami, 1993). The *Apriori* algorithm introduced there mines frequent item sets based on the parameter *confidence*. It is the ratio of the number of objects possessing all attributes in X and in Y to the number of objects that possess the attributes in X. An association rule $X \rightarrow Y$ with a confidence value of 1 is called a strong association rule, meaning that every object having attributes in X also has attributes in Y.

In FCA, data is represented in a formal context, where rows represent objects and columns represent the attributes. An implication $X \rightarrow Y$ in a formal context has the same meaning as a strong association rule in data mining. Both in data mining and in FCA, the number of implications that hold in the dataset can be large. Hence one is interested in finding a small base that generates the whole set of implications holding in the data, which is called the *implicational theory* of the dataset. There may exist different bases for the same implicational theory. A base is called a *minimum base* if no other base with lower cardinality exists. In (Guigues & Duquenne, 1986), the authors described a minimum base called the *Duquenne-Guigues Base*, or the *stem base* of a formal context. In (Ganter, 1984, 2010) Ganter introduced an algorithm for computing the stem base. An alternative approach was presented in (Obiedkov & Duquenne, 2007). It is well-known that even such a minimum base can have a large size, namely exponential in the size of the given formal context (Kuznetsov, 2004a) (see also (Kautz, Kearns, & Selman, 1995) for the same result formulated in a different setting).

Given this fact, it is clearly not possible to efficiently compute the implicational base of a given dataset. The most well-known algorithm (Ganter, 1984), besides generating the implications of the base, generates the so-called concept intents as well, which can be exponentially more than the implications themselves. That is, the runtime of the algorithm is not bounded by a polynomial in the size of the output, i.e., it is not output-polynomial (Johnson, Yannakakis, & Papadimitriou, 1988). Similarly, the time complexity of the algorithm introduced by Obiedkov and Duquenne (2007) also depends on the number of concept intents. In the light of our current knowledge, it is not clear whether the stem base can efficiently be computed. Distel and Sertkaya (2009a, 2011) have shown that this problem is at least as hard as enumerating the minimal transversals of a given hypergraph, which is a long-standing open problem (Eiter & Gottlob, 1995). In a later work, it was shown that deciding whether an implication belongs to the base of a formal context was also shown to be harder than recognizing the minimal transversals of a hypergraph (Sertkaya, 2009b), which was later shown to be intractable (Babin & Kuznetsov, 2013).

Nevertheless, using methods from FCA, especially the idea of computing a small base of the axioms that hold in a DL interpretation has attracted attention in the DL community. Baader (1995) has used FCA for an efficient computation of an extended subsumption hierarchy of a set of DL concepts. More precisely, he used attribute exploration for computing the subsumption hierarchy of all conjunctions of a set of DL concepts. Baader and Molitor (2000) have used FCA for supporting bottom-up construction of DL knowledge bases, where the knowledge engineer does not directly define the concepts of her application domain, but she gives typical examples of a concept, and the system comes up with a concept description for these examples. Rudolph (2004, 2006) has combined DLs and FCA for acquiring complete relational knowledge about an application domain. In his approach, which he calls relational exploration, he uses DLs for defining FCA attributes, and FCA for refining DL knowledge bases. More precisely, DLs make use of the interactive knowledge acquisition method of FCA, and FCA benefits from DLs in terms of expressing relational knowledge. Baader, Ganter, Sertkaya, and Sattler (2007) used the FCA-based knowledge acquisition technique attribute exploration for detecting missing CIs and assertions in a DL knowledge base by asking questions to a domain expert. The questions asked by the algorithm in this approach are basically the base computed from the assertions.

As already mentioned, Baader and Distel (2008, 2009) proposed an FCA-based approach for mining a base of all $\mathcal{EL}^{\perp}_{gfp}$ CIs holding in a given model. Their approach is based on the notion of a *model-based most-specific concept*, which is the most-specific concept description expressible in $\mathcal{EL}^{\perp}_{gfp}$ that contains a given set of objects in its extension. The authors show that in this DL such a concept description exits for arbitrary finite models and present a way of computing it. The fixpoint semantics of this DL can express cycles and guarantees the existence of a finite base, however it has the drawback that the base also contains cyclic concept descriptions, which are too hard to comprehend by domain experts. To overcome this problem, Distel (2011) proposes to unravel the cyclic concept descriptions until a certain role depth still guaranteeing the completeness of the resulting base. The problem with this approach, however, is the large role depth, as we analyze in the upcoming sections. In a later work, Borchmann and Distel (2011) implement this approach and present the results of evaluating it on the DBpedia knowledge graph (Auer, Bizer, Kobilarov, Lehmann, Cyganiak, & Ives, 2007; Lehmann et al., 2015). An approach for analysing data in an RDF-triple store using FCA to identify clusters has been proposed by Dau and Sertkaya (2011). In his dissertation, Borchmann (2014) considers noisy data and shows how a base of confident ${\mathcal{EL}^{\perp}}_{gfp}$ concept inclusions can be extracted from a DL interpretation by extending Distel's work. Borchmann et al. (2016) address the problems with the fixpoint semantics of $\mathcal{EL}^{\perp}_{gfp}$. They propose an approach for mining $\mathcal{EL}^{\perp}$ bases with a predefined and fixed role depth of concept expressions.

Monnin et al. compare, using FCA techniques, data present in DBpedia with the constraints of a given ontology to check if the data is compliant with it (Monnin, Lezoche, Napoli, & Coulet, 2017). Kriegel (2019a, 2020a) among other contributions employs FCA notions to build ontologies in DLs more expressive than $\mathcal{EL}^{\perp}$, building upon the framework already established for $\mathcal{EL}^{\perp}$ (Borchmann et al., 2016) and $\mathcal{EL}^{\perp}_{gfp}$ (Distel, 2011). He investigates the problem of learning axioms in a probabilistic variant of $\mathcal{EL}^{\perp}$ (Kriegel, 2017; Kriegel, 2019b) and also from streams of interpretations (Kriegel, 2016, 2020b). Learnability of CIs in lightweight DLs from a given set of interpretations has been studied by Klarman and Britz (2015) and exact learnability of lightweight DL KBs in Angluin's framework via queries has been studied by Konev, Lutz, Ozaki, and Wolter (2017). In the context of learning DLs, we also highlight works based on Inductive Logic Programming (Fanizzi, d'Amato, & Esposito, 2008; Funk, Jung, Lutz, Pulcini, & Wolter, 2019; Iannone, Palmisano, & Fanizzi, 2007; Lehmann, 2009, 2010; Lehmann & Hitzler, 2010; Lisi, 2011). For a survey on applying FCA methods in the DL community see the survey by Sertkaya (2010) and for a more recent survey on learning DL knowledge bases from data, see the work by Ozaki (2020).

3. Preliminaries

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

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

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

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

$\mathcal{EL}^\perp$ (Baader, Brandt, & 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, & Côté, 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 $\mathcal{EL}^\perp$. $\mathcal{EL}^\perp$ *concept expressions* are built according to the grammar rule $C, D ::= A \mid \top \mid \perp \mid C \sqcap D \mid \exists r.C$ 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 $\mathcal{EL}^\perp$ *TBox* is a finite set of *concept inclusions* (CIs) $C \sqsubseteq D$, where C, D are $\mathcal{EL}^\perp$ concept expressions. We may omit '$\mathcal{EL}^\perp$' 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 shorthand 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 $\mathcal{EL}^\perp$ 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 $\mathcal{EL}^\perp$ concept expressions as usual:

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

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 $X^\mathcal{I} \neq \emptyset$. 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. $L_\mathcal{I}(x) = \{A \in \mathsf{N_C} \mid x \in A^\mathcal{I}\}$.

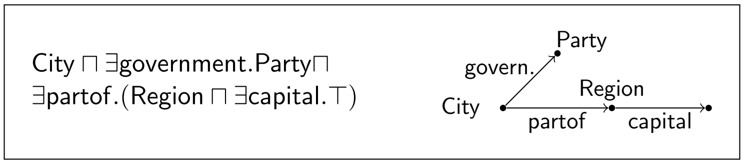

Figure 2: A concept expression and its description tree.

The *description tree* of an $\mathcal{EL}^\perp$ 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 $L_C(\rho_C) = \emptyset$ 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 identifying the roots ρ_{D_1}, ρ_{D_2} as one root ρ_C with $L_C(\rho_C) = L_{D_1}(\rho_{D_1}) \cup L_{D_2}(\rho_{D_2})$;
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 $L(v) = \{P_i \mid 1 \leq i \leq k\}$, $(v, r_j, w_j) \in E$ (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 (Figure 2).

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 $\mathcal{EL}^\perp$ 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 $\mathcal{G}_d^x = (V_d, E_d, L_d)$ 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.

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. $V = \bigtimes_{i=1}^{n} V_i$;

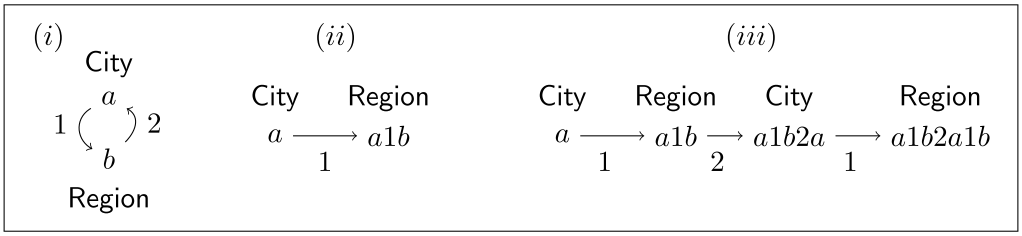

Figure 3: Unravelling of the description graph of the interpretation $\mathcal{I}$ in (i). For readability, partof has been replaced with symbol 1 and capital with symbol 2. (ii) depicts $\mathcal{G}(\mathcal{I})^a_1$ and (iii) depicts $\mathcal{G}(\mathcal{I})^a_3$.

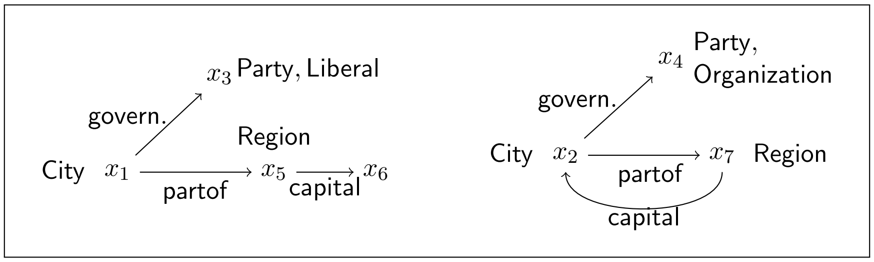

Figure 4: Description graph of the interpretation $\mathcal{I} = \{\{x_1, \cdots, x_7\}, \cdot^{\mathcal{I}}\}$ where $\{x_1, x_2\} = \mathsf{City}^{\mathcal{I}}$, $\{x_3, x_4\} = \mathsf{Party}^{\mathcal{I}}$, $\{(x_1, x_5), (x_2, x_7)\} = \mathsf{partof}^{\mathcal{I}}$, etc.

2. $E = \{((v_1, \ldots, v_n), r, (w_1, \ldots, w_n)) \mid r \in \mathsf{N_R}, (v_i, r, w_i) \in E_i, \text{ for all } 1 \leq i \leq n\}$;

3. $L(v_1, \ldots, v_n) = \bigcap_{i=1}^n L_i(v_i)$.

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 $\mathcal{G}_1, \ldots, \mathcal{G}_n$.

4. 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 $\mathcal{I} \models C \sqsubseteq D$ we also have that $\mathcal{I} \models \exists r.C \sqsubseteq \exists r.D$ 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. *$\mathcal{EL}^\perp_{rhs}$ and $\mathcal{EL}^\perp_{lhs}$ do not have the FBP.*

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

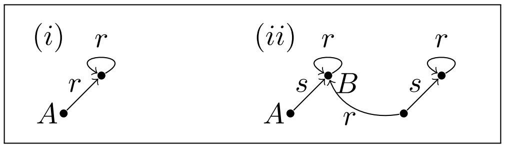

Figure 5: Lack of the FBP for $\mathcal{EL}^\perp_{rhs}$ (i) and $\mathcal{EL}^\perp_{lhs}$ (ii).

The main difficulty in creating an $\mathcal{EL}^\perp$ 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, $\mathcal{EL}^\perp$ concept expressions cannot express cycles. The difficulty can be overcomed by extending $\mathcal{EL}^\perp$ with greatest fix-point semantics. It is known that the resulting DL, called $\mathcal{EL}^\perp_{gfp}$, has the FBP (Baader & Distel, 2008; Distel, 2011). The authors then show how to transform an $\mathcal{EL}^\perp_{gfp}$ base into an $\mathcal{EL}^\perp$ base, thus, establishing that $\mathcal{EL}^\perp$ also enjoys the FBP.

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

Example 2. *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 $1 \leq i \leq 3$. 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 $\mathcal{EL}^\perp$ base for $\mathcal{I}$ has a concept expression with role depth exponential in the size of $\mathcal{I}$.*

Proof. (Sketch) We can generalise Example 2 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 $d \geq \prod_{i=1}^n p_i - 1 \geq 2^n$. □

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, 2004b). The DL setting is more challenging than

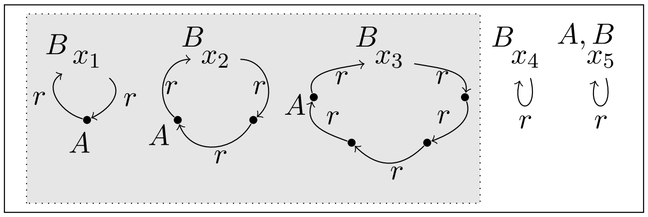

Figure 6: Description graph of an interpretation $\mathcal{I}$. Let $X = \{x_1, x_2, x_3\}$. For all $d < 29$ we have $x_4 \in \mathbf{C}\left(\prod_{x \in X} \mathcal{G}(\mathcal{I})^x_d\right)^{\mathcal{I}} = (B \sqcap \exists r^d.\top)^{\mathcal{I}}$. However, for all $k \geq 29$, $x_4 \notin \mathbf{C}\left(\prod_{x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

classical FCA, and so, this lower bound also holds in the problem we consider. In Section 5, 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 $\mathcal{I}$.

5. 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 $d \geq 0$ 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 $\emptyset \models C \sqsubseteq D$ (that is, any interpretation satisfies $C \sqsubseteq D$).*

For a given $X \subseteq C^{\mathcal{I}}$ and a role depth d there may be multiple MMSCs (always at least one (Borchmann et al., 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 $X = \emptyset$ then $\mathsf{mmsc}\,(X, \mathcal{I}, d) \equiv \bot$ for any interpretation $\mathcal{I}$ and $d \in \mathbb{N}$.

Example 3. *Consider the interpretation $\mathcal{I}$ in Figure 4 and let $X = \{x_1, x_2\}$. We have that*

$$\mathsf{mmsc}\,(X, \mathcal{I}, 1) \equiv \mathsf{City} \sqcap \exists \mathsf{government}.\mathsf{Party} \sqcap \exists \mathsf{partof}.\mathsf{Region}.$$

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

$$\mathsf{mmsc}\,(X, \mathcal{I}, 2) \equiv \mathsf{mmsc}\,(X, \mathcal{I}, 1) \sqcap \exists \mathsf{partof}.(\mathsf{Region} \sqcap \exists \mathsf{capital}.\top)$$

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

$$\mathsf{mmsc}\,(X, \mathcal{I}, 2) \equiv \mathsf{mmsc}\,(X, \mathcal{I}, k)\,.$$

A straightforward (and inefficient) way of computing $\mathsf{mmsc}\,(X, \mathcal{I}, d)$, for a fixed d, would be conjoining every $\mathcal{EL}^\perp$ 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 et al., 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 $d \geq 0$, 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 $\mathcal{EL}^\perp$ 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:*

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

where $\mathsf{v_{num}}(\mathbf{w})$ 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 4. *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 et al., 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 $\{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$, $x' \in \Delta^{\mathcal{I}}$, 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 Appendix B 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 $k \leq d$, 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

$$(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_d^{x_i}, (x_1, \ldots, x_n))$$

to $(\mathcal{G}(\mathcal{I}), x')$. Then, by Claim 1 there is a simulation $Z_k : (\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})_k^{x_i}, (x_1, \ldots, x_n)) \mapsto (\mathcal{G}(\mathcal{I}), x')$ (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 $\{\mathcal{G}_i \mid 1 \leq i \leq n\}$ be n description graphs such that $\mathcal{G}_i = (V_i, E_i, L_i)$. Also let $v_i \in V_i$. Then:*

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

Proof. Let $\mathbf{w}$ be an arbitrary walk in $\prod_{i=1}^{n} \mathcal{G}_i, (v_i)_{1\leq i \leq n}$ 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, $\mathsf{v_{num}}(\mathbf{w}) \leq \prod_{i=1}^{n} \mathsf{mvf}(\mathcal{G}_i, v_i)$. Since $\mathbf{w}$ is arbitrary, we can conclude that

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

□

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 5 illustrates one such case.

Example 5. *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 $\prod_{i=1}^{|\{x_1,x_2\}|} \mathcal{G}(\mathcal{I})$ which starts at the node (x_1, x_2) is: (x_1, x_2),* partof*, (x_5, x_7),* capital*, (x_6, x_2). Thus*

$$\mathsf{mvf}\left(\prod_{i=1}^{|\{x_1,x_2\}|} \mathcal{G}(\mathcal{I}), (x_1, x_2)\right) = 2.$$

Take $X = \{x_1, x_7\}$. Since x_1 and x_7 do not share labels in their outgoing edges

$$\mathsf{mvf}\left(\prod_{i=1}^{|\{x_1,x_7\}|} \mathcal{G}(\mathcal{I}), (x_1, x_7)\right) = 1.$$

Observe that we do not need to index the description graph since all vertices correspond to elements in the same interpretation ($\mathcal{I}$). Moreover, we essentially index the product over the vertices of $\mathcal{G}(\mathcal{I})$ that appear in the vertices of the product graph in each case.

The observations about the MVF in Example 5 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 $1 \leq i \leq n$ 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 $\mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)\right) \leq \mathsf{mvf}\left(\mathcal{G}(\mathcal{I}), x_i\right)$.*

Proof. (Sketch) As it happens in Example 5, 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 $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$. The function $d_{\mathcal{I}} : \mathcal{P}(\Delta^{\mathcal{I}}) \mapsto \mathbb{N}$ is defined as follows:*

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

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

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

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:*

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

Proof. (Sketch) Let $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$. If $k \leq d_{\mathcal{I}}\left(X\right)$, the lemma holds trivially. For $k > d_{\mathcal{I}}\left(X\right)$ 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 $k > d_{\mathcal{I}}\left(X\right)$, 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 so, we can apply Lemma 1 to conclude that: $\mathsf{mmsc}\left(X, \mathcal{I}, d_{\mathcal{I}}\left(X\right)\right)^{\mathcal{I}} \subseteq \mathsf{mmsc}\left(X, \mathcal{I}, k\right)^{\mathcal{I}}$. Otherwise, if $X_{lim} \neq \emptyset$, the lemma is a direct consequence of Definition 5 and Lemma 2. □

In this paper, we write $\mathsf{mmsc}\left(X, \mathcal{I}\right)$ as a shorthand for $\mathsf{mmsc}\left(X, \mathcal{I}, d_{\mathcal{I}}\left(X\right)\right)$. 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. A direct consequence of Lemma 4.4 (vi) of (Borchmann et al., 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*

$$\{\exists r.\mathsf{mmsc}\left(X, \mathcal{I}\right) \mid r \in \mathsf{N_R} \textit{ and } X \subseteq \Delta^{\mathcal{I}}, X \neq \emptyset\}$$

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 $\mathcal{EL}^{\bot}$ 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,*

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

is a finite $\mathcal{EL}^{\bot}$ 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 $\mathcal{B}(\mathcal{I})$. □

In Theorem 2, the number of CIs in $\mathcal{B}(\mathcal{I})$ is double-exponential in the size of $\mathcal{I}$. That is because $\Lambda_{\mathcal{I}}$ is exponential in $M_{\mathcal{I}}$, which in turn is exponential in the size of $\mathcal{I}$. So $\Lambda_{\mathcal{I}}$ is double-exponential in the size of $\mathcal{I}$. Our base $\mathcal{B}(\mathcal{I})$ is the union of two sets of CIs polynomial in the size of $\Lambda_{\mathcal{I}}$ and, therefore, double-exponential in the size of $\mathcal{I}$. The number of CIs in the work by Borchmann et al. (2016) and Distel (2011), proven to be minimal, is also double-exponential in the worst case. This is a consequence of the following facts: (1) minimal bases can be exponential in the size of the formal context (Kuznetsov, 2004a) and (2) the formal context in the mentioned works is exponential in the size of the interpretation given as input. In general, our base may not be minimal and this is left as an open question.

Example 6. *A base with adaptable role depth for the graph in Figure 4 can be*

$$\begin{aligned}\mathcal{B}(\mathcal{I}) = \{\mathsf{City} &\equiv \exists \mathsf{govern}.\mathsf{Party} \sqcap \exists \mathsf{partof}.(\mathsf{Region} \sqcap \exists \mathsf{capital}.\top),\\ \mathsf{City} &\sqsubseteq \exists \mathsf{govern}.\mathsf{Party} \sqcap \exists \mathsf{partof}.\mathsf{Region},\\ \mathsf{Region} &\equiv \exists \mathsf{capital}.\top,\\ \mathsf{Liberal} &\sqsubseteq \mathsf{Party},\\ \mathsf{Organisation} &\sqsubseteq \mathsf{Party}\}.\end{aligned}$$

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 $\mathsf{mmsc}\left(\{x_1\}, \mathcal{I}\right)$ is 10. Additionally, previous works computed the depth of the concept expressions based on the number of reachable vertices (Distel, 2011). Let $\mathsf{reach}(\mathcal{G}, v)$ be the function that returns how many vertices in $\mathcal{G}$ are reachable from v. This function gives an upper bound of the MVF for any vertex of any description graph, that is, $\mathsf{mvf}(\mathcal{G}, v) \leq \mathsf{reach}(\mathcal{G}, v)$, since a vertex can only be visited in a path from v if it is reachable from v. Theorem 3 shows that our approach in fact represents an improvement in terms of depth.

Theorem 3. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite $\mathcal{EL}^{\perp}$ interpretation and $X \subseteq \Delta^{\mathcal{I}}$. Let $d_{\mathcal{I}}(X)$ be the (adaptable) depth according to Definition 5 and $d_{gfp}(X, \mathcal{I})$ be the depth for the $\mathcal{EL}^{\perp}_{gfp}$ MMSC of X w.r.t. $\mathcal{I}$, according to Lemma 5.5 in (Distel, 2011). Then, $d_{\mathcal{I}}(X) \leq d_{gfp}(X, \mathcal{I})$.*

Furthermore, the difference between these two metrics can be quite large. For instance, when a vertex is the root of a tree shaped subgraph, as Example 7 illustrates.

Example 7. *Let $\mathcal{G} = (V, E, L)$ with $V = \{1, \ldots, 2^n - 1\}$ for some $n \in \mathbb{N}$ and*

$$E = \{(i, r, 2i) \mid 1 \leq i < 2^{n-1}\} \cup \{(i, r, 2i+1) \mid 1 \leq i < 2^{n-1}\}.$$

That is, $\mathcal{G}$ is a binary tree rooted in 1. If we take $v = 1$, then $\mathsf{reach}(\mathcal{G}, v) = |V| = 2^n - 1$, while $\mathsf{mvf}(\mathcal{G}, v) = n$. In this case, the number of reachable vertices grows exponentially with n, while the MVF grows only linearly.

Example 8 illustrates how the advantage of adaptable role depth over the depth from the $\mathcal{EL}^{\perp}_{gfp}$ approach.

Example 8. *Consider the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where*

$$\begin{aligned} \Delta^{\mathcal{I}} &= \{x_0, x_1, x_2, x_3\} \\ r^{\mathcal{I}} &= \{(x_0, x_0), (x_1, x_1), (x_2, x_2), (x_3, x_3), (x_0, x_1), (x_0, x_2), (x_0, x_3)\}. \end{aligned}$$

Figure 7 depicts the description graph of $\mathcal{I}$.

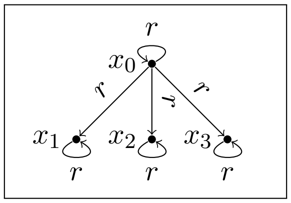

Figure 7: Description graph of the interpretation $\mathcal{I}$ from Example 8

We will compare the depths for the MMSC of the $\{x_0\}$ using our approach and using the $\mathcal{EL}^{\perp}_{gfp}$ approach (Distel, 2011).

First, we will compute the depth for this MMSC with adaptable role depth. As there are walks of arbitrary length starting from x_0 in $\mathcal{G}(\mathcal{I})$, we have from Definition 5 that

$$d_{\mathcal{I}}(\{x_0\}) = \mathsf{mvf}(\mathcal{G}(\mathcal{I}), x_0) \cdot \mathsf{mmvf}(\mathcal{G}(\mathcal{I})) = 2 \cdot 2 = 4.$$

Now, we will focus on the $\mathcal{EL}^{\perp}_{gfp}$ approach, but before we compute the depth we discuss necessary notions that are particular to $\mathcal{EL}^{\perp}_{gfp}$ concepts.

The MMSC of $\{x_0\}$ in $\mathcal{EL}^{\perp}_{gfp}$ is a pair $(A_0, \mathcal{T}_0)$ where A_0 is a fresh concept name and $\mathcal{T}_0$ is a TBox composed of $|\Delta^{\mathcal{I}}|$ equivalences whose left-hand sides are the fresh concept names associated to A_0 and to all vertices reachable from x_0 in $\mathcal{G}(\mathcal{I})$ (see (Distel, 2011), Definitions 2.16 and 2.23).

More specifically, if A_i is a fresh concept name associated with x_i, then $\mathcal{T}_0$ is as follows

$$\begin{aligned}\mathcal{T}_0 = \{&A_0 \equiv \exists r.A_0 \sqcap \exists r.A_1 \sqcap \exists r.A_2 \sqcap \exists r.A_3,\\ &A_1 \equiv \exists r.A_1,\\ &A_2 \equiv \exists r.A_2,\\ &A_3 \equiv \exists r.A_3\}\end{aligned}$$

In other words, $\mathcal{T}_0$ has $|\Delta^{\mathcal{I}}|$-many defined concepts, in symbols, $|\mathsf{N_C}^{def}(\mathcal{T}_0)| = |\Delta^{\mathcal{I}}|$. The conversion from $\mathcal{EL}^{\bot}_{gfp}$ to $\mathcal{EL}^{\bot}$ unravels the MMSC of $\{x_0\}$ up to depth

$$d_{gfp}(\{x_0\}, \mathcal{I}) = |\mathsf{N_C}^{def}(\mathcal{T}_X)| \cdot |\Delta^{\mathcal{I}}| + 1 = 4 \cdot 4 + 1 = 17.$$

Also, if we increase the number of successors of x_0 that also only have themselves as their own successors, $d_{gfp}(\{x_0\}, \mathcal{I})$ increases quadratically with $|\Delta^{\mathcal{I}}|$ while our new depth for the MMSC of $\{x_0\}$ stays the same.

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.

6. Computing the MVF

As discussed in Section 5, the MVF is the key to provide an upper bound for the fixpoint for each MMSC. Moreover, Theorem 3 shows that it improves the existing bound. Still, $\mathsf{reach}(\mathcal{G}, v)$ can be computed in polynomial time, which could be a potential advantage of using this metric over the MVF.

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 $V_1, \dots, V_n$ of V such that for all $1 \leq i \leq n$: 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.*

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, & 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*

$$V^* = \{\mathsf{scc}(\mathcal{G}, u) \mid u \in V\} \quad E^* = \{(\mathsf{scc}(\mathcal{G}, u), \mathsf{scc}(\mathcal{G}, v)) \mid (u, r, v) \in E,\ \mathsf{scc}(\mathcal{G}, u) \neq \mathsf{scc}(\mathcal{G}, v)\}.$$

Also, if $\mathbf{w}^*$ is path in $\mathcal{G}^*$, the weight of $\mathbf{w}^*$, *in symbols $\mathsf{weight}(\mathbf{w}^*)$, 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.

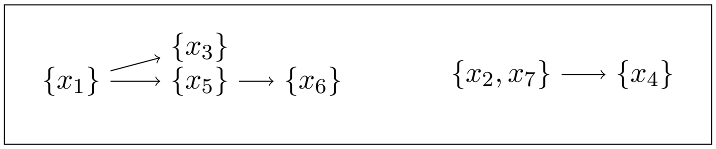

Figure 8: Condensation of the description graph in Figure 4. Every vertex is an SCC of the original graph and the edges indicate accessibility between the SCCs. Also, the condensation has no labels.

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 $\mathbf{w}^* = V_1, \dots, V_m$ 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.

Example 9. *The condensation showing the strongly connected components of the graph in Figure 4 is depicted in Figure 8. The output of $\mathsf{scc}(\mathcal{G}, x_7)$ is the vertex $\{x_2, x_7\}$ in the condensed graph. The function $\mathsf{maxWeight}(\mathcal{G}, \mathsf{scc}(\mathcal{G}, x_7), wgt)$ outputs 2 because*

$$\mathsf{maxWeight}(\mathcal{G}, \mathsf{scc}(\mathcal{G}, x_4), wgt) = 1$$

(the size of the nodes in its scc). The output of $\mathsf{maxWeight}(\mathcal{G}, \mathsf{scc}(\mathcal{G}, x_1), wgt)$ is 4. This is the sum of $\mathsf{maxWeight}$ with the vertices x_3, x_5, and x_6, given as input, which are, respectively, 1, 2, and 1.

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 $\mathsf{maxWeight}$ 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 $\mathsf{scc}(\mathcal{G}, v)$.*

```
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., $\mathsf{mvf}(G, v)$
$V^* \leftarrow \mathsf{SCC}(\mathcal{G})$
$E^* \leftarrow \text{condense}(\mathcal{G}, V^*)$
$\mathcal{G}^* \leftarrow (V^*, E^*)$
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 $wgt[V']$
```

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 $\mathsf{maxWeight}$ 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. □

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 & Toth, 1982) and the depth-first transversal via $\mathsf{maxWeight}$ 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. The modified version calls the function $\mathsf{maxWeight}$ from an unvisited SCC until all vertices are visited, avoiding re-calculating the MVF of vertices along the way.

7. Confident Bases with Adaptable Role Depth

We investigate an extension of our framework to the case where the goal is to obtain a base with GCIs that may only partially match the finite interpretation, as originally considered by Borchmann (2013a, 2013b, 2014) for $\mathcal{EL}^{\bot}_{gfp}$ GCIs. The goal of such approach is to cope with noisy data. We employ a confidence measure to decide whether a GCI should be a consequence of the desired base. We adapt such notion of confidence to $\mathcal{EL}^{\bot}$ as follows.

Definition 9 (Confidence of GCIs (Borchmann, 2014)). *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation over $\mathsf{N_C}$ and $\mathsf{N_R}$, and let C, D be $\mathcal{EL}^{\bot}$ concept expressions. Then the* confidence *of a GCI $C \sqsubseteq D$, in symbols $\mathrm{conf}_{\mathcal{I}}(C \sqsubseteq D)$, is defined as:*

$$\mathrm{conf}_{\mathcal{I}}(C \sqsubseteq D) \coloneqq \begin{cases} 1 & \textit{if } C^{\mathcal{I}} = \emptyset \\ \frac{|(C \sqcap D)^{\mathcal{I}}|}{|C^{\mathcal{I}}|} & \textit{otherwise.} \end{cases}$$

Definition 10 (Confidence-Based Theory of Finite Interpretations (Borchmann, 2014)). *Let $\mathcal{I}$ be a finite interpretation over $\mathsf{N_C}$ and $\mathsf{N_R}$, and let $c \in [0,1]$. Then the* confidence-based theory *of $\mathcal{I}$, in symbols $Th_c(\mathcal{I})$, is defined as:*

$$Th_c(\mathcal{I}) \coloneqq \{C \sqsubseteq D \mid \mathrm{conf}_{\mathcal{I}}(C \sqsubseteq D) \geq c\}$$

As Borchmann (2014) notes, $Th_c(\mathcal{I})$ is not closed by consequence, thus there can be both CIs that are entailed by $Th_c(\mathcal{I})$ that have lower confidence, and there might also exist bases for $Th_c(\mathcal{I})$ which contain axioms with confidence lower than c. A *confident base* $\mathcal{B}$ for an interpretation $\mathcal{I}$ is a finite subset of $Th_c(\mathcal{I})$ such that $\mathcal{B} \models C \sqsubseteq D$ iff $Th_c(\mathcal{I}) \models C \sqsubseteq D$ for all $\mathcal{EL}^{\bot}$ concept inclusions. We can build a confident base for $\mathcal{I}$ as follows

$$\begin{aligned} \mathcal{B}_{\mathrm{c}}(\mathcal{I}) \coloneqq \{&\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I}) \mid Y \subseteq X \subseteq \Delta^{\mathcal{I}}, \\ &1 > \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})) \geq c\}. \end{aligned}$$

Example 10. *A base with confidence $c = 0.4$ for the graph in Figure 4 can be $\mathcal{B}(\mathcal{I})$ as in Example 6 with the additional rules:*

$$\begin{aligned} \{\mathsf{Party} &\sqsubseteq \mathsf{Liberal}, \\ \mathsf{Party} &\sqsubseteq \mathsf{Organisation}, \\ \mathsf{City} &\sqsubseteq \exists \mathsf{govern}.\mathsf{Organisation}, \\ \mathsf{City} &\sqsubseteq \exists \mathsf{govern}.\mathsf{Liberal}, \\ \mathsf{Region} &\sqsubseteq \exists \mathsf{capital}.\mathsf{City}\}. \end{aligned}$$

Next, we prove that replacing the MMSC defined in $\mathcal{EL}^{\bot}$ by our MMSC with adaptable depth in the construction of a confident base still preserves the relation between set inclusion and confidence of GCIs as Lemma 8 shows.

Lemma 8 (Lemma 5.2.12 in (Borchmann, 2014)). *Let $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation, and let $Z \subseteq Y \subseteq X$. Then*

$$\begin{aligned} \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z, \mathcal{I})) = {} & \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})) \\ & \cdot \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(Y, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z, \mathcal{I})). \end{aligned}$$

Proof. The proof is analogous to that of Lemma 5.2.12 in (Borchmann, 2014). We just replace concept expressions in $\mathcal{EL}^{\bot}_{gfp}$ by concept expressions in $\mathcal{EL}^{\bot}$ and adopt the notion of MMSC in our setting with adaptable role depth.

If $\mathsf{mmsc}\,(X, \mathcal{I})^{\mathcal{I}} = \emptyset$, then as $Z \subseteq Y \subseteq X$ we have that $\mathsf{mmsc}\,(Z, \mathcal{I})^{\mathcal{I}} \subseteq \mathsf{mmsc}\,(Y, \mathcal{I})^{\mathcal{I}} \subseteq \mathsf{mmsc}\,(X, \mathcal{I})^{\mathcal{I}}$. Hence $\mathsf{mmsc}\,(Z, \mathcal{I})^{\mathcal{I}} = \mathsf{mmsc}\,(Y, \mathcal{I})^{\mathcal{I}} = \emptyset$ and both sides of the equation become 1.

When $\mathsf{mmsc}\,(X,\mathcal{I})^{\mathcal{I}} \neq \emptyset$ and $\mathsf{mmsc}\,(Y,\mathcal{I})^{\mathcal{I}} = \emptyset$, both sides of the equation become 0, as $\mathsf{mmsc}\,(Z,\mathcal{I})^{\mathcal{I}} = \emptyset$.

Lastly, if both $\mathsf{mmsc}\,(X,\mathcal{I})^{\mathcal{I}} \neq \emptyset$ and $\mathsf{mmsc}\,(Y,\mathcal{I})^{\mathcal{I}} \neq \emptyset$ we have that:

$$\begin{gathered}
\mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z,\mathcal{I})) = \\
\frac{|(\mathsf{mmsc}\,(X,\mathcal{I}) \sqcap \mathsf{mmsc}\,(Z,\mathcal{I}))^{\mathcal{I}}|}{|\mathsf{mmsc}\,(X,\mathcal{I})^{\mathcal{I}}\,|} = \\
\frac{|(\mathsf{mmsc}\,(X,\mathcal{I}) \sqcap \mathsf{mmsc}\,(Y,\mathcal{I}))^{\mathcal{I}}|}{|\mathsf{mmsc}\,(X,\mathcal{I})^{\mathcal{I}}\,|} \cdot \frac{|(\mathsf{mmsc}\,(X,\mathcal{I}) \sqcap \mathsf{mmsc}\,(Z,\mathcal{I}))^{\mathcal{I}}|}{|(\mathsf{mmsc}\,(X,\mathcal{I}) \sqcap \mathsf{mmsc}\,(Y,\mathcal{I}))^{\mathcal{I}}|} = \\
\frac{|(\mathsf{mmsc}\,(X,\mathcal{I}) \sqcap \mathsf{mmsc}\,(Y,\mathcal{I}))^{\mathcal{I}}|}{|\mathsf{mmsc}\,(X,\mathcal{I})^{\mathcal{I}}\,|} \cdot \frac{|(\mathsf{mmsc}\,(Y,\mathcal{I}) \sqcap \mathsf{mmsc}\,(Z,\mathcal{I}))^{\mathcal{I}}|}{|\mathsf{mmsc}\,(Y,\mathcal{I})^{\mathcal{I}}|} = \\
\mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y,\mathcal{I})) \cdot \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(Y,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z,\mathcal{I})).
\end{gathered}$$

□

Before we discuss the implications of Lemma 8 when considering bases (and not only individual GCIs), we need one additional result. Proposition 3 displays another property of our MMSC that mirrors the MMSC notion in $\mathcal{EL}^{\bot}_{gfp}$. First we prove the following proposition.

Proposition 3. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation, $X, Y \subseteq \Delta^{\mathcal{I}}$ and $k \in \mathbb{N}$. Then, it holds that: $X \subseteq Y \implies \mathcal{I} \models \mathsf{mmsc}\,(X,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y,\mathcal{I})$.*

Proof. According to Definition 2, we have that $X \subseteq \mathsf{mmsc}\,(X,\mathcal{I},k)^{\mathcal{I}}$ and $Y \subseteq \mathsf{mmsc}\,(Y,\mathcal{I},k)^{\mathcal{I}}$. Thus, we have:

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

Since $\mathsf{mmsc}\,(Y,\mathcal{I},k)^{\mathcal{I}}$ has role depth k, we obtain from Definition 2 that:

$$\emptyset \models \mathsf{mmsc}\,(X,\mathcal{I},k) \sqsubseteq \mathsf{mmsc}\,(Y,\mathcal{I},k)\,.$$

Then Lemma 4 implies that:

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

□

Finally, as a consequence of the results shown earlier, we can obtain a confident base with adaptable role depth.

Theorem 4. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation, and let $c \in [0,1]$. Let $\mathcal{B}$ be a base of $\mathcal{I}$. Define*

$$\begin{aligned}
\mathrm{Lux}(\mathcal{I},c) :=\{&\mathsf{mmsc}\,(X,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y,\mathcal{I}) \mid \mathsf{mmsc}\,(X,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y,\mathcal{I}) \in \mathcal{B}_{\mathrm{c}}(\mathcal{I}), \\
&\nexists Z \subseteq \Delta^{\mathcal{I}} : \emptyset \models \mathsf{mmsc}\,(Y,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z,\mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(X,\mathcal{I})\}.
\end{aligned}$$

Then $\mathrm{Lux}(\mathcal{I},c) \cup \mathcal{B}$ *is a finite confident base of* $\mathrm{Th}_c(\mathcal{I})$*.*

Proof. The proof is analogous to that of Theorem 5.2.13 in (Borchmann, 2014). It is sufficient to show that all CIs in $\mathcal{B}_{\mathrm{c}}(\mathcal{I})$ are entailed by $\mathrm{Lux}(\mathcal{I}, c)$. Let $(\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})) \in \mathcal{B}_{\mathrm{c}}(\mathcal{I})$.

Then $Y \subseteq X \subseteq \Delta^{\mathcal{I}}$, and from Proposition 3 we get: $\mathcal{I} \models \mathsf{mmsc}\,(Y, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(X, \mathcal{I})$. Since $\Delta^{\mathcal{I}}$ is finite, there are sets $Z_n \subsetneq \cdots \subsetneq Z_0 = \Delta^{\mathcal{I}}$ satisfying:

$$\mathsf{mmsc}\,(Y, I) \equiv_{\mathcal{I}} \mathsf{mmsc}\,(Z_n, \mathcal{I}) \sqsubset_{\mathcal{I}} \cdots \sqsubset_{\mathcal{I}} \mathsf{mmsc}\,(Z_0, \mathcal{I}) \equiv_{\mathcal{I}} \mathsf{mmsc}\,(X, \mathcal{I})$$

such that there are no sets $W \subseteq \Delta^{\mathcal{I}}$ with:

$$\mathsf{mmsc}\,(Z_i, \mathcal{I}) \sqsubset_{\mathcal{I}} \mathsf{mmsc}\,(W, \mathcal{I}) \sqsubset_{\mathcal{I}} \mathsf{mmsc}\,(Z_{i-1}, \mathcal{I}) \tag{1}$$

for any $i \in \{1, \ldots, n\}$. Then, by Lemma 8 it is true that:

$$\mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})) = \prod_{i=0}^{n-1} \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(Z_i, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z_{i+1}, \mathcal{I})).$$

As the confidence of a CI is always an element of $[0, 1]$, we obtain from this equality that:

$$\mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(Z_i, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z_{i+1}, \mathcal{I})) \geq \mathrm{conf}_{\mathcal{I}}(\mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})) \geq c$$

and due to Equation 1 we obtain $(\mathsf{mmsc}\,(Z_i, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z_{i+1}, \mathcal{I})) \in \mathrm{Lux}(\mathcal{I}, c)$ for all $i \in \{0, \ldots, n-1\}$.

Since $\{\mathsf{mmsc}\,(Z_i, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Z_{i+1}, \mathcal{I}) \mid i \in \{0, \ldots, n-1\}\} \models \mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})$ we obtain:

$$\mathrm{Lux}(\mathcal{I}, c) \models \mathsf{mmsc}\,(X, \mathcal{I}) \sqsubseteq \mathsf{mmsc}\,(Y, \mathcal{I})$$

□

8. Conclusion

We introduce a way of computing $\mathcal{EL}^{\perp}$ bases from finite interpretations that adapts the role depth of concepts according to the structure of interpretations. Our definition relies on a notion that relates vertices in a graph to sets of vertices, called MVF. The role depth of expressions computed with our approach is guaranteed to not exceed the role depth of expressions used to compute bases in the literature and, in some cases, it can give expressions with an exponentially smaller role depth. We have also shown that the MVF computation can be performed in polynomial time in the size of the underlying graph structure. In addition, we considered the problem of mining CIs in the presence of noise in the dataset. We applied the confidence measure from association rule mining to define confident bases. That is, bases with CIs that do not need to hold completely in the dataset but on a large proportion of the domain of elements. Our $\mathcal{EL}^{\perp}$ 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 and implement our approach using knowledge graphs as datasets.

Acknowledgements

We would like to thank Prof. Sebastian Rudolph for contributing with the examples in Figure 5 and Francesco Kriegel for the discussion in the early phases of this work. We also would like to thank the anonymous reviewers for their valuable remarks. Ozaki is supported by the Research Council of Norway, project number 316022, led by herself. Guimarães is supported by the ERC project LOPRE (819416), led by Prof. Saket Saurabh. 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.

Appendix A. Proofs for Section 4

We prove that $\mathcal{EL}_{rhs}$ and $\mathcal{EL}_{lhs}$ do not have the finite base property (Propositions 4 and 5).

Proposition 4. *$\mathcal{EL}^\bot_{rhs}$ does not have the finite base property.*

Proof. Consider the interpretation $\mathcal{I} = (\{x_1, x_2\}, \cdot^\mathcal{I})$ where $\{(x_1, x_2), (x_2, x_2)\} = r^\mathcal{I}$, $\{x_1\} = A^\mathcal{I}$ and every other concept and role name is mapped by $\cdot^\mathcal{I}$ to $\emptyset$ (Figure 5 (*i*)). In $\mathcal{I}$, $A^\mathcal{I} = \{x_1\}$ and for all $n \in \mathbb{N}^+$, $x_1 \in (\exists r^n.\top)^\mathcal{I}$. Assume that $\mathcal{B}$ is a base for $\mathcal{I}$ and $\mathcal{EL}_{rhs}$. As $\mathcal{B}$ is a (finite) TBox formulated in $\mathcal{EL}_{rhs}$ with symbols from $\Sigma_\mathcal{I}$, it can only have CIs of the form $A \sqsubseteq C$. Since $\mathcal{I} \models A \sqsubseteq \exists r^n.\top$, for all $n \in \mathbb{N}^+$, it follows that $\mathcal{B}$ is infinite, which is a contradiction. □

Proposition 5. *$\mathcal{EL}^\bot_{lhs}$ does not have the finite base property.*

Proof. In this proof, assume that CIs are formulated in $\mathcal{EL}_{lhs}$. Consider the interpretation

$$\mathcal{I} = (\{x_1, x_2, x_3, x_4\}, \cdot^\mathcal{I})$$

with

$$\begin{aligned} r^\mathcal{I} &= \{(x_2, x_2), (x_4, x_4), (x_3, x_2)\} \\ s^\mathcal{I} &= \{(x_1, x_2), (x_3, x_4)\} \\ A^\mathcal{I} &= \{x_1\} \\ B^\mathcal{I} &= \{x_2\} \end{aligned}$$

and every other concept and role name is mapped by $\cdot^\mathcal{I}$ to $\emptyset$ (see Figure 5 (*ii*)).

By definition of $\mathcal{I}$, for all $n \in \mathbb{N}^+$, we have that $\mathcal{I} \models \exists s.\exists r^n.B \sqsubseteq A$. So if $\mathcal{B}$ is a base for $\mathcal{EL}_{lhs}$ and $\mathcal{I}$ then $\mathcal{B} \models \exists s.\exists r^n.B \sqsubseteq A$ for all $n \in \mathbb{N}^+$. Now, observe that there is no D such that

1. $\emptyset \models \exists s.\exists r^n.B \sqsubseteq D$,

2. $\emptyset \not\models D \sqsubseteq \exists s.\exists r^n.B$ (where $\not\models$ means 'does not entail'),

3. and $\mathcal{I} \models D \sqsubseteq A$.

The reason for the above is because $x_3 \in D^{\mathcal{I}}$ for all D satisfying (1) and (2) but $x_3 \notin A^{\mathcal{I}}$. Moreover, there is no $k \in \mathbb{N}^+$ such that $\mathcal{I} \models \exists r^k.B \sqsubseteq B$ or $\mathcal{I} \models \exists r^k.B \sqsubseteq A$ (because $x_3 \notin B^{\mathcal{I}}$ and $x_2, x_3 \notin A^{\mathcal{I}}$ but $x_2, x_3 \in (\exists r^k.B)^{\mathcal{I}}$). So, $\mathcal{B}$ can only entail $\exists s.\exists r^n.B \sqsubseteq A$ if there is a CI in $\mathcal{B}$ with a concept equivalent to $\exists s.\exists r^n.B$. This concept needs to have role depth n. Since $\mathcal{B} \models \exists s.\exists r^n.B \sqsubseteq A$ for all $n \in \mathbb{N}^+$, there are CIs with role depth n for all $n \in \mathbb{N}^+$. This means that $\mathcal{B}$ cannot be finite. □

Next, we prove the result which shows that the depth of roles in a base has an exponential lower bound.

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. For any $n \geq 1$, we consider the interpretation $\mathcal{I}$ where for every $i \in \{1, \cdots, n\}$ and $k \geq 1$, there is $x_i \in \Delta^{\mathcal{I}}$ that satisfies $x_i \in (\exists r^{k \cdot p_i - 1}.A)^{\mathcal{I}}$, $x_i \in B^{\mathcal{I}}$, and $x_i \notin (\exists r^l.A)^{\mathcal{I}}$ for $l \notin \{k \cdot p_i - 1 \mid k \geq 1\}$ where p_i is the i-th prime number.

We know that $\min(\bigcap_{i=1}^{n} \{k \cdot p_i \mid k \geq 1\}) = \prod_{i=1}^{n} p_i$ (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, $d = min(\bigcap_{i=1}^{n} \{k \cdot p_i - 1 \mid k \geq 1\})$, is the minimal number such that $B^{\mathcal{I}} = (\exists r^d.A)^{\mathcal{I}}$. Since $d = \prod_{i=1}^{n} p_i - 1 \geq 2^n$, the statement holds because a base for $\mathcal{I}$ should entail the CI $B \sqsubseteq \exists r^d.A$. For this to happen, it should have a CI with role depth at least d. □

Appendix B. Proofs for Section 5

Now we will prove Claim 1, which is part of Lemma 2 that underlies our approach. Before that, we need an additional result regarding simulations, which allows us to view them as functions. The next lemma is a direct consequence of proposition A.7 by Borchmann et al.(2016).

Lemma 9. *Let $Z : (\mathcal{G}_1, v_1) \mapsto (\mathcal{G}_2, v_2)$ be a simulation where $\mathcal{G}_1 = (V_1, E_1, L_1)$ is a tree-shaped $\mathcal{EL}^{\bot}$ description graph rooted in v_1, and $\mathcal{G}_2 = (V_2, E_2, L_2)$ is an $\mathcal{EL}^{\bot}$ description graph. Then, there exists a simulation $Z' : (\mathcal{G}_1, v_1) \mapsto (\mathcal{G}_2, v_2)$ such that for every $v \in V_1$, there is at most one $w \in V_2$ such that $(v, w) \in Z'$.*

Proof. According to Proposition A.7 by Borchmann et al.(2016), since $\mathcal{G}_1$ is a tree-shaped $\mathcal{EL}^{\bot}$ description graph, there existence of the simulation $Z : (\mathcal{G}_1, v_1) \mapsto (\mathcal{G}_2, v_2)$ implies that there is an homomorphism φ from $(\mathcal{G}_1, v_1)$ to $(\mathcal{G}_2, v_2)$. Using this homomorphism, we can simply take $Z' = \{(v, \varphi(v)) \mid v \in V_1\}$. □

Now we proceed to the claim's actual proof.

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 $k \in \mathbb{N}$.

Proof. Let $\mathcal{G}, \mathcal{G}', v$ and v' as stated earlier and consider the unravellings

$$\mathcal{G}^v_d = (V_d, E_d, L_d) \text{ and } \mathcal{G}^v_k = (V_k, E_k, L_k)$$

of $\mathcal{G}$. Now, assume that there is a simulation $Z_d : (\mathcal{G}^v_d, v) \mapsto (\mathcal{G}', v')$. By Lemma 9, we can assume w.l.o.g. that for each $\mathbf{w} \in V_d$ there exists at most one $u \in V'$ such that $(\mathbf{w}, u) \in Z_d$. Therefore, we can define a function z such that $z(\mathbf{w})$ is the only vertex in V' such that $(\mathbf{w}, z(\mathbf{w})) \in Z_d$.

If $k \leq d$ then, as $\mathcal{G}^v_k$ is a subtree of $\mathcal{G}^v_d$ (and thus, $V_k \subseteq V_d$), one can just take $Z_k = \{(\mathbf{w}, z(\mathbf{w})) \mid \mathbf{w} \in V_k\}$ as simulation. We now argue about the case where $k > d$. Recall that the function δ returns the vertex of a graph that occurs at end of a path. We show that in any path of length d in $\mathcal{G}^v_d$, there are two vertices $\mathbf{w}_1$ and $\mathbf{w}_2$ such that $\delta(\mathbf{w}_1) = \delta(\mathbf{w}_2)$ and $z(\mathbf{w}_1) = z(\mathbf{w}_2)$.

In what follows, we use the fact that unravellings are trees, and thus, for each vertex in an unravelling, there is exactly one path starting from the root to it. So we can refer to this path without ambiguity. Moreover, if $\mathbf{w}$ is a vertex in an unravelling with root v, then the path distance of $\mathbf{w}$ is the length of the path from v to $\mathbf{w}$.

Now, let $\mathbf{w} = w_0 r_0 \dots r_{n-1} w_n$ be a vertex in V_d and let $\mathbf{w}_i = w_0 r_0 \dots r_{i-1} w_i$ for $0 \leq i \leq n$ be the vertices in the path from v to $\mathbf{w}$ ($\mathbf{w}_0 = v$ and $\mathbf{w}_n = \mathbf{w}$). The path from v to $\mathbf{w}$ in $\mathcal{G}^v_d$ determines a walk $\mathbf{w}^*$ in $\mathcal{G}$ starting at v as follows:

$$\mathbf{w}^* = \delta(\mathbf{w}_0) r_0 \dots r_{n-1} \delta(\mathbf{w}_n).$$

Due to the definition of mvf there can be at most $\mathsf{mvf}(\mathcal{G}, v)$ distinct values of δ for all vertices in the path from v to $\mathbf{w}$, that is, $|\{\delta(\mathbf{w}_i) \mid 0 \leq i \leq n\}| \leq \mathsf{mvf}(\mathcal{G}, v)$.

As Z_d is a simulation, the path from v to $\mathbf{w}$ also determines a walk in $\mathcal{G}'$ starting at v':

$$\mathbf{w}' = z(\mathbf{w}_0) r_0 \dots r_{n-1} z(\mathbf{w}_n).$$

Again, due to the definition of mvf there can be at most $\mathsf{mvf}(\mathcal{G}', v')$ distinct values of z for all vertices in the path from v to $\mathbf{w}$, that is, $|\{z(\mathbf{w}_i) \mid 0 \leq i \leq n\}| \leq \mathsf{mvf}(\mathcal{G}, v)$. Therefore, there are at most $\mathsf{mvf}(\mathcal{G}, v) \cdot \mathsf{mvf}(\mathcal{G}', v') = d$ distinct pairs $(\delta(\mathbf{w}'), z(\mathbf{w}'))$, where $\mathbf{w}'$ is a vertex in the path from v to $\mathbf{w}$, i.e.,

$$|\{(\delta(\mathbf{w}_i), z(\mathbf{w}_i)) \mid 0 \leq i \leq n\}| \leq d.$$

If a vertex $\mathbf{w}$ has path distance d from v in $\mathcal{G}^v_d$, then there are $d+1$ vertices in the path from v to $\mathbf{w}$. As there are at most d distinct pairs $(\delta(\mathbf{w}'), z(\mathbf{w}'))$, where $\mathbf{w}'$ is a vertex in this path, and $d+1$ vertices in the path from v to $\mathbf{w}$, the pigeonhole principle implies that there will be two vertices $\mathbf{w}_1, \mathbf{w}_2 \in V_d$ in the path from v to $\mathbf{w}$ such that both $z(\mathbf{w}_1) = z(\mathbf{w}_2)$ and $\delta(\mathbf{w}_1) = \delta(\mathbf{w}_2)$.

Let $\overline{V} \subseteq V_d$ be the set of all vertices such that there are no two distinct vertices $\mathbf{w}_1$ and $\mathbf{w}_2$ on the path from v to $\mathbf{w}$ with $\delta(\mathbf{w}_1) = \delta(\mathbf{w}_2)$ and $z(\mathbf{w}_1) = z(\mathbf{w}_2)$. Because of the previous argument, $\overline{V}$ contains only vertices whose path distance from v is strictly less than d.

Since $\mathcal{G}^v_d$ is a description tree with root v, there is exactly one directed path from v to any given vertex $\mathbf{w} \in V_d$. Hence, if $\mathbf{w} \in \overline{V}$ then every vertex $\mathbf{w}'$ on the path from v to $\mathbf{w}$ in $\mathcal{G}^v_d$ is also in $\overline{V}$. In other words, $\overline{V}$ spans a subtree of $\mathcal{G}^v_d$.

Now, let us consider the set $\overline{V}^+$ composed by the direct successors of the leaves of the subtree determined by $\overline{V}$, that is, $\overline{V}^+ = \{w_0r_0 \dots r_{n-1}w_n \in V_d \backslash \overline{V} \mid w_0r_0 \dots r_{n-2}w_{n-1} \in \overline{V}\}$. Since each vertex in $\overline{V}$ has path distance at most $d-1$ from v, each vertex $\overline{V}^+$ has path distance at most d from v. Together with the fact that $\overline{V}$ spans a subtree of $\mathcal{G}^v_d$, for each vertex $\mathbf{w} \in V_d$ with path distance d from v, there is exactly one vertex $\mathbf{w}' \in \overline{V}^+$ in the path from v to $\mathbf{w}$ (including the extremities).

As we assume $k > d$, we know that $\mathcal{G}^v_d$ is a subtree of $\mathcal{G}^v_k$, hence $\overline{V}$ also spans a subtree of $\mathcal{G}^v_k$. Therefore $\overline{V} \cup \overline{V}^+ \in V_k$ and for every vertex $\mathbf{w} \in V_k$ there is exactly one vertex $\mathbf{w}'$ in $\overline{V}^+$ in the path from v to $\mathbf{w}$ in $\mathcal{G}^v_k$. For each vertex $\mathbf{w} \in V_k$, such $\mathbf{w}'$ can be used to build a simulation from Z_d that includes $\mathbf{w}$, as we will show next.

For each vertex $\mathbf{w} \in \overline{V} \cup \overline{V}^+$, there is exactly one vertex $\mathbf{w}'$ in $\overline{V}$ in the path from v to $\mathbf{w}$ such that $\delta(\mathbf{w}) = \delta(\mathbf{w}')$ and $z(\mathbf{w}) = z(\mathbf{w}')$. Therefore, we can define a function $s : \overline{V} \cup \overline{V}^+ \mapsto \overline{V}$ which retrieves such vertex for every $\mathbf{w} \in \overline{V} \cup \overline{V}^+$.

Now, we can use this function s to find an alternative path in V_d for each vertex in V_k when extending Z_d to the vertices in $V_k \setminus V_d$. This notion is formalised by the function $f : V_k \mapsto \overline{V}$ defined next, where $\mathbf{w} = w_0r_0 \dots w_{|\mathbf{w}|-1}r_{|\mathbf{w}|-1}w_{|\mathbf{w}|}$.

$$f(\mathbf{w}) = \begin{cases} s(\mathbf{w}) & \text{if } \mathbf{w} \in \overline{V} \cup \overline{V}^+ \\ f(f(w_0r_0 \dots w_{|\mathbf{w}|-1})r_{|\mathbf{w}|-1}w_{|\mathbf{w}|}) & \text{otherwise.} \end{cases}$$

To clarify the rôle of f in this proof, consider a vertex $\mathbf{w} = w_0r_0 \dots r_{n-1}w_n \in V_k$ with $n > d$. As before, let $\mathbf{w}_i = w_0r_0 \dots r_{i-1}w_i$ for $0 \leq i \leq n$ be the vertices in the path from v to $\mathbf{w}$. Since the path distance from v to $\mathbf{w}$ is more than d, we know that there is one $1 \leq m \leq n$ such that $\mathbf{w}_m \in \overline{V}^+$. We also know that there is one $0 \leq j < m$ such that $s(\mathbf{w}_m) = \mathbf{w}_j$. When applying f to $\mathbf{w}$, we obtain the following:

$$f(\mathbf{w}) = f(\dots f \dots f(f(\mathbf{w}_m)r_mw_{m+1}) \dots)r_{n-1}w_n).$$

Since $\mathbf{w}_m \in \overline{V}^+$, we have that $f(\mathbf{w}_m) = \mathbf{w}_j$, which is closer to v than $\mathbf{w}_m$. As a consequence of $s(\mathbf{w}_j) = \mathbf{w}_m$ and the definitions of unravelling and simulation, we know that $(\delta(\mathbf{w}_j), r_m, \delta(\mathbf{w}_{m+1})) \in E$ and $(z(\mathbf{w}_j), r_m, z(\mathbf{w}_{m+1})) \in E'$. Because the relation between vertices in $\overline{V}^+$ and their image via the function s holds in each step of the recursion, we can add $(\mathbf{w}, z(f(\mathbf{w}))$ to Z_d for every vertex in V_k creating a new simulation.

We use this observation to define the relation Z_k as:

$$Z_k = \{(\mathbf{w}, z(f(\mathbf{w})) \mid \mathbf{w} \in V_k\}.$$

Now we show that Z_k is a simulation from $(\mathcal{G}^v_k, v)$ to $(\mathcal{G}', v')$.

1. Since $v \in \overline{V}$ and Z_d is a simulation satisfying the property of Lemma 9, $(v, z(f(v))) = (v, z(v)) = (v, v')$.

2. Since Z_d is a simulation and $f(\mathbf{w}) \in V_d$:

$$\begin{aligned} L_k(\mathbf{w}) = L(\delta(\mathbf{w})) &= L(\delta(f(\mathbf{w}))) \\ &= L_d(f(\mathbf{w})) \subseteq L'(z(f(\mathbf{w}))). \end{aligned}$$

3. Let $\mathbf{w} \in V_k$ and assume that $(\mathbf{w}, r, \mathbf{w}ry) \in E_k$.

   If $\mathbf{w}ry \in \overline{V} \cup \overline{V}^+$, then $\mathbf{w} \in \overline{V}$. Therefore, $(\mathbf{w}, r, \mathbf{w}ry) \in E_d$. We also have:

$$\begin{aligned}(z(f(\mathbf{w})), r, z(f(\mathbf{w}ry))) &= (z(s(\mathbf{w})), r, z(s(\mathbf{w}ry))) \\ &= (z(\mathbf{w}), r, z(\mathbf{w}ry)).\end{aligned}$$

   Moreover, $(z(\mathbf{w}), r, z(\mathbf{w}ry)) \in E'$ because Z_d is a simulation. Finally, by construction, $(\mathbf{w}ry, z(\mathbf{w}ry)) \in Z_k$.

   Otherwise, if $\mathbf{w}ry \notin \overline{V} \cup \overline{V}^+$, we have that $f(\mathbf{w}ry) = f(f(\mathbf{w})ry)$. Since $f(\mathbf{w}) \in \overline{V}$, $f(\mathbf{w})ry \in \overline{V} \cup \overline{V^+}$ and consequently $f(f(\mathbf{w})ry) = s(f(\mathbf{w})ry)$. By the definition of s: $z(s(f(\mathbf{w})ry)) = z(f(\mathbf{w})ry) = z(f(\mathbf{w}ry))$. Since Z_d is a simulation and $f(\mathbf{w}) \in V_d$, it holds that $(z(f(\mathbf{w})), r, z(f(\mathbf{w})ry)) = (z(f(\mathbf{w})), r, z(f(\mathbf{w}ry))) \in E'$. Thus, $(\mathbf{w}ry, z(f(\mathbf{w})ry)) = (\mathbf{w}ry, z(f(\mathbf{w}ry)) \in Z_k$ which concludes the proof of (S3) for Z_k.

Therefore, Z_k is a simulation from $(\mathcal{G}_k^v, v)$ to $(\mathcal{G}', v')$, which proves the claim. □

Lemma 3 refers to walks in a product graph. To simplify its proof we highlight a relationship between walks in the product graph and walks in their factors via Proposition 6.

Proposition 6. *Let $\mathcal{G}_1, \dots, \mathcal{G}_n$ be n description graphs, with $\mathcal{G}_i = (V_i, E_i, L_i)$ for $1 \leq i \leq n$. It holds that, for each walk $\mathbf{w}$ in $\prod_{i=1}^n \mathcal{G}_i$ starting at $(v_1, \dots, v_n)$, there is a walk in $\mathcal{G}_i$ starting at v_i with the same length, for all $1 \leq i \leq n$.*

Proof. Let $\mathbf{w}$ be a walk in $\prod_{i=1}^n \mathcal{G}_i$ starting in $(v_1, \dots, v_n)$ with length m as follows:

$$\mathbf{w} = (w_{1,0}, \dots, w_{n,0})r_0 \dots r_{m-1}(w_{1,m-1}, \dots, w_{n,m})).$$

The walk $\mathbf{w}_i = w_{i,0}r_0 \dots r_{m-1}w_{i,m}$ is a walk in $\mathcal{G}_i$ because $w_{i,j} \in V_i$ for $0 \leq j < m$ and $(w_{i,j}, r_j, w_{i,j+1}) \in E$ due to the definition of product. As $w_{i,0} = v_i$, by construction of $\mathbf{w}$, $\mathbf{w}_i$ starts at v_i. Additionally, by construction, $\mathbf{w}_i$ has length m, which concludes the proof. □

We use Proposition 6 in Lemma 3 below.

Lemma 3. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite interpretation and $X = \{x_1, \dots, x_n\} \subseteq \Delta^{\mathcal{I}}$. If for some $1 \leq i \leq n$ 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* $\mathsf{mvf}\,(\prod_{i=1}^n \mathcal{G}(\mathcal{I}), (x_1, \dots, x_n)) \leq \mathsf{mvf}\,(\mathcal{G}(\mathcal{I}), x_i)$.

Proof. Let $X = \{x_1, \dots, x_n\} \subseteq \Delta^{\mathcal{I}}$ and let

$$\begin{aligned}X_{lim} = \{x \in X \mid \exists m \in \mathbb{N} :& \text{ every walk} \\ &\text{starting from } x \text{ in } \mathcal{G}(\mathcal{I}) \text{ has length} \leq m\}.\end{aligned}$$

Assume $X_{lim} \neq \emptyset$ and let $x' \in X_{lim}$ be such that

$$\mathsf{mvf}(\mathcal{G}(\mathcal{I}), x') = \min_{x \in X_{lim}} \mathsf{mvf}(\mathcal{G}(\mathcal{I}), x).$$

Since $x' \in X_{lim}$, every walk in $\mathcal{G}(\mathcal{I})$ starting at x' has length bounded by $\mathsf{mvf}(\mathcal{G}(\mathcal{I}), x') - 1$. Due to the definition of product of description graphs (recall how the edges are built), this limitation extends to every walk in $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$ starting at $(x_1, \ldots, x_n)$: they have length at most $\min_{x \in X_{lim}} \mathsf{mvf}(\mathcal{G}(\mathcal{I}), x) - 1$. If there was a longer walk, there would be also a walk in in $\mathcal{G}(\mathcal{I})$ starting at x' with the same length due to Proposition 6. □

In the following, we prove that our adaptable role depth yields an upper bound of the actual fixpoint for an MMSC.

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:*

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

Proof. Let $X = \{x_1, \ldots, x_n\} \subseteq \Delta^{\mathcal{I}}$ and

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

If $k \leq d_{\mathcal{I}}(X)$ the lemma holds trivially. For $k > d_{\mathcal{I}}(X)$ we divide the proof in two cases. First, if $X_{lim} \neq \emptyset$ 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 $\mathsf{mvf}\left(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)\right) - 1 = d_{\mathcal{I}}(X)$.

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:

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

Otherwise, if $X_{lim} = \emptyset$, we can use the fact that $\mathsf{mmvf}(\mathcal{G}) \geq \mathsf{mvf}(\mathcal{G}, x')$ $\forall x' \in \Delta_{\mathcal{I}}$ to obtain:

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

Hence, if $X_{lim} = \emptyset$, the lemma is a direct consequence of Definition 5 and Lemma 2. □

To prove that $\mathcal{B}(\mathcal{I})$ defined in Theorem 2 is a base, we first recall a result related to the notion of MMSC.

Lemma 10. *(Borchmann et al., 2016) Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite $\mathcal{EL}^{\bot}$ interpretation. For all $X \subseteq \Delta^{\mathcal{I}}$ and $k \in \mathbb{N}$, it holds that $\emptyset \models \mathsf{mmsc}\left(\mathsf{mmsc}\left(X, \mathcal{I}, k\right)^{\mathcal{I}}, \mathcal{I}, k\right) \equiv \mathsf{mmsc}\left(X, \mathcal{I}, k\right)$.*

We will also need a property regarding the construction of concept expressions with MMSCs.

Lemma 11 (Adaptation of Proposition A.1 from (Borchmann et al., 2016))**.** *For all $\mathcal{EL}^{\bot}$ concept expressions C, D over $\mathsf{N_C} \cup \mathsf{N_R}$ and all $r \in \mathsf{N_R}$ it holds that:*

$$(\mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right) \sqcap D)^{\mathcal{I}} = (C \sqcap D)^{\mathcal{I}},$$

$$(\exists r.(\mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)))^{\mathcal{I}} = (\exists r.C)^{\mathcal{I}}.$$

Then, we define for each concept expression C and interpretation $\mathcal{I}$ a specific concept in $\Lambda_{\mathcal{I}}$ which is called the lower approximation of C in $\mathcal{I}$. We recall that, for $X \subseteq \Delta^{\mathcal{I}}$, we write $\mathsf{mmsc}\,(X, \mathcal{I})$ as a shorthand for $\mathsf{mmsc}\,(X, \mathcal{I}, d_{\mathcal{I}}\,(X))$.

Definition 11 (Lower Approximation (adapted from Definition 5.4 in (Distel, 2011)))**.** *Let C be an $\mathcal{EL}^{\bot}$ concept expression and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a model. Also let $\mathsf{N_C} \cup \mathsf{N_R}$ be a finite signature and $\mathcal{EL}^{\bot}(\mathsf{N_C}, \mathsf{N_R})$ the set of all $\mathcal{EL}^{\bot}$ concept expressions over $\mathsf{N_C} \cup \mathsf{N_R}$. Then, there are concept names $U \subseteq \mathsf{N_C}$ and pairs $\Pi \subseteq \mathsf{N_R} \times \mathcal{EL}^{\bot}(\mathsf{N_C}, \mathsf{N_R})$ such that:*

$$C = \bigsqcap U \sqcap \bigsqcap_{(r,E)\in\Pi} \exists r.E$$

We define the lower approximation of C in $\mathcal{I}$ as:

$$\begin{aligned} &approx(C, \mathcal{I}) = \\ &\quad \begin{cases} \bigsqcap U \sqcap \bigsqcap_{(r,E)\in\Pi} \exists r.\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right) & \text{if } C \neq \bot, \\ \bot & \text{otherwise.} \end{cases} \end{aligned}$$

Concept expressions built according to Definition 11 are always elements of $\Lambda_{\mathcal{I}}$ because they are a conjunction of elements in $M_{\mathcal{I}}$ (Definition 6). Next, with a straightforward, but nevertheless important, adaptation of the Lemma 5.8 from (Distel, 2011) we prove that the lower approximation of a concept and the concept itself have the same extension.

Lemma 12. *Let C be an $\mathcal{EL}^{\bot}$ concept expression and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a model. It holds that*

$$\mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = approx(C, \mathcal{I})^{\mathcal{I}} = C^{\mathcal{I}}.$$

Proof. If $C = \bot$ then $\mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = approx(C, \mathcal{I})^{\mathcal{I}} = \emptyset$. Otherwise, there are concept names $U \subseteq \mathsf{N_C}$ and pairs $\Pi \in \mathsf{N_R} \times \mathcal{EL}^{\bot}(\mathsf{N_C}, \mathsf{N_R})$ such that

$$C = \bigsqcap U \sqcap \bigsqcap_{(r,E)\in\Pi} \exists r.E$$

Using Lemma 11 we obtain:

$$\begin{aligned} C^{\mathcal{I}} &= (\bigsqcap U \sqcap \bigsqcap_{(r,E)\in\Pi} \exists r.E)^{\mathcal{I}} \\ &= (\bigsqcap U \sqcap \bigsqcap_{(r,E)\in\Pi} \exists r.\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right))^{\mathcal{I}} \\ &= (approx(C, \mathcal{I}))^{\mathcal{I}} \end{aligned}$$

Finally, we can apply Lemma 5 obtaining $\mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = approx(C, \mathcal{I})^{\mathcal{I}}$. □

Using these results, we can conclude that for each MMSC there is a concept expression in $\Lambda_{\mathcal{I}}$ with the same extension in $\mathcal{I}$. With this observation we can we can proceed to Theorem 2's proof.

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

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

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

Proof. As $\Lambda_{\mathcal{I}}$ is finite, so it is $\mathcal{B}(\mathcal{I})$. The concept inclusions are clearly sound and the soundness of the equivalences is due to Lemma 5.

Let $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be an arbitrary interpretation such that $\mathcal{J} \models \mathcal{B}(\mathcal{I})$. For completeness, we prove that for any $\mathcal{EL}^{\bot}$ concept expression C, $\mathcal{J} \models C \equiv \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)$. We prove this claim by induction of the structure of C.

Base case: If $C = \bot$ or $C = A$ where $A \in \mathsf{N_C}$, then $C \in \Lambda_{\mathcal{I}}$, by definition of $\Lambda_{\mathcal{I}}$. Then, by definition of $\mathcal{B}(\mathcal{I})$, we have that $C \equiv \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right) \in \mathcal{B}(\mathcal{I})$.

Step case ($\sqcap$): Suppose $C = E \sqcap F$ and the claim holds for E and F. By the inductive hypothesis, $\mathcal{B}(\mathcal{I}) \models E \equiv \mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)$ and $\mathcal{B}(\mathcal{I}) \models F \equiv \mathsf{mmsc}\left(F^{\mathcal{I}}, \mathcal{I}\right)$. Hence, for all interpretations $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{B}(\mathcal{I})$, we have that $E^{\mathcal{J}} = \mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{J}}$ and $F^{\mathcal{J}} = \mathsf{mmsc}\left(F^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{J}}$. By Lemma 12, there are $\overline{E}, \overline{F} \in \Lambda_{\mathcal{I}}$ such that $\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = \overline{E}^{\mathcal{I}}$ and $\mathsf{mmsc}\left(F^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = \overline{F}^{\mathcal{I}}$. Moreover, by Lemma 5, $\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = E^{\mathcal{I}}$ and $\mathsf{mmsc}\left(F^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = F^{\mathcal{I}}$. Therefore $(\overline{E} \sqcap \overline{F})^{\mathcal{I}} = \overline{E}^{\mathcal{I}} \cap \overline{F}^{\mathcal{I}} = E^{\mathcal{I}} \cap F^{\mathcal{I}} = (E \sqcap F)^{\mathcal{I}}$.

As $\overline{E} \sqcap \overline{F} \in \Lambda_{\mathcal{I}}$ (up to logical equivalence), $\overline{E} \sqcap \overline{F} \equiv \mathsf{mmsc}\left((\overline{E} \sqcap \overline{F})^{\mathcal{I}}, \mathcal{I}\right) \in \mathcal{B}(\mathcal{I})$ (again up to logical equivalence). Since $\mathcal{J}$ is a model of $\mathcal{B}(\mathcal{I})$, by Lemma 11:

$$\begin{aligned}\left(\overline{E} \sqcap \overline{F}\right)^{\mathcal{J}} &= \mathsf{mmsc}\left((\overline{E} \sqcap \overline{F})^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{J}} \\ &= \mathsf{mmsc}\left((E \sqcap F)^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{J}} \\ &= \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{J}}.\end{aligned}$$

To prove that $C^{\mathcal{J}} = \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{J}}$, in the following, we write $\overline{C}$ as a shorthand for $\overline{E} \sqcap \overline{F}$ and show that $\overline{C}^{\mathcal{J}} = C^{\mathcal{J}}$. Since $\overline{E} \in \Lambda_{\mathcal{I}}$, we have that $\mathcal{B}(\mathcal{I}) \models \overline{E} \equiv \mathsf{mmsc}\left(\overline{E}^{\mathcal{I}}, \mathcal{I}\right)$. Moreover, as $\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}} = \overline{E}^{\mathcal{I}}$, we have that

$$\mathcal{B}(\mathcal{I}) \models \overline{E} \equiv \mathsf{mmsc}\left(\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}}, \mathcal{I}\right).$$

By Lemma 10, it follows that

$$\emptyset \models \mathsf{mmsc}\left(\mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)^{\mathcal{I}}, \mathcal{I}\right) \equiv \mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right).$$

Therefore, $\mathcal{B}(\mathcal{I}) \models \overline{E} \equiv \mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)$ and as $\mathcal{B}(\mathcal{I}) \models E \equiv \mathsf{mmsc}\left(E^{\mathcal{I}}, \mathcal{I}\right)$, then $\mathcal{B}(\mathcal{I}) \models E \equiv \overline{E}$. Similarly we obtain $\mathcal{B}(\mathcal{I}) \models \overline{F} \equiv F$ and that $\mathcal{B}(\mathcal{I}) \models \overline{C} \equiv C$. As $\mathcal{J}$ was an arbitrarily chosen model of $\mathcal{B}(\mathcal{I})$, we conclude that $\mathcal{B}(\mathcal{I}) \models \overline{C} \equiv \mathsf{mmsc}\left(C^{\mathcal{I}}, \mathcal{I}\right)$ and $\mathcal{B}(\mathcal{I}) \models \overline{C} \equiv C$.

Step case ($\exists$): In this case, $C = \exists r.E$ for some $r \in \mathsf{N_R}$ and $\mathcal{EL}^{\bot}$ concept expression E. Let $\mathcal{J}$ be an interpretation such that $\mathcal{J} \models \mathcal{B}(\mathcal{I})$. We know that:

$$\begin{aligned} x \in C^{\mathcal{J}} &\iff x \in (\exists r.E)^{\mathcal{J}} \\ &\iff \exists y \in E^{\mathcal{J}} : (x,y) \in r^{\mathcal{J}}. \end{aligned}$$

By our induction hypothesis, $\mathcal{B}(\mathcal{I}) \models E \equiv \mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right)$, hence:

$$\begin{aligned} x \in C^{\mathcal{J}} &\iff \exists y \in \mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right)^{\mathcal{J}} : (x,y) \in r^{\mathcal{J}} \\ &\iff x \in (\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}}. \end{aligned}$$

In short, we proved that $C^{\mathcal{J}} = (\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}}$. Next, as $\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right) \in M_{\mathcal{I}}$, we know that

$$\begin{aligned} &\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right) \equiv \\ &\mathsf{mmsc}\left(\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right)^{\mathcal{I}},\mathcal{I}\right) \in \mathcal{B}(\mathcal{I}) \end{aligned}$$

With Lemma 11 we obtain:

$$\begin{aligned} (\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}} &= (\mathsf{mmsc}\left(\exists r.\mathsf{mmsc}\left(E^{\mathcal{I}},\mathcal{I}\right)^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}} \\ &= (\mathsf{mmsc}\left((\exists r.E)^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}} \\ &= (\mathsf{mmsc}\left(C^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}}. \end{aligned}$$

Thus, $C^{\mathcal{J}} = (\mathsf{mmsc}\left(C^{\mathcal{I}},\mathcal{I}\right))^{\mathcal{J}}$. Since $\mathcal{J}$ was chosen arbitrarily, we can conclude that $\mathcal{B}(\mathcal{I}) \models C \equiv \mathsf{mmsc}\left(C^{\mathcal{I}},\mathcal{I}\right)$.

Now, we prove that if $\mathcal{I} \models C \sqsubseteq D$, then $\mathcal{B}(\mathcal{I}) \models \mathsf{mmsc}\left(C^{\mathcal{I}},\mathcal{I}\right) \sqsubseteq \mathsf{mmsc}\left(D^{\mathcal{I}},\mathcal{I}\right)$. Let $\mathcal{J}$ be a model of $\mathcal{B}(\mathcal{I})$. We know from Lemmas 5 and 12 that there are $U, V \subseteq M_{\mathcal{I}}$ such that $C^{\mathcal{I}} = (\bigsqcap U)^{\mathcal{I}}$ and $D^{\mathcal{I}} = (\bigsqcap V)^{\mathcal{I}}$. From the definition of $\mathcal{B}(\mathcal{I})$, we obtain $\mathsf{mmsc}\left((\bigsqcap U)^{\mathcal{I}},\mathcal{I}\right) \sqsubseteq \mathsf{mmsc}\left((\bigsqcap V)^{\mathcal{I}},\mathcal{I}\right) \in \mathcal{B}(\mathcal{I})$. Therefore:

$$\mathcal{J} \models \mathsf{mmsc}\left((\bigsqcap U)^{\mathcal{I}},\mathcal{I}\right) \sqsubseteq \mathsf{mmsc}\left((\bigsqcap V)^{\mathcal{I}},\mathcal{I}\right)$$

Replacing $(\bigsqcap U)^{\mathcal{I}}$ with $C^{\mathcal{I}}$ and $(\bigsqcap V)^{\mathcal{I}}$ with $D^{\mathcal{I}}$ yields:

$$\mathcal{J} \models \mathsf{mmsc}\left(C^{\mathcal{I}},\mathcal{I}\right) \sqsubseteq \mathsf{mmsc}\left(D^{\mathcal{I}},\mathcal{I}\right).$$

Therefore, using the fact that $\mathcal{J} \models C \equiv \mathsf{mmsc}\left(C^{\mathcal{I}},\mathcal{I}\right)$ for every $\mathcal{EL}^{\bot}$ concept expression C (proved earlier) we can conclude that $\mathcal{J} \models C \sqsubseteq D$.

Since all the required concept inclusions hold in an arbitrary model of $\mathcal{B}(\mathcal{I})$, whenever they hold in $\mathcal{I}$ we have that $\mathcal{B}(\mathcal{I})$ is also complete for the $\mathcal{EL}^{\bot}$ CIs. □

Theorem 3. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a finite $\mathcal{EL}^{\bot}$ interpretation and $X \subseteq \Delta^{\mathcal{I}}$. Let $d_{\mathcal{I}}(X)$ be the (adaptable) depth according to Definition 5 and $d_{gfp}(X,\mathcal{I})$ be the depth for the $\mathcal{EL}^{\bot}_{gfp}$ MMSC of X w.r.t. $\mathcal{I}$, according to Lemma 5.5 in (Distel, 2011). Then, $d_{\mathcal{I}}(X) \leq d_{gfp}(X,\mathcal{I})$.*

Proof. Let us assume that the set X has n elements, denoted $x_1, \ldots, x_n$. According to Lemma 4.6 in (Distel, 2011), the MMSC of X w.r.t. $\mathcal{I}$ in $\mathcal{EL}^{\perp}_{gfp}$ is a concept expression $C = (A_X, \mathcal{T}_X)$ where, according to Definition 2.23 in (Distel, 2011), A_X is a fresh concept name and $\mathcal{T}_X$ is a set of equivalences (a TBox) where the left-hand side is a defined concept name (see Definition 2.16 in (Distel, 2011)) and the right-hand side is a concept expression built from the product graph $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$. This TBox has one defined concept name for each vertex in the product graph $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$ that is reachable from $(x_1, \ldots, x_n)$. Therefore, $\mathcal{T}_X$ has $\mathsf{reach}(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n))$ defined terms. According to Lemma 5.5 and equation 5.27 (definition of the base $\mathcal{B}_4$) from (Distel, 2011), the MMSC of X w.r.t. $\mathcal{I}$ has to be unravelled until depth $|\mathsf{N_C}^{def}(\mathcal{T}_X)| \cdot \Delta^{\mathcal{I}} + 1$ when converting the base to $\mathcal{EL}^{\perp}$. Furthermore, $|\mathsf{N_C}^{def}(\mathcal{T}_X)|$ corresponds to the number of reachable vertices from $(x_1, \ldots, x_n)$ in $\prod_{i=1}^{n} \mathcal{G}(\mathcal{I})$.

As mentioned before, the MVF of a vertex in a graph cannot be higher than the number of reachable vertices from the same vertex ($\mathsf{mvf}(\mathcal{G}, v) \leq \mathsf{reach}(\mathcal{G}, v)$). Now, we consider two cases according to Definition 5. If $d_{\mathcal{I}}(X) = \mathsf{mvf}(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n))$ then clearly $d_{\mathcal{I}}(X) \leq d_{gfp}(X, \mathcal{I})$. Otherwise, we have

$$\mathsf{mvf}(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)) \leq \mathsf{reach}(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)) = \mathsf{N_C}^{def}(\mathcal{T}_X).$$

And since and $\mathsf{mmvf}(\mathcal{G}(\mathcal{I})) \leq \Delta^{\mathcal{I}}$, we have

$$\mathsf{mvf}(\prod_{i=1}^{n} \mathcal{G}(\mathcal{I}), (x_1, \ldots, x_n)) \cdot \mathsf{mmvf}(\mathcal{G}(\mathcal{I})) \leq |\mathsf{N_C}^{def}(\mathcal{T}_X)| \cdot \Delta^{\mathcal{I}} + 1.$$

In other words, $d_{\mathcal{I}}(X) \leq d_{gfp}(X, \mathcal{I})$. □

Appendix C. Proofs for Section 6

In the following we present the proofs related to the computation of the MVF function. In particular, we provide proofs for the relationship between the condensation graph and the MVF function (Lemma 6), and the correctness of Algorithm 1 (Lemma 7).

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:*

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

Proof. 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 in $\mathcal{G}$ starting at v with $\mathsf{v_{num}}(\mathbf{w}) = \mathsf{weight}(\mathbf{w}^*)$. Let $v_1 = v$. For each $1 \leq i < m$, the induced walk must: visit v_i, then pass through all vertices in V_i (repeating vertices whenever needed), then visit a vertex $u_i \in V_i$ such that there is an edge $(u_i, r, v_{i+1}) \in E$ with $v_{i+1} \in V_{i+1}$ (this is possible due to the definitions of SCCs and condensation). When the walk reaches a vertex u_{m-1}, it must visit v_m and pass through every vertex in V_m before stopping. Such walk visits every vertex in $\bigcup_{i=1}^{m} V_i$, thus $\mathsf{v_{num}}(\mathbf{w}) = \mathsf{weight}(\mathbf{w}^*)$.

Now let $\mathbf{w}$ be a walk in $\mathcal{G}$ starting at v which is induced (as explained earlier) by a path $\mathbf{w}^*$ in $\mathcal{G}^*$ starting at $\mathsf{scc}(\mathcal{G}, v)$ with maximum weight. Assume that there is a walk $\overline{\mathbf{w}}$ in $\mathcal{G}$ starting at v such that $\mathsf{v_{num}}(\overline{\mathbf{w}}) > \mathsf{v_{num}}(\mathbf{w})$. Due to the definitions of SCC and condensation we know that there is a path $\overline{\mathbf{w}}^*$ in $\mathcal{G}^*$ starting at $\mathsf{scc}(\mathcal{G}, v)$ such that $\mathsf{v_{num}}\overline{\mathbf{w}} \leq \mathsf{weight}(\overline{\mathbf{w}}^*)$. However, this would imply that: $\mathsf{weight}(\mathbf{w}^*) = \mathsf{v_{num}}(\mathbf{w}) < \mathsf{v_{num}}\overline{\mathbf{w}} \leq \mathsf{weight}(\overline{\mathbf{w}}^*)$, which is a contradiction since we assume that $\mathbf{w}^*$ has maximal weight. Therefore, no walk in $\mathcal{G}$ starting at v can visit more vertices than $\mathsf{weight}(\mathbf{w}^*)$.

Since we have shown that for every path $\mathbf{w}^*$ in $\mathcal{G}^*$ starting at $\mathsf{scc}(\mathcal{G}, v)$, there is a walk $\mathbf{w}$ in $\mathcal{G}$ starting at v, with $\mathsf{v_{num}}(\mathbf{w}) = \mathsf{weight}(\mathbf{w}^*)$, we can conclude that the statement of this lemma holds. □

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. Let $\mathcal{G}^* = (V^*, E^*)$ be the condensation of $\mathcal{G}$. If $W' \in V^*$ is unreachable from $\mathsf{scc}(\mathcal{G}, v)$ then $wgt[V']$ will remain null as it will never be visited. Otherwise, W' will be visited in some call of $\mathsf{maxWeight}$. If it has no successors, the loop in Line 9 will not do anything, and thus $wgt[W'] = |W'|$ as expected. Instead, if $\mathsf{scc}(\mathcal{G}, v)$ has successors, then the maximum weight of a path starting 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. Since, the loop in Line 9 forces the maximum weights of the successors of W' to be calculated first, the value returned in Line 15 is correct. □

Appendix D. Compacting the Product Graph

The product graph employed in the calculation of the MMSC can have an exponential number of vertices in the size of the domain, more specifically, for a finite interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ the product graph can have up to $|\Delta^{\mathcal{I}}|^{|\Delta^{\mathcal{I}}|}$ vertices. However, some of these vertices are indistinguishable when navigating the product graph, and thus, we could group these vertices, producing an equivalent, but smaller graph. The idea is to group vertices into classes of equivalence as defined next.

Definition 12. *Let $\mathcal{G} = (V, E, L)$ be a description graph. We say that two vertices $v, w \in V$ are* similar *(written as $v \approx w$) if:*

1. *$L(v) = L(w)$.*
2. *$(v, r, v') \in E$ iff $\exists w' \in V$ such that $w' \approx v'$ and $(w, r, w') \in E$.*

The similarity relation between vertices induces a partitioning of V into equivalence classes. Denote by $[v]$ the equivalence class which contains the vertex v. The set of all equivalence classes of vertices is $[V]$, that is, $[V] = \{[v] \mid v \in V\}$.

Definition 13. *Let $\mathcal{G} = (V, E, L)$ be a description graph.*

The compact version *of $\mathcal{G}$, in symbols, $[\mathcal{G}] = ([V], [E], [L])$ is defined as follows:*

- *$[V] = \{[v] \mid v \in V\}$.*
- *$[E] = \{([v], r, [w]) \mid (v, r, w) \in E\}$.*
- *$[L]([v]) = L(v)$.*

The function $L(v)$ is well defined because all vertices in the same class have the same labels.

Lemma 13 shows that there will be always a simulation from the compact version of a description graph to the original one.

Lemma 13. *Let $\mathcal{G} = (V, E, L)$ be a description graph and $[\mathcal{G}] = ([V], [E], [L])$ its compact version. Now, let the relation Z_{cp} be defined as follows:*

$$Z_{cp} = \{([w], w) \mid w \in V\}.$$

For every $v \in V$, Z_{cp} is a simulation from $([\mathcal{G}], [v])$ to $(\mathcal{G}, v)$.

Proof. Let $\mathcal{G}$, $[\mathcal{G}]$, and Z_{cp} be as in the lemma's statement. Consider now a vertex $v \in V$.

1. By definition of Z_{cp}, $([v], v) \in Z_{cp}$.

2. By the first condition in Definition 13, for all $w \in V$ we have that:
$$[L]([w]) = L([w]) = L(w).$$

3. Let $([w], w) \in Z_{pc}$ and assume that $([w], r, [x]) \in E$. By Definition 13, $([w], r, [x]) \in E$ implies that there are $w' \in [w]$ and $x' \in [x]$ such that $(w', r, x') \in E$. By the second condition in Definition 12, since $w' \approx w$ and $(w', r, x') \in E$ there is some $x' \approx y \in V$ such that $(w, r, y) \in E$. Moreover, by construction of Z_{cp} and Definition 13 we obtain $([x], y) \in E$, as required.

Therefore, Z_{cp} is a simulation from $([\mathcal{G}], [v])$ to $(\mathcal{G}, v)$ for any $v \in V$. □

Lemma 14. *Let $\mathcal{G} = (V, E, L)$ be a description graph and $[\mathcal{G}] = ([V], [E], [L])$ its compact version. Now, let the relation Z_{pc} be defined as follows:*

$$Z_{pc} = \{(w, [w]) \mid w \in V\}.$$

For every $v \in V$, Z_{pc} is a simulation from $(\mathcal{G}, v)$ to $([\mathcal{G}], [v])$.

Proof. Let $\mathcal{G}$, $[\mathcal{G}]$, and Z_{pc} be as in the lemma's statement. Consider now a vertex $v \in V$.

1. By definition of Z_{pc}, $(v, [v]) \in Z_{pc}$.

2. As stated in Definition 13, for all $w \in V$ we have that:
$$[L]([w]) = L([w]) = L(w).$$

3. Let $(w, [w]) \in Z_{pc}$ and assume that $(w, r, x) \in E$. By Definition 13, we know that $([w], r, [x]) \in [E]$ and by construction of Z_{pc}, we have $(x, [x]) \in Z_{pc}$.

Therefore, Z_{pc} is a simulation from $(\mathcal{G}, v)$ to $([\mathcal{G}], [v])$ for any $v \in V$. □

Since our results depend on unravellings, we also need to consider simulations between unravellings of two description graphs, as follows.

Lemma 15. *Let $\mathcal{G} = (V, E, L)$ and $\mathcal{G}' = (V', E', L')$ be description graphs, $v \in V$ and $v' \in V'$. If there is a simulation $Z : (\mathcal{G}, v) \mapsto (\mathcal{G}', v')$, then for all $n \in \mathbb{N}$ there is a simulation $Z_n : (\mathcal{G}^v_n, v) \mapsto (\mathcal{G}'^{v'}_n, v')$.*

Proof. The proof is by induction on n.

Base: Assume $n = 0$. We will prove that the set $Z_0 = \{(v, \hat{v}) \mid (v, \hat{v}) \in Z\}$ is a simulation from $(\mathcal{G}_0^v, v)$ to $({\mathcal{G}'}_0^{v'}, v')$:

1. $(v, v') \in Z_0$ since we assume $(v, v') \in Z$.
2. $L(v) \subseteq L(\hat{v})$ for all $(v, v') \in Z_0$ because Z is a simulation.
3. $\mathcal{G}_0^v$ does not have edges, hence the last condition is trivially satisfied.

Therefore, Z_0 is a simulation from $(\mathcal{G}_0^v, v)$ to $({\mathcal{G}'}_0^{v'}, v')$.

Step: Assume that the lemma holds for all $0 \leq i < n$. Let

$$Z_n = Z_{n-1} \cup \{(\hat{\mathbf{w}}rx, \hat{\mathbf{w}}'ry) \mid (\hat{\mathbf{w}}, \hat{\mathbf{w}}') \in Z_{n-1}, (\delta(\hat{\mathbf{w}}), r, x) \in E, \text{ and } (x, y) \in Z\}.$$

We will prove that the set Z_n as is a simulation from $(\mathcal{G}_n^v, v)$ to $({\mathcal{G}'}_n^{v'}, v')$:

1. By construction, $Z_{n-1} \subseteq Z_n$. Since Z_{n-1} is a simulation from $(\mathcal{G}_{n-1}^v, v)$ to $(\mathcal{G}_{n-1}^v, v)$ (by the induction hypothesis), we have that $(v, v') \in Z_n$.
2. Let $(\mathbf{w}, \mathbf{w}') \in Z_n$. If $(\mathbf{w}, \mathbf{w}') \in Z_{n-1}$, we get as a consequence of the induction hypothesis that $L_n(\mathbf{w}) = L_{n-1}(\mathbf{w}) \subseteq L'_{n-1}(\mathbf{w}') = L'_n(\mathbf{w}')$. Otherwise, since we assume that Z is a simulation from $(\mathcal{G}, v)$ to $(\mathcal{G}', v')$, we get that $L_n(\delta(\mathbf{w})) = L(\delta(\mathbf{w})) \subseteq L'(\delta(\mathbf{w}')) = L'_n(\mathbf{w}')$.
3. Let $(\mathbf{w}, \mathbf{w}') \in Z_n$ and $(\mathbf{w}, s, \mathbf{w}st) \in E_n$. By the definition of unravelling we only need to consider the cases in which $\mathbf{w} \in V_{n-1}$. If $(\mathbf{w}, s, \mathbf{w}st) \in E_{n-1}$, we get as a consequence of the induction hypothesis, that there is a $\mathbf{w}'' \in V'_{n-1} \subseteq V'_n$ such that $(\mathbf{w}st, \mathbf{w}'') \in Z_{n-1}$ and $(\mathbf{w}', s, \mathbf{w}'') \in E'_{n-1} \subseteq E'_n$. Otherwise, we know that $|\mathbf{w}| = n - 1$ and thus $\mathbf{w}st \in V_n \setminus V_{n-1}$. By construction, we get $(\mathbf{w}, \mathbf{w}') \in Z_{n-1}$. Since $(\mathbf{w}, s, \mathbf{w}st) \in E_n$, there is an edge $(\delta(\mathbf{w}), s, t) \in E$. Also, since Z is a simulation and $(\delta(\mathbf{w}), \delta(\mathbf{w}')) \in Z$ due to the constructions of Z_0 and Z_n, we get that there must be some $u \in V'$ such that $(\delta(\mathbf{w}'), s, u) \in E'$ and $(t, u) \in Z$. Therefore, $(\mathbf{w}', s, \mathbf{w}su) \in E'_n$, and by definition of Z_n, get that $(\mathbf{w}st, \mathbf{w}'su) \in Z_n$.

Therefore, the lemma holds for all $n \in \mathbb{N}$. □

Finally, we prove that we can replace the product graph in Definition 5 with its compact version, potentially having a lower depth while ensuring that Lemma 4 still holds.

Lemma 16. *Let $\mathcal{G} = (V, E, L)$ and $\mathcal{G}' = (V', E', L')$ be description graphs, $v \in V$, $v' \in V'$, $n \in \mathbb{N}^*$. Also, let*

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

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

Proof. Let us assume that there is a simulation $Z'_d : (\mathcal{G}^v_{d'}, v) \mapsto (\mathcal{G}', v')$. From Lemma 13, we know that there is a simulation from $([\mathcal{G}], [v])$ to $(\mathcal{G}, v)$. Hence, we get from Lemma 15 that there is a simulation from $([\mathcal{G}]^{[v]}_{d'}, [v])$ to $(\mathcal{G}^v_{d'}, v)$. Since the composition of simulations is a simulation, we know that there is a simulation $Z^c_{d'} : ([\mathcal{G}]^{[v]}_{d'}, [v]) \mapsto (\mathcal{G}', v')$.

The existence of such $Z^c_{d'}$ together with Lemma 2 ensures that there is a simulation $Z'_k : ([\mathcal{G}]^{[v]}_k, [v]) \mapsto (\mathcal{G}', v')$. Lemma 14 states that there is a simulation $Z_{pc} : (\mathcal{G}, v) \mapsto ([\mathcal{G}], [v])$, and as a consequence of Lemma 15, we know that there is a simulation $Z_2 : (\mathcal{G}^v_k, v) \mapsto ([\mathcal{G}]^{[v]}_k, [v])$. Then, we can take $Z_k = Z_2 \circ Z'_k$, which is a simulation from $(\mathcal{G}^v_k, v)$ to $([\mathcal{G}]^{[v]}_k, [v])$.

Hence, if there is a simulation $Z'_d : (\mathcal{G}^v_{d'}, v) \mapsto (\mathcal{G}', v')$, then there is a simulation $Z_k : (\mathcal{G}^v_k, v) \mapsto (\mathcal{G}', v')$ for all $k \in \mathbb{N}$. □

The compact version of a description graph will have at most as many vertices as the original one since there can be at most $|V|$ equivalence classes. In our a case, since we use product graphs whose factors are all the same graph, we can give an upper bound on the number of vertices of its compact version. If $\mathcal{G} = (V, E, L)$ is a description graph, $\prod_{i=1}^n \mathcal{G}$ can have up to $|V|^n$ vertices, but the compact version of this product can only have up to: $\frac{(|V|+n-1)!}{n!(|V|-1)!}$. Before we prove this result, we will need the following lemma, which establishes a sufficient condition for two vertices to belong to the same equivalent class in $\prod_{i=1}^n \mathcal{G}$.

Lemma 17. *Let $\mathcal{G} = (V, E, L)$ be a description graph, $\mathcal{G}_P = (V_P, E_P, L_P) = \prod_{i=1}^n \mathcal{G}$, and $n \in \mathbb{N}^*$. Also, let $v = (v_1, \ldots, v_n)$ and $w = (w_1, \ldots, w_n)$ be vertices in V_P. If v and w are permutations of each other, that is, $|\{v_i = x \mid 1 \leq i \leq n\}| = |\{w_i = x \mid 1 \leq i \leq n\}|$ for all $x \in V$, then $[v] = [w]$.*

Proof. We will show that $[v] = [w]$ by proving each required condition separately. We just need to show that defining classes of equivalence via the notion of permutation satisfies the same conditions specified before.

1. By the definition of product graph, we have that $L_P(v) = \bigcap_{i=1}^n L(v_i)$. Since, v and w are permutations of each other $\bigcap_{i=1}^n L(v_i) = \bigcap_{i=1}^n L(w_i)$. Hence $L_P(v) = L_P(w)$.

2. Let $(v, r, v') \in E_P$. By the definition of product graph, we have that $(v_1, r, v'_1) \in E$ for all $1 \leq i \leq n$. As v and w are permutations of each other, there is a bijective function $\mathrm{p} : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $w_j = v_{\mathrm{p}(j)}$ for all $1 \leq j \leq n$. Now, consider the vertex $w' = (v'_{\mathrm{p}(1)}, \ldots, v'_{\mathrm{p}(n)})$. We know that v' and w' are permutations of each other. Moreover, we know that $(w_j, r, w'_j) \in E$ for all $1 \leq j \leq n$ because $(v_{\mathrm{p}(j)}, r, v'_{\mathrm{p}(j)}) \in E$ as a consequence of $(v, r, v') \in E_P$.

3. This case is analogous to item (2), and we can conclude that if $(v', r, v) \in E$, then there is a permutation w' such that $(w, r, w') \in E_P$.

Hence, using permutation groups as classes, preserves the properties of equivalence classes stated before. That is, we can conclude that $[v] = [w]$ whenever v and w are permutations of each other. □

Lemma 18 shows how to obtain the value we claimed before.

Lemma 18. *Let $\mathcal{G} = (V, E, L)$ be a description graph, $\mathcal{G}_P = (V_P, E_P, L_P) = \prod_{i=1}^{n} \mathcal{G}$ and $n \in \mathbb{N}^*$. Then $[\mathcal{G}_P]$ has at most $\frac{(|V|+n-1)!}{n!(|V|-1)!}$ vertices.*

Proof. Let $\mathcal{G}$, $\mathcal{G}_P$ and n be as stated. Let $(v_1, \ldots, v_n)$ and $w_1, \ldots, w_n$ be vertices in V_P.

Lemma 17 shows that we grouping vertices that are permutations of each other gives an upper limit to the number of equivalence classes, and thus, of the number of vertices in $[\mathcal{G}_P]$.

This means that if we count all possible combinations of n elements drawn from V with repetition, we obtain an upper bound for the number of equivalence classes in $[V_P]$. The number of combinations of $|V|$ elements taken n at a time with repetition is given by the following formula (Benjamin & Quinn, 2003)

$$\frac{(|V|+n-1)!}{n!(|V|-1)!}.$$

Hence, there can be at most this many vertices in $[V]$. □

Finally, Lemma 19 proves that the compact version has at most the same quantity of vertices as the original product graph.

Lemma 19. *Let $\mathcal{G} = (V, E, L)$ be a description graph, $n \in \mathbb{N}^*$, $\mathcal{G}_P = (V_P, E_P, L_P) = \prod_{i=1}^{n} \mathcal{G}$ and $[\mathcal{G}_P] = (V_C, E_C, L_C)$. Then $[\mathcal{G}_P]$ has at most as many vertices as $\mathcal{G}_P$.*

Proof. First, let $m, n \in \mathbb{N}^*$, $f(m, n) = \frac{(m+n-1)!}{n!(m-1)!}$ and $g(m, n) = m^n$. We will prove that $f(m, n) \leq g(m, n)$.

We can write $f(m, n+1)$ in terms of $f(m, n)$ as:

$$f(m, n+1) = \frac{m+n}{n+1} \cdot f(m, n).$$

We can do the analogous for $g(m, n)$:

$$g(m, n+1) = m \cdot g(m, n).$$

When $n = 1$ we have:

$$f(m, 1) = m = g(m, n).$$

Therefore, to prove that $f(m, n) \leq g(m, n)$ for all $m, n \in \mathbb{N}^*$ it is sufficient to show that $\frac{m+n}{n+1} \leq m$ for all $m, n \in \mathbb{N}^*$, which we do next:

$$\begin{aligned} \frac{m+n}{n+1} &\leq m \\ m+n &\leq mn+m \\ m-m-mn+n &\leq 0 \\ -mn+n &\leq 0 \\ n(1-m) &\leq 0 \end{aligned}$$

Since $n \leq 1$, $n(1-m) \leq 0$ iff $1 - m \leq 0$, that is, $m \geq 1$, which is already assumed. Hence, we have shown that $f(m,n) \leq g(m,n)$ for $m, n \in \mathbb{N}^*$.

If we replace m with $|V|$, $g(m,n)$ is the number of vertices in $\mathcal{G}_P$ while $f(m,n)$ gives the number of vertices in $[\mathcal{G}_P]$ according to Lemma 18. Hence, we proved that $[\mathcal{G}_P]$ will have at most as many vertices as $\mathcal{G}_P$. □

References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 207–216.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. G. (2007). Dbpedia: A nucleus for a web of open data. In Aberer, K., Choi, K., Noy, N. F., Allemang, D., Lee, K., Nixon, L. J. B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., & Cudré-Mauroux, P. (Eds.), *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, Vol. 4825 of *Lecture Notes in Computer Science*, pp. 722–735. Springer.

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*, pp. 168–178.

Baader, F. (2003). Terminological cycles in a description logic with existential restrictions. In *IJCAI*, pp. 325–330. Morgan Kaufmann.

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the $\mathcal{EL}$ envelope. In Kaelbling, L. P., & Saffiotti, A. (Eds.), *IJCAI*, pp. 364–369. Professional Book Center.

Baader, F., & Distel, F. (2008). A finite basis for the set of $\mathcal{EL}$-implications holding in a finite model. In Medina, R., & Obiedkov, S. (Eds.), *ICFCA 2008*, pp. 46–61. Springer-Verlag.

Baader, F., & Distel, F. (2009). Exploring finite models in the description logic $\mathcal{EL}_{gfp}$. In Ferré, S., & Rudolph, S. (Eds.), *Proceedings of the 7th International Conference on Formal Concept Analysis, (ICFCA 2009)*, pp. 146–161. Springer-Verlag.

Baader, F., Ganter, B., Sertkaya, B., & Sattler, U. (2007). Completing description logic knowledge bases using formal concept analysis. In Veloso, M. M. (Ed.), *IJCAI*, pp. 230–235. AAAI Press.

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

Baader, F., & Molitor, R. (2000). Building and structuring description logic knowledge bases using least common subsumers and concept analysis. In *ICCS*, pp. 292–305. Springer.

Babin, M. A., & Kuznetsov, S. O. (2013). Computing premises of a minimal cover of functional dependencies is intractable. *Discrete Applied Mathematics*, *161*(6), 742–749.

Benjamin, A. T., & Quinn, J. J. (2003). *Proofs That Really Count: The Art of Combinatorial Proof*. MATHEMATICAL ASSN OF AMER.

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

Borchmann, D. (2013a). Axiomatizing $\mathcal{EL}^\perp$-expressible terminological knowledge from erroneous data. In Benjamins, V. R., d'Aquin, M., & Gordon, A. (Eds.), *Proceedings of the 7th International Conference on Knowledge Capture, K-CAP 2013, Banff, Canada, June 23-26, 2013*, pp. 1–8. ACM.

Borchmann, D. (2013b). Towards an error-tolerant construction of $\mathcal{EL}^\perp$ -ontologies from data using formal concept analysis. In Cellier, P., Distel, F., & Ganter, B. (Eds.), *ICFCA*, Vol. 7880 of *Lecture Notes in Computer Science*, pp. 60–75. Springer.

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

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

Dau, F., & Sertkaya, B. (2011). Formal concept analysis for qualitative data analysis over triple stores. In Troyer, O. D., Medeiros, C. B., Billen, R., Hallot, P., Simitsis, A., & Mingroot, H. V. (Eds.), *Advances in Conceptual Modeling. Recent Developments and New Directions - ER 2011 Workshops FP-UML, MoRE-BI, Onto-CoM, SeCoGIS, Variability@ER, WISM, Brussels, Belgium, October 31 - November 3, 2011. Proceedings*, Vol. 6999 of *Lecture Notes in Computer Science*, pp. 45–54. Springer.

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

Distel, F., & Sertkaya, B. (2011). On the complexity of enumerating pseudo-intents. *Discrete Appliead Mathematics*, *159*(6), 450–466.

Eiter, T., & Gottlob, G. (1995). Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, *24*(6), 1278–1304.

Fanizzi, N., d'Amato, C., & Esposito, F. (2008). DL-FOIL concept learning in description logics. In Zelezný, F., & Lavrac, N. (Eds.), *Proceedings of Inductive Logic Programming, 18th International Conference, ILP*, Vol. 5194 of *Lecture Notes in Computer Science*, pp. 107–121. Springer.

Funk, M., Jung, J. C., Lutz, C., Pulcini, H., & Wolter, F. (2019). Learning description logic concepts: When can positive and negative examples be separated?. In Kraus, S. (Ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 1682–1688. ijcai.org.

Ganter, B. (1984). Two basic algorithms in concept analysis. Tech. rep. Preprint-Nr. 831, Technische Hochschule Darmstadt, Darmstadt, Germany.

Ganter, B. (2010). Two basic algorithms in concept analysis. In Kwuida, L., & Sertkaya, B. (Eds.), *Proceedings of the 8th International Conference on Formal Concept Analysis, (ICFCA 2010)*, Vol. 5986 of *Lecture Notes in Artificial Intelligence*, pp. 329–359. Springer-Verlag. Reprint of (Ganter, 1984).

Ganter, B., & Wille, R. (1999). *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin/Heidelberg.

Guigues, J.-L., & Duquenne, V. (1986). Familles minimales d'implications informatives resultant d'un tableau de données binaries. *Mathématiques, Informatique et Sciences Humaines*, *95*, 5–18.

Guimarães, R., Ozaki, A., Persia, C., & Sertkaya, B. (2021). Mining EL bases with adaptable role depth. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 6367–6374. AAAI Press.

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

Horridge, M. (2011). *Justification based explanation in ontologies*. Ph.D. thesis, University of Manchester, UK.

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: the making of a web ontology language. *Journal of Web Semantics*, *1*(1), 7–26.

Iannone, L., Palmisano, I., & Fanizzi, N. (2007). An algorithm based on counterfactuals for concept learning in the semantic web. *Appl. Intell.*, *26*(2), 139–159.

Johnson, D. S., Yannakakis, M., & Papadimitriou, C. H. (1988). On generating all maximal independent sets. *Information Processing Letters*, *27*(3), 119–123.

Kautz, H., Kearns, M., & Selman, B. (1995). Horn approximations of empirical data. *Artificial Intelligence*, *74*, 129–145.

Klarman, S., & Britz, K. (2015). Ontology learning from interpretations in lightweight description logics. In Inoue, K., Ohwada, H., & Yamamoto, A. (Eds.), *Inductive Logic Programming - 25th International Conference, ILP 2015, Kyoto, Japan, August 20-22, 2015, Revised Selected Papers*, Vol. 9575 of *Lecture Notes in Computer Science*, pp. 76–90. Springer.

Konev, B., Lutz, C., Ozaki, A., & Wolter, F. (2017). Exact learning of lightweight description logic ontologies. *J. Mach. Learn. Res.*, *18*, 201:1–201:63.

Kriegel, F. (2016). Axiomatization of general concept inclusions from streams of interpretations with optional error tolerance. In Kuznetsov, S. O., Napoli, A., & Rudolph, S. (Eds.), *Proceedings of the 5th International Workshop "What can FCA do for Artificial Intelligence"? co-located with the European Conference on Artificial Intelligence, FCA4AI@ECAI 2016, The Hague, the Netherlands, August 30, 2016*, Vol. 1703 of *CEUR Workshop Proceedings*, pp. 9–16. CEUR-WS.org.

Kriegel, F. (2017). Probabilistic implication bases in FCA and probabilistic bases of GCIs in $\mathcal{EL}^{\bot}$. *Int. J. Gen. Syst.*, *46*(5), 511–546.

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., & Manna, M. (Eds.), *JELIA 2019*, pp. 399–417. Springer.

Kriegel, F. (2020a). Constructing and extending description logic ontologies using methods of formal concept analysis. *Künstliche Intell.*, *34*(3), 399–403.

Kriegel, F. (2020b). Most specific consequences in the description logic $\mathcal{EL}$. *Discret. Appl. Math.*, *273*, 172–204.

Kuznetsov, S. O. (2004a). On the intractability of computing the Duquenne-Guigues Base. *Journal of Universal Computer Science*, *10*(8), 927–933.

Kuznetsov, S. O. (2004b). On the intractability of computing the duquenne-guigues base. *J. UCS*, *10*(8), 927–933.

Lehmann, J. (2009). Dl-learner: Learning concepts in description logics. *J. Mach. Learn. Res.*, *10*, 2639–2642.

Lehmann, J. (2010). *Learning OWL Class Expressions*, Vol. 6 of *Studies on the Semantic Web*. IOS Press.

Lehmann, J., & Hitzler, P. (2010). Concept learning in description logics using refinement operators. *Mach. Learn.*, *78*(1-2), 203–250.

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

Lisi, F. A. (2011). Al-quin: An onto-relational learning system for semantic web mining. *Int. J. Semantic Web Inf. Syst.*, *7*(3), 1–22.

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

Monnin, P., Lezoche, M., Napoli, A., & Coulet, A. (2017). Using formal concept analysis for checking the structure of an ontology in LOD: the example of dbpedia. In *ISMIS*, pp. 674–683. Springer.

Obiedkov, S. A., & Duquenne, V. (2007). Attribute-incremental construction of the canonical implication basis. *Annals of Mathematics and Artificial Intelligence*, *49*(1-4), 77–99.

Ozaki, A. (2020). Learning description logic ontologies: Five approaches. where do they stand?. *Künstliche Intelligenz*, *34*(3), 317–327.

Peñaloza, R. (2009). *Axiom pinpointing in description logics and beyond*. Ph.D. thesis, Dresden University of Technology.

Rudolph, S. (2004). Exploring relational structures via $\mathcal{FLE}$. In Wolff, K. E., Pfeiffer, H. D., & Delugach, H. S. (Eds.), *ICCS*, pp. 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. (2009a). Some computational problems related to pseudo-intents. In Ferré, S., & Rudolph, S. (Eds.), *Proceedings of the 7th International Conference on Formal Concept Analysis, (ICFCA 2009)*, Vol. 5548 of *Lecture Notes in Artificial Intelligence*, pp. 130–145. Springer-Verlag.

Sertkaya, B. (2009b). Towards the complexity of recognizing pseudo-intents. In Dau, F., & Rudolph, S. (Eds.), *Proceedings of the 17th International Conference on Conceptual Structures, (ICCS 2009)*, Lecture Notes in Computer Science. Springer-Verlag.

Sertkaya, B. (2010). A survey on how description logic ontologies benefit from formal concept analysis. In Kryszkiewicz, M., & Obiedkov, S. (Eds.), *CLA*, Vol. 672 of *CEUR Workshop Proceedings*, pp. 2–21.

Spackman, K. A., Campbell, K. E., & Côté, R. A. (1997). SNOMED RT: a reference terminology for health care. In *AMIA 1997, American Medical Informatics Association Annual Symposium, Nashville, TN, USA, October 25-29, 1997*. AMIA.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, *1*(2), 146–160.