



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

**Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory**

LTCS–Report

Efficient Axiomatization of OWL 2 EL Ontologies from Data by means of Formal Concept Analysis (Extended Version)

Francesco Kriegel

LTCS-Report 23-01

This is an extended version of an article accepted at AAAI 2024.

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

<http://lat.inf.tu-dresden.de>

Visiting Address:
Nöthnitzer Str. 46
Dresden

Efficient Axiomatization of OWL 2 EL Ontologies from Data by means of Formal Concept Analysis (Extended Version)

Francesco Kriegel

Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany
francesco.kriegel@tu-dresden.de

Abstract

We present an FCA-based axiomatization method that produces a complete \mathcal{EL} TBox (the terminological part of an OWL 2 EL ontology) from a graph dataset in at most exponential time. We describe technical details that allow for efficient implementation as well as variations that dispense with the computation of extremely large axioms, thereby rendering the approach applicable albeit some completeness is lost. Moreover, we evaluate the prototype on real-world datasets.

1 Introduction

Description Logics (DLs) (Baader, Horrocks, Lutz, Sattler, 2017) are formal languages used in knowledge-based systems that reason and make inferences about complex domains, particularly where precision and explainability are essential. By representing knowledge as ontologies built with DLs, these systems can perform automated reasoning to answer queries and thereby assist in making decisions based on the encoded knowledge. DLs are fundamental to the Semantic Web (Hitzler, Krötzsch, Rudolph, 2010) [1], a vision of the World Wide Web where information is represented in a machine-readable format. They provide the logical underpinning for the Web Ontology Language (OWL) [2], which is widely used in the Semantic Web to enable better interoperability across different applications, domains, and natural languages.

In e-commerce, DL ontologies can be used to categorize products into different classes and sub-classes based on their attributes, features, and properties. This enables efficient search and navigation for users on e-commerce platforms, such as eBay and Alibaba (Shi, J. Chen, Dong, Khan, Liang, Zhou, Wu, Horrocks, 2023). In finance, where accuracy and explicability are crucial, the DL formalism enables clear and unambiguous representation of financial concepts, such as assets, liabilities, investments, and transactions. Two examples are the Financial Regulation Ontology (FRO) [3], and the Financial Industry Business Ontology (FIBO) [4]. Another example is the Dow Jones Knowledge Graph (Horrocks, Olivares, Cocchi, Motik, Roy, 2022), which does not use DLs but similar Semantic Web technologies such as SHACL (Bogaerts, Jakubowski, Van den Bussche, 2022).

Moreover, DLs and ontologies have found extensive applications in healthcare and life sciences. The Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT,

or SCT) [5] is an ontology that represents medical terms used in electronic health records, such as clinical findings, symptoms, and diagnoses. It is employed in clinical decision support systems to assist healthcare professionals in making accurate diagnoses, suggesting appropriate treatments, and predicting outcomes based on patient-specific information. The Gene Ontology (GO) [6], the world’s largest source of information on the functions of genes, is a foundation for computational analysis of large-scale molecular biology and genetics experiments in biomedical research.

Among the different DLs, the \mathcal{EL} family (Baader, Brandt, Lutz, 2005, 2008) stands out as a lightweight option. \mathcal{EL} is designed to strike a balance between expressivity and computational complexity, making it an ideal choice for applications where scalability and latency are crucial. It offers a more restricted set of constructs compared to other DLs, but can thus handle large-scale ontologies efficiently. The Web Ontology Language includes it as the profile OWL 2 EL [7].

Every DL ontology is subdivided into two parts. The ABox consists of factual statements about specific individuals or objects in the domain, such as assignment of individuals to concepts and linkage between individuals by roles. The TBox defines concepts and their hierarchy, roles and their characteristics, and constraints or rules that govern all individuals in the domain. By separating ABox and TBox, a DL ontology provides a clear distinction between instance-level and schema-level knowledge. This separation enables reusing the same TBox across different ABoxes, which in turn promotes scalability and maintainability, as changes in the TBox are propagated to all associated ABoxes.

SCT and GO are formulated in \mathcal{EL} . For instance, SCT contains the TBox statement

Common_cold

$$\begin{aligned} \sqsubseteq & \text{Disease} \sqcap \exists \text{causative_agent.Virus} \\ & \sqcap \exists \text{finding_site.Upper_respiratory_tract_structure} \\ & \sqcap \exists \text{pathological_process.Infectious_process} \end{aligned}$$

which is a concept inclusion (CI) and expresses that a common cold (the premise) is a disease that has as pathological process an infectious process, is caused by a virus, and can be found in the upper respiratory tract (the conclusion). We could express that Alice is diagnosed with common cold by the ABox statement $\text{alice} : \exists \text{has_diagnose.Common_cold}$.

Building and maintaining DL ontologies is a laborious task, especially for large domains. Knowledge engineers

and domain experts work together to transfer their knowledge into an ontology. While the ABox is usually filled with observed data, constructing the TBox is a more complex endeavour. Assistance by automated approaches or guidance by interactive approaches is often valuable. For instance, a selection of individuals in the data can be described by a single concept (Funk, Jung, Lutz, 2022; Funk, Jung, Lutz, Pulcini, Wolter, 2019; Zarri , Turhan, 2013) that the experts integrate into the conceptual hierarchy of the TBox. They can also model the schema of an ontology as a diagram (similar to UML class diagrams) that is then automatically translated into a TBox (Sarker, Krisnadhi, Hitzler, 2016). New ontologies can be constructed from existing ones as well. Two or more ontologies can be integrated by ontology alignment (J. Chen, Jim nez-Ruiz, Horrocks, X. Chen, Myklebust, 2023; Jimeno-Yepes, Jim nez-Ruiz, Llavori, Rebholz-Schuhmann, 2009). Conversely, a part of an ontology representing a sub-domain can be extracted by modularization (Cuenca Grau, Horrocks, Kazakov, Sattler, 2008), uniform interpolation or forgetting (Lutz, Wolter, 2011; Zhao, Schmidt, Wang, Zhang, Feng, 2020), or other techniques (Alghamdi, Schmidt, Del-Pinto, Gao, 2021).

Formal Concept Analysis (FCA) (Ganter, Wille, 1999) is a mathematical theory that represents data as formal contexts in which objects are described by their attributes. These attributes are similar to atomic statements in propositional logic and unary predicates in first-order logic. FCA has two main applications: the concept lattice reveals the conceptual hierarchy in the data (Wille, 1982), and the canonical implication base is a complete set of implications, i.e. it entails all implications valid in the data (Guigues, Duquenne, 1986; Stumme, 1996). No complete set with fewer implications exists (Distel, 2011; Wild, 1994). If the data is not explicitly available but is only known by an expert, attribute exploration (Ganter, 1984) enables interactive construction of the implication base.

The data analysis capabilities of FCA have been successfully employed in DLs, especially for the construction and extension of DL ontologies. Given a finite set of concepts, the hierarchy of all their conjunctions can efficiently be computed (Baader, 1995). In order to support a bottom-up construction of DL ontologies, one can first compute most specific concepts for all individuals and then efficiently build the hierarchy of their least common subsumers (Baader, Molitor, 2000; Baader, Sertkaya, 2004). An ontology should be extended when it is incomplete since missing statements have been identified that should be entailed. Interactively completing the ontology using FCA is possible when attention is restricted to CIs over conjunctions from a fixed set of concepts (Baader, Ganter, Sertkaya, Sattler, 2007). Moreover, it can be extended with new statements that are guessed by machine-learning approaches based on knowledge-graph embeddings (Jackermeier, J. Chen, Horrocks, 2023; Shi, J. Chen, Dong, Khan, Liang, Zhou, Wu, Horrocks, 2023). However, some of these embeddings fail to capture the semantics (Jain, Kalo, Balke, Krestel, 2021) and, in effect, a large amount of useless, false predictions might be generated. This major issue can possibly be remedied by novel embedding approaches (Abboud, Ceylan, Lukasiewicz, Sal-

vatori, 2020; Asaadi, Giesbrecht, Rudolph, 2023).

Axiomatization is another approach to constructing ontologies. In general, axiomatization is the task of describing a dataset (or any other formal object) by means of logical statements or axioms, viz. such that a logical formula (in the underlying logic) holds in the data iff. that formula is entailed by these axioms. In addition, axiomatization enables data analysis by transferring the given data into meaningful logical statements. By a suitable choice of the logical formalism, interesting and condensed insights into the analyzed data can be obtained.

In FCA, the canonical implication base axiomatizes data in form of a formal context by means of implications in propositional logic. By exploiting the similarity between \mathcal{EL} CIs and FCA implications, a complete \mathcal{EL} TBox can be axiomatized from observed graph data (Baader, Distel, 2008). If the data is deemed incomplete, the latter approach can interact with the experts to ask for additional data when the validity of a TBox statement cannot be determined yet (Baader, Distel, 2009). Both the unsupervised and the interactive approach terminate with a TBox that is sound and complete for the provided data, i.e. it entails a TBox statement if and only if that statement holds in the data. Moreover, the FCA-based axiomatization method was extended towards more expressive DLs (Kriegel, 2017, 2019b). Confident CI bases axiomatize all CIs that are valid already for a sufficiently large portion of all objects (Borchmann, 2013, 2015). There are also other interactive approaches (Klarman, Britz, 2015; Konev, Lutz, Ozaki, Wolter, 2017) but which seem to have only limited practical value since the experts are required to terminate the process manually when they believe that the target TBox has been found (i.e. completeness of the constructed TBox is not guaranteed by the approach, but must be detected by the experts).

Our contributions are as follows. We reconsider the FCA-based approach to completely axiomatizing \mathcal{EL} CIs from graph data (Baader, Distel, 2008, 2009) and

1. thoroughly revise and simplify its technical description including proofs,
2. equip it with support for already known CIs valid in the data (thus enabling it for ontology completion),
3. analyze its computational complexity,
4. explain how further types of TBox statements supported by the \mathcal{EL} family that are not just syntactic sugar can be completely axiomatized, viz. range restrictions and role inclusions,
5. describe how it can be implemented efficiently,
6. introduce variations that dispense with the computation of disjointness axioms or extremely large CIs without practical relevance, thereby rendering the approach applicable in practise, albeit some completeness is lost,
7. and evaluate the implementation on real-world datasets.

This extended version contains all technical details and proofs not included in the conference article for space restrictions.

2 Preliminaries

2.1 The \mathcal{EL} Family of DLs and OWL 2 EL

Fix a signature consisting of *individual names (INs)*, *concept names (CNs)*, and *role names (RNs)*. *Concept descriptions (CDs)* are built by $C ::= \top \mid \perp \mid A \mid C \sqcap D \mid \exists r.C$ where A ranges over all CNs and r over all RNs. We call \top the *top CD*, \perp the *bottom CD*, $C \sqcap D$ a *conjunction*, and $\exists r.C$ an *existential restriction (ER)*. A *TBox* is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$, *range restrictions (RRs)* $\top \sqsubseteq \forall r.C$, and *role inclusions (RIs)* $R \sqsubseteq s$, involving CDs C, D , RNs r, s , and *role chains* $R ::= \varepsilon \mid r \mid R \circ R$. An *ABox* is a finite set of *concept assertions (CAs)* $a : C$ and *role assertions (RAs)* $(a, b) : r$. An *ontology* consists of a TBox and an ABox. The \mathcal{EL} family and OWL 2 EL additionally allow for nominals $\{a\}$ in CDs, but we ignore these to avoid overfitting in the axiomatization method. We also ignore concrete domains (datatypes for strings, numbers, etc.) as no \mathcal{EL} reasoner currently supports them. As syntactic sugar we have *disjointness axioms* $C_1 \sqcap \dots \sqcap C_n \sqsubseteq \perp$, *domain restrictions* $\exists r. \top \sqsubseteq C$, *concept equivalences* $C \equiv D$ for the CIs $C \sqsubseteq D$ and $D \sqsubseteq C$, *role equivalences* $r \equiv s$ for the RIs $r \sqsubseteq s$ and $s \sqsubseteq r$, *transitivity axioms* $r \circ r \sqsubseteq r$, and *reflexivity axioms* $\varepsilon \sqsubseteq r$.

\mathcal{EL} can be translated into first-order logic and thus has a model-theoretic semantics, based on *interpretations* \mathcal{I} consisting of a non-empty set $\text{Dom}(\mathcal{I})$, called the *domain*, and of a function $\cdot^{\mathcal{I}}$ that gives meaning to the INs a , CNs A , and RNs r in the signature by assigning them to elements $a^{\mathcal{I}}$, subsets $A^{\mathcal{I}}$, and binary relations $r^{\mathcal{I}}$ of $\text{Dom}(\mathcal{I})$. The interpretation function is extended to compound CDs: $\top^{\mathcal{I}} := \text{Dom}(\mathcal{I})$, $\perp^{\mathcal{I}} := \emptyset$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$, and $(\exists r.C)^{\mathcal{I}} := \{x \mid r^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\}$ where $r^{\mathcal{I}}(x) := \{y \mid (x, y) \in r^{\mathcal{I}}\}$. Furthermore, \mathcal{I} satisfies a CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, a RR $\top \sqsubseteq \forall r.C$ if $\bigcup \{r^{\mathcal{I}}(x) \mid x \in \text{Dom}(\mathcal{I})\} \subseteq C^{\mathcal{I}}$, a RI $R \sqsubseteq s$ if $R^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ where $\varepsilon^{\mathcal{I}} := \{(x, x) \mid x \in \text{Dom}(\mathcal{I})\}$ and $(R \circ S)^{\mathcal{I}} := \{(x, z) \mid (x, y) \in R^{\mathcal{I}} \text{ and } (y, z) \in S^{\mathcal{I}} \text{ for some } y\}$, a CA $a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and a RA $(a, b) : r$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. We say that \mathcal{I} is a *model* of \mathcal{O} if \mathcal{I} satisfies all axioms in \mathcal{O} .

Reasoning is the process of deciding or enumerating consequences of an ontology. An ontology \mathcal{O} entails an axiom α , written $\mathcal{O} \models \alpha$, if α is satisfied in every model of \mathcal{O} . In this case α follows from the axioms in \mathcal{O} by logical inference. Entailment in the \mathcal{EL} family can be decided in polynomial time with the Completion algorithm (Baader, Brandt, Lutz, 2005, 2008), which is implemented in the reasoner ELK (Kazakov, Klinov, 2015; Kazakov, Krötzsch, Simančík, 2014). It uses rules to materialize implicit consequences by adding these as axioms to the ontology. In order to check whether an axiom α (in a particular normal form) follows from \mathcal{O} , one simply needs to check whether α is materialized until the algorithm terminates. Subsumption is a special form of entailment: D *subsumes* C w.r.t. \mathcal{T} , written $C \sqsubseteq^{\mathcal{T}} D$, if \mathcal{T} entails $C \sqsubseteq D$.

2.2 Simulations and the DL $\mathcal{EL}_{\text{si}}^{\perp}$

Given interpretations \mathcal{I} and \mathcal{J} , a *simulation* from \mathcal{I} to \mathcal{J} is a relation $\mathfrak{S} \subseteq \text{Dom}(\mathcal{I}) \times \text{Dom}(\mathcal{J})$ that fulfills the following conditions:

(S1) If $x \in A^{\mathcal{I}}$ and $(x, y) \in \mathfrak{S}$, then $y \in A^{\mathcal{J}}$.

(S2) If $(x, x') \in r^{\mathcal{I}}$ and $(x, y) \in \mathfrak{S}$, then $(y, y') \in r^{\mathcal{J}}$ and $(x', y') \in \mathfrak{S}$ for some y' .

We write $(\mathcal{I}, x) \preceq (\mathcal{J}, y)$ if there is a simulation from \mathcal{I} to \mathcal{J} that contains (x, y) . This binary relation \preceq is a pre-order (reflexive and transitive). If \mathcal{I} and \mathcal{J} are clear from the context, then it suffices to write $x \preceq y$. We denote by $x \approx y$ that $x \preceq y$ and $y \preceq x$, and by $x \prec y$ that $x \preceq y$ but $y \not\preceq x$.

Since the empty relation is a simulation and the union of simulations is a simulation, there is a maximal simulation from \mathcal{I} to \mathcal{J} , denoted by $\mathfrak{S}_{\mathcal{I}, \mathcal{J}}$. It is computable in polynomial time by starting with the full relation and subsequently deleting pairs that violate Condition (S1) or (S2). The relations \preceq and $\mathfrak{S}_{\mathcal{I}, \mathcal{J}}$ are equal since $x \preceq y$ iff. $(x, y) \in \mathfrak{S}_{\mathcal{I}, \mathcal{J}}$.

Simulations characterize the semantics of \mathcal{EL} : we can rewrite each CD C but \perp into a tree-shaped interpretation \mathcal{I}_C with root x_C such that $C^{\mathcal{J}} = \{y \mid (\mathcal{I}_C, x_C) \preceq (\mathcal{J}, y)\}$ for each \mathcal{J} . The idea underlying the DL $\mathcal{EL}_{\text{si}}^{\perp}$ (Lutz, Piro, Wolter, 2010) is to replace (\mathcal{I}_C, x_C) with an arbitrary finite, pointed interpretation (\mathcal{I}, x) : it features the additional CDs $\exists^{\text{sim}}(\mathcal{I}, x)$ where $(\exists^{\text{sim}}(\mathcal{I}, x))^{\mathcal{J}} := \{y \mid (\mathcal{I}, x) \preceq (\mathcal{J}, y)\}$.

We often use denotations $\exists^{\text{sim}}(\mathcal{C}, c)$, $\exists^{\text{sim}}(\mathcal{D}, d)$, ... for such CDs. For convenience we may use ABox notation: we specify $\text{Dom}(\mathcal{C})$ as usual but describe the function $\cdot^{\mathcal{C}}$ by the set $\{x : A \mid x \in A^{\mathcal{C}}\} \cup \{(x, y) : r \mid (x, y) \in r^{\mathcal{C}}\}$.

Lemma I. *Each $\mathcal{EL}_{\text{si}}^{\perp}$ CD C can be transformed in polynomial time into an equivalent CD that is either \perp or has the form $\exists^{\text{sim}}(\mathcal{C}, c)$.*

Proof. If C contains \perp as a sub-CD, then C is equivalent to \perp . Otherwise, we rewrite C as follows. Assume that

$$C = \prod_{i=1}^{\ell} A_i \sqcap \prod_{j=1}^m \exists r_j . D_j \sqcap \prod_{k=1}^n \exists^{\text{sim}}(\mathcal{E}_k, e_k).$$

We first recursively rewrite each D_j into the form $\exists^{\text{sim}}(\mathcal{D}_j, d_j)$. Afterwards, we obtain \mathcal{C} as the union of all \mathcal{D}_j and all \mathcal{E}_k , augmented with a fresh root element c which is assigned to the CNs A_i , has the objects d_j as r_j -successors, and gets all assignments of the objects e_k . Formally, it has the domain $\text{Dom}(\mathcal{C}) := \{c\} \cup \bigcup_{j=1}^m \text{Dom}(\mathcal{D}_j) \cup \bigcup_{k=1}^n \text{Dom}(\mathcal{E}_k)$ and its interpretation function is defined as

$$\begin{aligned} \cdot^{\mathcal{C}} := & \bigcup_{i=1}^{\ell} \{c : A_i\} \\ & \cup \bigcup_{j=1}^m (\{(c, d_j) : r_j\} \cup \cdot^{\mathcal{D}_j}) \\ & \cup \bigcup_{k=1}^n (\{c : A \mid e_k : A \in \cdot^{\mathcal{E}_k}\} \\ & \quad \cup \{(c, x) : s \mid (e_k, x) : s \in \cdot^{\mathcal{E}_k}\} \cup \cdot^{\mathcal{E}_k}). \end{aligned}$$

This mere rewriting of C into a another graph representation can obviously be computed in polynomial time. \square

Extensions of $\mathcal{EL}_{\text{si}}^{\perp}$ CDs can be read off from maximal simulations as follows, where $\mathfrak{S}(x) := \{y \mid (x, y) \in \mathfrak{S}\}$.

Lemma 1. $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}} = \mathfrak{S}_{\mathcal{C}, \mathcal{I}}(c)$

Proof. Recall that $(\mathcal{C}, c) \preceq (\mathcal{I}, x)$ iff. $(c, x) \in \mathfrak{S}_{\mathcal{C}, \mathcal{I}}$, where $\mathfrak{S}_{\mathcal{C}, \mathcal{I}}$ is the maximal simulation from \mathcal{C} to \mathcal{I} . We conclude that $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}} = \mathfrak{S}_{\mathcal{C}, \mathcal{I}}(c)$. \square

Subsumption w.r.t. the empty TBox is characterized as follows. See Proposition 3 for non-empty TBoxes.

Lemma 2. *The following are equivalent:*

1. $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq^{\emptyset} \exists^{\text{sim}}(\mathcal{D}, d)$
2. $(\mathcal{D}, d) \preceq (\mathcal{C}, c)$
3. $c \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\mathcal{C}}$

Proof. Statements 2 and 3 are equivalent by definition.

Let $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq^{\emptyset} \exists^{\text{sim}}(\mathcal{D}, d)$. Since the reflexive relation $\{(x, x) \mid x \in \text{Dom}(\mathcal{C})\}$ is a simulation on \mathcal{C} that contains (c, c) , we have $c \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{C}}$. It follows that $c \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\mathcal{C}}$.

Consider a simulation \mathfrak{S} from \mathcal{D} to \mathcal{C} that contains (d, c) . To verify the subsumption, assume further that \mathcal{I} is an interpretation and $x \in \text{Dom}(\mathcal{I})$. Now let $x \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}}$, i.e. there is a simulation \mathfrak{T} from \mathcal{C} to \mathcal{I} that contains (c, x) . The composition of \mathfrak{S} and \mathfrak{T} is a simulation from \mathcal{D} to \mathcal{I} that contains (d, x) , and thus $x \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\mathcal{I}}$. \square

The set of *top-level conjuncts* of an \mathcal{EL}_{si} CD $\exists^{\text{sim}}(\mathcal{C}, c)$ is

$$\text{Conj}(\exists^{\text{sim}}(\mathcal{C}, c)) := \{A \mid c \in A^{\mathcal{C}}\} \\ \cup \{\exists r. \exists^{\text{sim}}(\mathcal{C}, u) \mid (c, u) \in r^{\mathcal{C}}\}.$$

It is a finger exercise to show that $\exists^{\text{sim}}(\mathcal{C}, c)$ is equivalent to the conjunction $\prod \text{Conj}(\exists^{\text{sim}}(\mathcal{C}, c))$. Furthermore, $C \sqsubseteq^{\emptyset} D$ iff, for each $F \in \text{Conj}(D)$, there is some $E \in \text{Conj}(C)$ such that $E \sqsubseteq^{\emptyset} F$. The latter *atomic subsumptions* have the following cases:

- If both atoms are CNs, then $A \sqsubseteq^{\emptyset} B$ iff. $A = B$.
- If both atoms are ERs, then $\exists r. G \sqsubseteq^{\emptyset} \exists s. H$ iff. $r = s$ and $G \sqsubseteq^{\emptyset} H$.
- There are no atomic subsumptions between a CN and an ER, i.e. $A \not\sqsubseteq^{\emptyset} \exists s. H$ and $\exists r. G \not\sqsubseteq^{\emptyset} B$.

Consider an \mathcal{EL}_{si} CD $\exists^{\text{sim}}(\mathcal{C}, c)$, and let $n \geq 0$. The *unfolding* of $\exists^{\text{sim}}(\mathcal{C}, c)$ up to depth n , denoted as $\text{Unf}^n(\mathcal{C}, c)$ is recursively defined as follows:

- $\text{Unf}^0(\mathcal{C}, c) := \prod \{A \mid c \in A^{\mathcal{C}}\}$
- $\text{Unf}^{n+1}(\mathcal{C}, c) := \prod \{A \mid c \in A^{\mathcal{C}}\} \\ \cap \prod \{\exists r. \text{Unf}^n(\mathcal{C}, c') \mid (c, c') \in r^{\mathcal{C}}\}$

We often use the abbreviation $C \upharpoonright_n := \text{Unf}^n(\mathcal{C}, c)$ when $C = \exists^{\text{sim}}(\mathcal{C}, c)$.

Lemma II. *For each \mathcal{EL}_{si} CD $\exists^{\text{sim}}(\mathcal{C}, c)$ and for each finite interpretation \mathcal{I} , there is a number $n \geq 0$ with $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}} = (\text{Unf}^n(\mathcal{C}, c))^{\mathcal{I}}$.*

Proof. First of all, consider an arbitrary $d \in \text{Dom}(\mathcal{C})$. Since $\text{Unf}^m(\mathcal{C}, d) \sqsubseteq^{\emptyset} \text{Unf}^n(\mathcal{C}, d)$ if $m \geq n$, we then have $(\text{Unf}^m(\mathcal{C}, d))^{\mathcal{I}} \subseteq (\text{Unf}^n(\mathcal{C}, d))^{\mathcal{I}}$. As the domain $\text{Dom}(\mathcal{I})$ is finite, there must exist a number $\ell(d) \geq 0$ with $(\text{Unf}^n(\mathcal{C}, d))^{\mathcal{I}} = (\text{Unf}^{\ell(d)}(\mathcal{C}, d))^{\mathcal{I}}$ for each $n \geq \ell(d)$. It follows that

$$(\text{Unf}^{\ell(d)}(\mathcal{C}, d))^{\mathcal{I}} = \bigcap \{(\text{Unf}^n(\mathcal{C}, d))^{\mathcal{I}} \mid n \geq 0\}.$$

We will show that $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}} = (\text{Unf}^{\ell(c)}(\mathcal{C}, c))^{\mathcal{I}}$. For $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq^{\emptyset} \text{Unf}^{\ell(c)}(\mathcal{C}, c)$, the inclusion \subseteq holds. We

proceed with the opposite inclusion, and therefore consider an object $x \in (\text{Unf}^{\ell(c)}(\mathcal{C}, c))^{\mathcal{I}}$. To verify that $x \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}}$, we show that the following relation is a simulation from \mathcal{C} to \mathcal{I} containing (c, x) .

$$\mathfrak{S} := \{(d, y) \mid y \in (\text{Unf}^{\ell(d)}(\mathcal{C}, d))^{\mathcal{I}}\}$$

Clearly, $x \in (\text{Unf}^{\ell(c)}(\mathcal{C}, c))^{\mathcal{I}}$ yields $(c, x) \in \mathfrak{S}$.

- (S1) Let $(d, y) \in \mathfrak{S}$ and $d \in A^{\mathcal{C}}$. Then the CN A is a top-level conjunct of $\text{Unf}^{\ell(d)}(\mathcal{C}, d)$, and thus $y \in A^{\mathcal{I}}$.
- (S2) Assume $(d, y) \in \mathfrak{S}$ and $(d, d') \in r^{\mathcal{C}}$, and further let $\ell \geq \ell(d') + 1$ and $\ell \geq \ell(d)$. Then the ER $\exists r. \text{Unf}^{\ell-1}(\mathcal{C}, d')$ is a top-level conjunct of $\text{Unf}^{\ell}(\mathcal{C}, d)$, and so there is y' with $(y, y') \in r^{\mathcal{I}}$ and $y' \in (\text{Unf}^{\ell-1}(\mathcal{C}, d'))^{\mathcal{I}}$, i.e. $(y', d') \in \mathfrak{S}$. \square

\mathcal{EL}_{si} is equi-expressive to $\mathcal{EL}_{\text{gfp}}$ (Baader, 2003), which is an extension of \mathcal{EL} with greatest fixed-point semantics, and there are polynomial-time computable translations between them.

An $\mathcal{EL}_{\text{gfp}}$ CD is a pair (X, \mathcal{T}) consisting of a *concept variable* X and a TBox \mathcal{T} that contains one *concept definition* $Y \equiv C_Y$ for each concept variable Y , where C_Y is an \mathcal{EL} CD in which concept variables can be used in place of CNs.

The extension of (X, \mathcal{T}) in an interpretation \mathcal{I} is defined by means of the gfp-model of \mathcal{I} w.r.t. \mathcal{T} , which we introduce next. To this end, we consider interpretations \mathcal{J} with same domain $\text{Dom}(\mathcal{I})$ and of which the function $\cdot^{\mathcal{J}}$ coincides with $\cdot^{\mathcal{I}}$ on all INs, CNs, and RNs in the signature, but additionally sends each concept variable Y to a subset $Y^{\mathcal{J}}$ of $\text{Dom}(\mathcal{I})$. These extended interpretations can be ordered: $\mathcal{J}_1 \leq \mathcal{J}_2$ if $Y^{\mathcal{J}_1} \subseteq Y^{\mathcal{J}_2}$ for each concept variable Y . The mapping f sends each extended interpretation \mathcal{J} to the extended interpretation $f(\mathcal{J})$ where $Y^{f(\mathcal{J})} := (C_Y)^{\mathcal{J}}$ for all concept variables Y . Each fixed point of f is a model of \mathcal{T} . Since f is order-preserving, f has a greatest fixed point by Tarski's fixed-point theorem (Tarski, 1955), which we denote as \mathcal{I}^* and call the *gfp-model* of \mathcal{T} based on \mathcal{I} . With that we define the extension $(X, \mathcal{T})^{\mathcal{I}} := (C_X)^{\mathcal{I}^*}$.

Obviously, every \mathcal{EL} CD C is equivalent to the $\mathcal{EL}_{\text{gfp}}$ CD $(X, \{X \equiv C\})$.

Proposition III. *The description logics $\mathcal{EL}_{\text{gfp}}$ and \mathcal{EL}_{si} are polynomially equivalent.*

Proof. Given an \mathcal{EL}_{si} CD $\exists^{\text{sim}}(\mathcal{I}, x)$, we define the $\mathcal{EL}_{\text{gfp}}$ CD (x, \mathcal{T}) where the TBox \mathcal{T} consists of the concept definitions

$$y \equiv \prod \{A \mid y \in A^{\mathcal{I}}\} \cap \prod \{\exists r. z \mid (y, z) \in r^{\mathcal{I}}\}$$

for all $y \in \text{Dom}(\mathcal{I})$. Both CDs are equivalent (Baader, 2003, Proposition 6). Obviously, this translation from \mathcal{EL}_{si} to $\mathcal{EL}_{\text{gfp}}$ can be computed in polynomial time.

Vice versa, let (X, \mathcal{T}) be any $\mathcal{EL}_{\text{gfp}}$ CD. W.l.o.g. \mathcal{T} is normalized (Baader, 2003), i.e. all fillers of ERs occurring in right-hand sides of concept definitions in \mathcal{T} are concept variables. It is now straightforward to transform \mathcal{T} into an interpretation \mathcal{I} that contains all concept variables in its domain. The CDs (X, \mathcal{T}) and $\exists^{\text{sim}}(\mathcal{I}, X)$ are equivalent (Baader, 2003, Proposition 6). Since the normalization of

\mathcal{T} can be obtained in polynomial time (Baader, 2003), this translation from $\mathcal{EL}_{\text{gfp}}$ to \mathcal{EL}_{si} is computable in polynomial time as well. \square

2.3 Rule-Based Calculus for Subsumption

We introduce a rule-based calculus with which subsumption w.r.t. a TBox \mathcal{T} of $\mathcal{EL}_{\text{si}}^\perp$ CIs can be decided. Without loss of generality, \mathcal{T} must satisfy the following conditions:

- All CIs in \mathcal{T} have the form $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \exists^{\text{sim}}(\mathcal{F}, f)$ or $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$.
- If two interpretations \mathcal{E} and \mathcal{F} occur in \mathcal{T} (not necessarily in the same CI), and there is an object $x \in \text{Dom}(\mathcal{E}) \cap \text{Dom}(\mathcal{F})$, then \mathcal{E} and \mathcal{F} are equal on their parts reachable from x , i.e. there are simulations between \mathcal{E} and \mathcal{F} that contain (x, x) and otherwise only contain pairs in which both components are equal.

According to the first condition, the TBox \mathcal{T} can be partitioned into a subset \mathcal{T}_\perp consisting of all CIs with conclusion \perp and a subset \mathcal{T}_+ consisting of the remaining CIs. The latter condition ensures that we do not need to distinguish between “the x in \mathcal{E} ” and “the x in \mathcal{F} ”

First, we are going to prove that, in order to decide whether $\exists^{\text{sim}}(\mathcal{C}, c)$ is subsumed by $\exists^{\text{sim}}(\mathcal{D}, d)$ w.r.t. \mathcal{T}_+ , we can first *saturate* the left interpretation \mathcal{C} by means of the CIs in \mathcal{T}_+ , which yields the *saturation* $\text{sat}(\mathcal{C}, \mathcal{T}_+)$, and then check if $\exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$ is subsumed by $\exists^{\text{sim}}(\mathcal{D}, d)$, now w.r.t. the empty TBox \emptyset . Specifically, $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is obtained by exhaustive applications of the \sqsubseteq_+ -Rule starting with \mathcal{C} .

\sqsubseteq_+ -Rule. If there is an object x in $\text{Dom}(\mathcal{C})$ and a CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \exists^{\text{sim}}(\mathcal{F}, f)$ in \mathcal{T}_+ such that $(\mathcal{E}, e) \preceq (\mathcal{C}, x)$ but $(\mathcal{F}, f) \not\preceq (\mathcal{C}, x)$, then yield the interpretation \mathcal{C}' with domain $\text{Dom}(\mathcal{C}') := \text{Dom}(\mathcal{C}) \cup \text{Dom}(\mathcal{F})$ and function

$$\begin{aligned} \cdot^{\mathcal{C}'} &:= \cdot^{\mathcal{C}} \cup \cdot^{\mathcal{F}} \cup \{x : A \mid f : A \in \cdot^{\mathcal{F}}\} \\ &\quad \cup \{(x, y) : r \mid (f, y) : r \in \cdot^{\mathcal{F}}\}. \end{aligned}$$

The obtained interpretation satisfies $(\mathcal{F}, f) \preceq (\mathcal{C}', x)$. The following is an easy consequence.

Proposition IV. *$\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is a model of \mathcal{T}_+ , contains \mathcal{C} as a sub-interpretation, and is computable in polynomial time.*

Proof. The first statement holds since the \sqsubseteq_+ -Rule is not applicable to $\text{sat}(\mathcal{C}, \mathcal{T}_+)$. The second statement holds since the \sqsubseteq_+ -Rule only extends the initial interpretation \mathcal{C} . It remains to show the third statement. The domain of $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is polynomially large since it is a subset of the union of $\text{Dom}(\mathcal{C})$ and all $\text{Dom}(\mathcal{F})$ where $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \exists^{\text{sim}}(\mathcal{F}, f)$ is in \mathcal{T}_+ . For each domain element and for each CI in \mathcal{T}_+ , the \sqsubseteq_+ -Rule can be applied at most once, and one application takes polynomial time. Consequently, $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ can be computed in polynomial time. \square

Lemma V. *The following statements are equivalent:*

1. $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq_{\mathcal{T}_+} \exists^{\text{sim}}(\mathcal{D}, d)$
2. $\exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c) \sqsubseteq^\emptyset \exists^{\text{sim}}(\mathcal{D}, d)$
3. $(\mathcal{D}, d) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$

Proof. We will implicitly use Lemma 2 throughout the proof. With that, Statements 2 and 3 are equivalent.

Next, we prove that Statement 1 implies Statement 2. So let $\exists^{\text{sim}}(\mathcal{C}, c)$ be subsumed by $\exists^{\text{sim}}(\mathcal{D}, d)$ w.r.t. \mathcal{T}_+ . Since \mathcal{C} is a sub-interpretation of $\text{sat}(\mathcal{C}, \mathcal{T}_+)$, the relation $\{(x, x) \mid x \in \text{Dom}(\mathcal{C})\}$ is a simulation from \mathcal{C} to $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ containing (c, c) . It follows that $(\mathcal{C}, c) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$, i.e. $c \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\text{sat}(\mathcal{C}, \mathcal{T}_+)}$. As $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is a model of \mathcal{T}_+ , the assumption yields that the CI $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, d)$ is satisfied in $\text{sat}(\mathcal{C}, \mathcal{T}_+)$. We infer that $c \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\text{sat}(\mathcal{C}, \mathcal{T}_+)}$, i.e. $\exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c) \sqsubseteq^\emptyset \exists^{\text{sim}}(\mathcal{D}, d)$.

Last, we show that Statement 2 implies Statement 1. We therefore assume that $\exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$ is subsumed by $\exists^{\text{sim}}(\mathcal{D}, d)$, i.e. $(\mathcal{D}, d) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$, and we consider a model \mathcal{I} of the TBox \mathcal{T}_+ where $u \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}}$, i.e. $(\mathcal{C}, c) \preceq (\mathcal{I}, u)$. We are going to prove that $(\text{sat}(\mathcal{C}, \mathcal{T}_+), c) \preceq (\mathcal{I}, u)$. From this and the assumption it follows that $(\mathcal{D}, d) \preceq (\mathcal{I}, u)$, i.e. $u \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\mathcal{I}}$. Since \mathcal{I} and u are arbitrary, we have verified that \mathcal{T}_+ entails $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, d)$.

It remains to demonstrate that $(\text{sat}(\mathcal{C}, \mathcal{T}_+), c) \preceq (\mathcal{I}, u)$, which we do by induction along the sequence of applications of the \sqsubseteq_+ -Rule that produces $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ from \mathcal{C} . To this end, assume that $(\mathcal{C}', c) \preceq (\mathcal{I}, u)$ and that \mathcal{C}'' is obtained from \mathcal{C}' by one application of the \sqsubseteq_+ -Rule, say for the object $x \in \text{Dom}(\mathcal{C}')$ and the CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \exists^{\text{sim}}(\mathcal{F}, f)$, i.e. $(\mathcal{E}, e) \preceq (\mathcal{C}', x)$ and $(\mathcal{F}, f) \not\preceq (\mathcal{C}', x)$. In order to show that $(\mathcal{C}'', c) \preceq (\mathcal{I}, u)$, we consider the union $\mathfrak{S} := \mathfrak{S}_{\mathcal{C}', \mathcal{I}} \cup \mathfrak{S}_{\mathcal{F}, \mathcal{I}}$ of the maximal simulation from \mathcal{C}' to \mathcal{I} and the maximal simulation from \mathcal{F} to \mathcal{I} , and validate that it is a simulation from \mathcal{C}'' to \mathcal{I} that contains (c, u) . Since (c, u) is in $\mathfrak{S}_{\mathcal{C}', \mathcal{I}}$, also \mathfrak{S} contains (c, u) . We now verify that \mathfrak{S} satisfies Conditions (S1) and (S2). The only interesting case deals with the object x at which (\mathcal{F}, f) is merged into \mathcal{C}' , yielding \mathcal{C}'' . Therefore let $(x, v) \in \mathfrak{S}$. Since $x \neq f$ (otherwise $(\mathcal{F}, f) \preceq (\mathcal{C}', x)$ by our second assumption on \mathcal{T} , which would contradict applicability of the \sqsubseteq_+ -Rule), it follows that $(x, v) \in \mathfrak{S}_{\mathcal{C}', \mathcal{I}}$, i.e. $(\mathcal{C}', x) \preceq (\mathcal{I}, v)$, and thus $(\mathcal{E}, e) \preceq (\mathcal{I}, v)$. Since \mathcal{I} is a model of \mathcal{T}_+ , we infer that $(\mathcal{F}, f) \preceq (\mathcal{I}, v)$, i.e. $(f, v) \in \mathfrak{S}_{\mathcal{F}, \mathcal{I}}$.

(S1) Assume that $x \in A^{\mathcal{C}''}$. We distinguish two cases.

- If $x \in A^{\mathcal{C}'}$, then $v \in A^{\mathcal{I}}$ since $(x, v) \in \mathfrak{S}_{\mathcal{C}', \mathcal{I}}$.
- Otherwise, we have $f \in A^{\mathcal{F}}$ and thus $v \in A^{\mathcal{I}}$ since $(f, v) \in \mathfrak{S}_{\mathcal{F}, \mathcal{I}}$.

(S2) Now let $(x, y) \in r^{\mathcal{C}''}$. Again, we consider two cases.

- In the first case we have $(x, y) \in r^{\mathcal{C}'}$. Since $(x, v) \in \mathfrak{S}_{\mathcal{C}', \mathcal{I}}$, there is some w such that $(v, w) \in r^{\mathcal{I}}$ and $(y, w) \in \mathfrak{S}_{\mathcal{C}', \mathcal{I}}$. Of course, then $(y, w) \in \mathfrak{S}$.
- It remains the case where $(f, y) \in r^{\mathcal{F}}$. From $(f, v) \in \mathfrak{S}_{\mathcal{F}, \mathcal{I}}$ we infer that there is some w such that $(v, w) \in r^{\mathcal{I}}$ and $(y, w) \in \mathfrak{S}_{\mathcal{F}, \mathcal{I}}$. Then $(y, w) \in \mathfrak{S}$. \square

Next, we take the CIs in \mathcal{T}_\perp into account. They can render $\mathcal{EL}_{\text{si}}^\perp$ CDs unsatisfiable w.r.t. the whole TBox \mathcal{T} .

Lemma VI. *$\exists^{\text{sim}}(\mathcal{C}, c)$ is unsatisfiable w.r.t. \mathcal{T} , i.e. $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq_{\mathcal{T}} \perp$, iff. there is a CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$ in \mathcal{T}_\perp where $(\mathcal{E}, e) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), x)$ for an object x reachable from c .*

Proof. Assume that $\exists^{\text{sim}}(\mathcal{C}, c)$ is unsatisfiable w.r.t. \mathcal{T} , i.e. $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{J}} = \emptyset$ for each model \mathcal{J} of \mathcal{T} . Consider the saturation $\text{sat}(\mathcal{C}, \mathcal{T}_+)$. It follows from Lemma V that $(\mathcal{C}, c) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$. Now denote by \mathcal{S} the sub-interpretation of $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ that consists only of all objects reachable from c . Then $(\mathcal{C}, c) \preceq (\mathcal{S}, c)$ and thus $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{S}} \neq \emptyset$. We conclude that \mathcal{S} is no model of \mathcal{T} . However, $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is a model of \mathcal{T}_+ by Proposition IV and thus also \mathcal{S} . So there must be a CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$ in \mathcal{T}_\perp with $(\exists^{\text{sim}}(\mathcal{E}, e))^{\mathcal{S}} \neq \emptyset$, which means that there is an object $x \in \text{Dom}(\mathcal{S})$ such that $(\mathcal{E}, e) \preceq (\mathcal{S}, x)$. By choice of \mathcal{S} , x is reachable from c in $\text{sat}(\mathcal{C}, \mathcal{T}_+)$, i.e. we have $(\mathcal{E}, e) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), x)$.

In the converse direction, assume that x is reachable from c , say on a path with RNs r_1, \dots, r_k , and that there is a CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$ in \mathcal{T}_\perp such that $(\mathcal{E}, e) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), x)$. We infer that

$$\begin{aligned} \exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq^{\mathcal{T}} \exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c) \\ \sqsubseteq^{\emptyset} \exists r_1 \dots \exists r_k. \exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), x) \\ \sqsubseteq^{\emptyset} \exists r_1 \dots \exists r_k. \exists^{\text{sim}}(\mathcal{E}, e) \\ \sqsubseteq^{\mathcal{T}} \exists r_1 \dots \exists r_k. \perp \\ \sqsubseteq^{\emptyset} \perp, \end{aligned}$$

where the first subsumption follows from Lemma V. \square

Similarly, an interpretation \mathcal{I} cannot be extended to a model of \mathcal{T} if there is a CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$ in \mathcal{T}_\perp such that $(\mathcal{E}, e) \preceq (\text{sat}(\mathcal{I}, \mathcal{T}_+), x)$ for some domain element x .

In accordance with the above lemma, we define the following second rule that detects unsatisfiability.

\sqsubseteq_\perp -Rule. If there is an object x in $\text{Dom}(\mathcal{C})$ and a CI $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$ in \mathcal{T}_\perp such that $(\mathcal{E}, e) \preceq (\mathcal{C}, x)$, then fail.

Consider a CD $C := \exists^{\text{sim}}(\mathcal{C}, c)$. Exhaustively applying the \sqsubseteq_+ -Rule and the \sqsubseteq_\perp -Rule either fails in which case C is unsatisfiable w.r.t. \mathcal{T} and we define $C^{\mathcal{T}} := \perp$, or produces an interpretation \mathcal{C}' and then we set $C^{\mathcal{T}} := \exists^{\text{sim}}(\mathcal{C}', c)$. Moreover, let $\perp^{\mathcal{T}} := \perp$. We call $C^{\mathcal{T}}$ the *most specific consequence* of C w.r.t. \mathcal{T} , and it characterizes subsumption as follows.

Proposition 3. *Subsumption in $\mathcal{EL}_{\text{si}}^\perp$ can be decided in polynomial time. In particular, $C \sqsubseteq^{\mathcal{T}} D$ iff: $C^{\mathcal{T}} \sqsubseteq^{\emptyset} D$.*

Proof. If rule application fails, then C is unsatisfiable w.r.t. \mathcal{T} by Lemma VI, i.e. $C^{\mathcal{I}} = \emptyset$ for every model \mathcal{I} of \mathcal{T} . It follows that $C \equiv^{\mathcal{T}} \perp$, and $C^{\mathcal{T}} = \perp$. Thus $C \sqsubseteq^{\mathcal{T}} D$ and $C^{\mathcal{T}} \sqsubseteq^{\emptyset} D$ hold for all $\mathcal{EL}_{\text{si}}^\perp$ CDs D .

Otherwise, C is satisfiable w.r.t. \mathcal{T} , and $C^{\mathcal{T}} = \exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$. So $C^{\mathcal{T}} \sqsubseteq^{\emptyset} D$ iff. $C \sqsubseteq^{\mathcal{T}_+} D$ by Lemma V. It is trivial that $C \sqsubseteq^{\mathcal{T}_+} D$ implies $C \sqsubseteq^{\mathcal{T}} D$, and it remains to verify the converse implication.

First of all, $\mathcal{EL}_{\text{si}}^\perp$ is invariant under direct products, i.e. $C^{\mathcal{I} \times \mathcal{J}} = C^{\mathcal{I}} \times C^{\mathcal{J}}$ for all $\mathcal{EL}_{\text{si}}^\perp$ CDs C and for all interpretations \mathcal{I} and \mathcal{J} . This follows from the observation that the direct product operation \times is the infimum operation in the set of all (equivalence classes of) pointed interpretations ordered by \preceq , i.e. $(\mathcal{I} \times \mathcal{J}, (x, y)) \preceq (\mathcal{I}, x)$ and $(\mathcal{I} \times \mathcal{J}, (x, y)) \preceq (\mathcal{J}, y)$, and for each (\mathcal{C}, c) , if $(\mathcal{C}, c) \preceq (\mathcal{I}, x)$ and $(\mathcal{C}, c) \preceq (\mathcal{J}, y)$, then $(\mathcal{C}, c) \preceq (\mathcal{I} \times \mathcal{J}, (x, y))$.

Now let $C \sqsubseteq^{\mathcal{T}} D$, and consider a model \mathcal{I} of \mathcal{T}_+ where $x \in C^{\mathcal{I}}$. Since C is satisfiable w.r.t. \mathcal{T} , there is a model \mathcal{J} of \mathcal{T} such that $C^{\mathcal{J}} \neq \emptyset$. Product invariance yields that $\mathcal{I} \times \mathcal{J}$ is a model of \mathcal{T}_+ . It is also a model of the other CIs in \mathcal{T} : for each $E \sqsubseteq \perp \in \mathcal{T}_\perp$, we have $E^{\mathcal{J}} = \emptyset$ and thus $E^{\mathcal{I} \times \mathcal{J}} = \emptyset$ by product invariance. Since $C^{\mathcal{J}} \neq \emptyset$, there is some element $y \in C^{\mathcal{J}}$, and thus $(x, y) \in C^{\mathcal{I} \times \mathcal{J}}$. Since $\mathcal{I} \times \mathcal{J}$ is a model of \mathcal{T} , it follows that $(x, y) \in D^{\mathcal{I} \times \mathcal{J}}$, and thus $x \in D^{\mathcal{I}}$ by product invariance. We conclude that $C \sqsubseteq^{\mathcal{T}_+} D$.

Last, we analyze the computational complexity. Let $C := \exists^{\text{sim}}(\mathcal{C}, c)$ and $D := \exists^{\text{sim}}(\mathcal{D}, d)$. According to Proposition IV, the saturation $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is computable in polynomial time. Then, to compute $C^{\mathcal{T}}$ we try for the polynomially many objects in this saturation and the polynomially many CIs of the form $\exists^{\text{sim}}(\mathcal{E}, e) \sqsubseteq \perp$ in \mathcal{T}_\perp whether the \sqsubseteq_\perp -Rule fails, which needs polynomial time. If it fails, then $C^{\mathcal{T}} = \perp$ and thus the subsumption $C \sqsubseteq^{\mathcal{T}} D$ holds. Otherwise, we have $C^{\mathcal{T}} = \exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$ and to decide $C \sqsubseteq^{\mathcal{T}} D$ we check if there is a simulation from \mathcal{D} to $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ containing (d, c) , see Lemma 2, which needs polynomial time since the maximal simulation can be computed in polynomial time. \square

Lemma VII. *$\mathcal{EL}_{\text{si}}^\perp$ has the polynomial countermodel property: if a TBox \mathcal{T} does not entail a CI $C \sqsubseteq D$, then there is a model of \mathcal{T} that is a countermodel to $C \sqsubseteq D$ and has polynomial size (w.r.t. \mathcal{T} and C only).*

Proof. Assume $C := \exists^{\text{sim}}(\mathcal{C}, c)$ and $D := \exists^{\text{sim}}(\mathcal{D}, d)$ where w.l.o.g. each of the interpretations \mathcal{C} and \mathcal{D} has only one connected component. Since $C \not\sqsubseteq^{\mathcal{T}} D$, Proposition 3 yields that $C^{\mathcal{T}} \not\sqsubseteq^{\emptyset} D$. In particular, C is satisfiable w.r.t. \mathcal{T} and $C^{\mathcal{T}} = \exists^{\text{sim}}(\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$.

Since $C \sqsubseteq^{\mathcal{T}_+} C$ and $C \not\sqsubseteq^{\mathcal{T}_+} D$, we infer with Lemma V that $(\mathcal{C}, c) \preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$ but $(\mathcal{D}, d) \not\preceq (\text{sat}(\mathcal{C}, \mathcal{T}_+), c)$. By Lemma 2 it follows that $c \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\text{sat}(\mathcal{C}, \mathcal{T}_+)}$ but $c \notin (\exists^{\text{sim}}(\mathcal{D}, d))^{\text{sat}(\mathcal{C}, \mathcal{T}_+)}$, i.e. the saturation $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is a countermodel to $C \sqsubseteq D$.

According to Proposition IV the saturation $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ is a model of the sub-TBox \mathcal{T}_+ . Since the \sqsubseteq_\perp -Rule does not fail, this saturation is also a model of the sub-TBox \mathcal{T}_\perp , and thus of the whole TBox \mathcal{T} . Last, $\text{sat}(\mathcal{C}, \mathcal{T}_+)$ has polynomial size by Proposition IV. \square

Lemma VIII. *$C \sqsubseteq^{\mathcal{T}} D$ iff. $C \sqsubseteq^{\mathcal{T}} D \upharpoonright_n$ for each n .*

Proof. The only-if direction follows from $D \sqsubseteq^{\emptyset} D \upharpoonright_n$ for every $n \geq 0$. Regarding the if direction, assume that $\mathcal{T} \not\models C \sqsubseteq D$. By Lemma VII there is a finite model \mathcal{I} of \mathcal{T} that does not satisfy $C \sqsubseteq D$, i.e. $C^{\mathcal{I}} \not\sqsubseteq D^{\mathcal{I}}$. Lemma II further yields $D^{\mathcal{I}} = (D \upharpoonright_n)^{\mathcal{I}}$ for some $n \geq 0$. We conclude that \mathcal{I} neither satisfies the CI $C \sqsubseteq D \upharpoonright_n$, and thus $\mathcal{T} \not\models C \sqsubseteq D \upharpoonright_n$. \square

2.4 Formal Concept Analysis

FCA is concerned with analyzing a *formal context* $\mathbb{K} := (G, M, I)$, where G is a set of *objects*, M is a set of *attributes*, and $I \subseteq G \times M$ is an *incidence relation*. We express by $(g, m) \in I$ that the object g has the attribute m . The incidence relation I is used to define two operations:

- For every subset $A \subseteq G$, let A^I be the set of all attributes that the objects in A have in common, i.e. $A^I := \{m \mid (g, m) \in I \text{ for each } g \in A\}$.
- For every subset $B \subseteq M$, define B^I as the set of all objects that have all attributes in B , i.e. $B^I := \{g \mid (g, m) \in I \text{ for each } m \in B\}$.

An *implication* is an expression $U \rightarrow V$ where U and V are subsets of M . The context \mathbb{K} *satisfies* $U \rightarrow V$ if every object in G that has all attributes in U also has each attribute in V ; and \mathbb{K} is a *model* of an implication set \mathcal{L} if it satisfies every implication in \mathcal{L} . Furthermore, \mathcal{L} *entails* $U \rightarrow V$, denoted as $\mathcal{L} \models U \rightarrow V$, if $U \rightarrow V$ is satisfied in every model of \mathcal{L} .

Implication entailment can be decided in linear time by means of the algorithm `LinClosure` (Beeri, Bernstein, 1979).¹ This is done by computing the closure $U^\mathcal{L}$ of the premise U w.r.t. \mathcal{L} and then checking if V is a subset. Here, it suffices to know that $U^\mathcal{L}$ can be obtained, in the naïve way, from $U' := U$ by adding all attributes in Y to U' whenever one finds an implication $X \rightarrow Y$ in \mathcal{L} for which U' contains all attributes in the premise X but not all in the conclusion Y — the final U' is the closure.

An *implication base* of \mathbb{K} relative to \mathcal{L} is an implication set \mathcal{B} of which \mathbb{K} is a model and that together with \mathcal{L} is *complete*, i.e. $\mathcal{B} \cup \mathcal{L}$ entails each implication satisfied in \mathbb{K} . A *pseudo-intent* of \mathbb{K} w.r.t. \mathcal{L} is a subset $P \subseteq M$ such that

- (PI1) P is no intent (i.e. $P \neq P^{II}$),
- (PI2) P is closed under the implications in \mathcal{L} , i.e. for each implication $U \rightarrow V \in \mathcal{L}$, if $U \subseteq P$, then $V \subseteq P$ (i.e. $P = P^\mathcal{L}$), and
- (PI3) for each pseudo-intent Q , if $Q \subset P$, then $Q^{II} \subseteq P$.

The *canonical implication base* $\text{Can}(\mathbb{K}, \mathcal{L})$ consists of all implications $P \rightarrow P^{II}$ where P is a pseudo-intent (Guigues, Duquenne, 1986; Stumme, 1996). It can be computed in exponential time with the algorithm `NextClosure` (Ganter, 1984), and no base with fewer implications exists (Distel, 2011; Wild, 1994).

FCA can be seen as \mathcal{EL} without RNs and \perp . More specifically, a formal context $\mathbb{K} := (G, M, I)$ encodes the same information as the interpretation \mathcal{I} over the signature that contains all attributes in M as CNs, with domain $\text{Dom}(\mathcal{I}) := G$, and where $m^\mathcal{I} := \{g \mid (g, m) \in I\}$ for each $m \in M$. Then, implications $U \rightarrow V$ satisfied in \mathbb{K} correspond to CIs $\bigcap U \sqsubseteq \bigcap V$ satisfied in \mathcal{I} , using the syntactic sugar $\bigcap \{C_1, \dots, C_n\} := C_1 \sqcap \dots \sqcap C_n$ and $\bigcap \emptyset := \top$.

3 Axiomatization of \mathcal{EL} TBoxes

We first focus on axiomatizing CIs. RRs and RIs will be considered in Section 3.4. As input we expect graph data in form of an interpretation \mathcal{I} , which includes knowledge graphs, graph databases, and RDF data: the CNs are the node labels and the RNs are the edge labels. Preprocessing of a knowledge graph might be necessary, e.g. to correctly treat the metadata as well as to materialize the modelling conventions (Krötzsch, 2019). Using an interpretation

¹see also (Maier, 1983) for an alternative presentation.

as input means that we expect data under closed-world assumption — axiomatizing CIs from data under open-world assumption (e.g. an ABox) is pointless since only tautologies could be produced: for each non-trivial CI $C \sqsubseteq D$, there could be a still unknown individual in C that is not in D . Further given is a TBox \mathcal{T} of which \mathcal{I} is a model and that contains known CIs that should be preserved by the axiomatization. Note that \mathcal{T} might be empty. We will compute a CI base in the following sense.

Definition 4. A TBox is *complete* for \mathcal{I} if it entails all CIs satisfied in \mathcal{I} . A *CI base* of \mathcal{I} relative to \mathcal{T} is a TBox \mathcal{B} of which \mathcal{I} is a model and such that $\mathcal{B} \cup \mathcal{T}$ is complete for \mathcal{I} .

A CI base \mathcal{B} together with the given TBox \mathcal{T} axiomatizes all CIs satisfied in \mathcal{I} . We also call \mathcal{B} a *completion* of \mathcal{T} w.r.t. \mathcal{I} as we obtain a complete TBox by adding all CIs in \mathcal{B} to \mathcal{T} .

We will convert the interpretation \mathcal{I} into a formal context such that its implication base can be rewritten into a CI base of \mathcal{I} . We therefore use induced contexts (Rudolph, 2004).²

Definition 5. Let \mathbf{M} be a set of CDs. The *induced context* is $\mathbb{K}_\mathcal{I} := (\text{Dom}(\mathcal{I}), \mathbf{M}, I)$ where $(x, C) \in I$ iff. $x \in C^\mathcal{I}$.

Lemma 6. Given subsets $\mathbf{C}, \mathbf{D} \subseteq \mathbf{M}$, the CI $\bigcap \mathbf{C} \sqsubseteq \bigcap \mathbf{D}$ is satisfied in \mathcal{I} iff. the implication $\mathbf{C} \rightarrow \mathbf{D}$ is satisfied in $\mathbb{K}_\mathcal{I}$.

It follows that, if \mathcal{B} is an implication base of $\mathbb{K}_\mathcal{I}$, then the TBox $\bigcap \mathcal{B} := \{\bigcap \mathbf{C} \sqsubseteq \bigcap \mathbf{D} \mid \mathbf{C} \rightarrow \mathbf{D} \in \mathcal{B}\}$ has \mathcal{I} as a model. Whether this TBox is complete depends on the choice of the attribute set \mathbf{M} , which we will address next. The naïve way was to take the infinite set of all CDs as the attribute set \mathbf{M} , but it would then be unclear how a base could actually be computed.

3.1 Model-based Most Specific Concepts

For each subset X of $\text{Dom}(\mathcal{I})$, we denote by $X^\mathcal{I}$ the *model-based most specific CD (MMSCD)* of X in \mathcal{I} that is determined up to equivalence by the following conditions:

- (M1) $X \subseteq (X^\mathcal{I})^\mathcal{I}$
- (M2) for each CD C , if $X \subseteq C^\mathcal{I}$, then $X^\mathcal{I} \sqsubseteq^\emptyset C$.

We will omit braces and write $X^{\mathcal{II}}$ instead of $(X^\mathcal{I})^\mathcal{I}$.

The extended interpretation function $C \mapsto C^\mathcal{I}$, which maps each CD C to the set $C^\mathcal{I}$ of all objects in $\text{Dom}(\mathcal{I})$ satisfying C , corresponds to the mapping $B \mapsto B^I$ of a formal context $\mathbb{K} := (G, M, I)$, which sends each subset B of M to the set B^I of all objects in G having every attribute in B . Likewise, the mapping $X \mapsto X^\mathcal{I}$ from subsets of $\text{Dom}(\mathcal{I})$ to their MMSCDs corresponds to the mapping $A \mapsto A^I$ of \mathbb{K} . As in FCA these two operators form a Galois connection, i.e. all subsets X, Y of $\text{Dom}(\mathcal{I})$ and all CDs C, D satisfy the following properties (Baader, Distel, 2008).

- (G1) $X \subseteq C^\mathcal{I}$ iff. $X^\mathcal{I} \sqsubseteq^\emptyset C$
- (G2) $X^\mathcal{I} \sqsubseteq^\emptyset Y^\mathcal{I}$ if $X \subseteq Y$
- (G3) $X \subseteq X^{\mathcal{II}}$
- (G4) $X^\mathcal{I} \equiv^\emptyset X^{\mathcal{III}}$
- (G5) $C^\mathcal{I} \subseteq D^\mathcal{I}$ if $C \sqsubseteq^\emptyset D$
- (G6) $C^{\mathcal{II}} \sqsubseteq^\emptyset C$
- (G7) $C^\mathcal{I} = C^{\mathcal{III}}$

²Note that interpretations and binary power context families are the same. Earlier papers that apply FCA in a DL setting also needed to define formal contexts, of which some are similar.

For cycles in the interpretation \mathcal{I} , some MMSCDs might not be expressible in \mathcal{EL}^\perp but in $\mathcal{EL}_{\text{si}}^\perp$. The MMSCD of \emptyset is \perp and the MMSCD of each singleton $\{x\}$ is $\exists^{\text{sim}}(\mathcal{I}, x)$. MMSCDs of sets with two or more objects can be obtained as least common subsumers (Baader, Distel, 2008), computed by products. For instance, the MMSCD of $\{x, y\}$ in \mathcal{I} is $\exists^{\text{sim}}(\mathcal{I} \times \mathcal{I}, (x, y))$. However, this approach is infeasible when all MMSCDs in \mathcal{I} must be computed, like in our reduction to FCA. A more efficient way uses the powering, which is a permutation-invariant representation of all powers of \mathcal{I} of any arity.

Definition 7. The *powering* $\wp(\mathcal{I})$ is the interpretation with domain $\text{Dom}(\wp(\mathcal{I})) := \wp(\text{Dom}(\mathcal{I}))$ and interpr. function

$$\begin{aligned} \cdot^{\wp(\mathcal{I})} &:= \{X : A \mid X \subseteq A^{\mathcal{I}}\} \\ &\cup \left\{ (X, Y) : r \mid \begin{array}{l} Y \text{ is a minimal hitting set} \\ \text{of } \{r^{\mathcal{I}}(x) \mid x \in X\} \end{array} \right\}. \end{aligned}$$

Recall that a *hitting set* of a set \mathcal{S} of sets is a set H such that $H \cap S \neq \emptyset$ for each set $S \in \mathcal{S}$. We call H *minimal* if no strict subset is a hitting set. All minimal hitting sets can efficiently be computed with the algorithm HS-DAG (Greiner, Smith, Wilkerson, 1989; Reiter, 1987).

All MMSCDs but of \emptyset are computable by the powering. We use the following lemma to show this.

Lemma IX. $(\mathcal{J}, y) \preceq (\wp(\mathcal{I}), X)$ iff $(\mathcal{J}, y) \preceq (\mathcal{I}, x)$ for each $x \in X$.

Proof. For the only-if direction, assume that \mathfrak{S} is a simulation from \mathcal{J} to $\wp(\mathcal{I})$ that contains (y, X) . We will show that the following relation is a simulation from \mathcal{J} to \mathcal{I} that contains all pairs (y, x) where $x \in X$.

$$\mathfrak{T} := \{(v, u) \mid (v, U) \in \mathfrak{S} \text{ and } u \in U \text{ for some } U\}$$

Clearly, $(y, X) \in \mathfrak{S}$ implies $(y, x) \in \mathfrak{T}$ for each $x \in X$.

(S1) Let $(v, u) \in \mathfrak{T}$ and $v \in A^{\mathcal{J}}$. The former implies that there is some U with $(v, U) \in \mathfrak{S}$ and $u \in U$. Since \mathfrak{S} satisfies Condition (S1), it follows that $U \in A^{\wp(\mathcal{I})}$. This means $U \subseteq A^{\mathcal{I}}$, and thus $u \in A^{\mathcal{I}}$.

(S2) Consider $(v, u) \in \mathfrak{T}$ and $(v, v') \in r^{\mathcal{J}}$. By definition of \mathfrak{T} there is a set U with $(v, U) \in \mathfrak{S}$ and $u \in U$. Since \mathfrak{S} fulfills Condition (S2), we infer that there is some set U' with $(v', U') \in \mathfrak{S}$ and $(U, U') \in r^{\wp(\mathcal{I})}$. The latter means that U' is a minimal hitting set of $\{r^{\mathcal{I}}(u) \mid u \in U\}$, and so there is some object $u' \in U'$ with $(u, u') \in r^{\mathcal{I}}$. Last, we obviously have $(v', u') \in \mathfrak{T}$.

We proceed with the if direction. Therefore consider, for each $x \in X$, a simulation \mathfrak{T}_x from \mathcal{J} to \mathcal{I} that contains (y, x) . We will verify that the relation defined below is a simulation from \mathcal{J} to $\wp(\mathcal{I})$ that contains (y, X) .

$$\mathfrak{S} := \left\{ (v, U) \mid \begin{array}{l} \text{for each } u \in U, \text{ there is } x \in X \\ \text{with } (v, u) \in \mathfrak{T}_x \end{array} \right\}$$

We have $(y, X) \in \mathfrak{S}$ since $(y, x) \in \mathfrak{T}_x$ for each $x \in X$.

(S1) Take $(v, U) \in \mathfrak{S}$ with $v \in A^{\mathcal{J}}$. Then, for each $u \in U$, there is some $x_u \in X$ with $(v, u) \in \mathfrak{T}_{x_u}$. Since each \mathfrak{T}_{x_u} satisfies Condition (S1), we have $u \in A^{\mathcal{I}}$ for each $u \in U$, i.e. $U \subseteq A^{\wp(\mathcal{I})}$.

(S2) Finally, let $(v, U) \in \mathfrak{S}$ where $(v, v') \in r^{\mathcal{J}}$. The definition of \mathfrak{S} yields, for each $u \in U$, some $x_u \in X$ with $(v, u) \in \mathfrak{T}_{x_u}$. Consider one $x \in X$. For Condition (S2) fulfilled by \mathfrak{T}_{x_u} , there is some u' with $(v', u') \in \mathfrak{T}_{x_u}$ and $(u, u') \in r^{\mathcal{I}}$.

Clearly, the set $\{u' \mid u \in U\}$ is a hitting set of $\{r^{\mathcal{I}}(u) \mid u \in U\}$. It follows that there is a minimal hitting set U' of $\{r^{\mathcal{I}}(u) \mid u \in U\}$ with $U' \subseteq \{u' \mid u \in U\}$, and then $(U, U') \in r^{\wp(\mathcal{I})}$. We further have $(v', U') \in \mathfrak{S}$ since $(v', u') \in \mathfrak{T}_{x_u}$ for each $u \in U$. \square

It follows immediately that $\exists^{\text{sim}}(\wp(\mathcal{I}), X)$ satisfies Condition (M2). Specifically for $(\mathcal{J}, y) := (\wp(\mathcal{I}), X)$ we obtain that $(\wp(\mathcal{I}), X) \preceq (\mathcal{I}, x)$ for each $x \in X$, and thus $\exists^{\text{sim}}(\wp(\mathcal{I}), X)$ also satisfies Condition (M1).

Proposition 8. $X^{\mathcal{I}} \equiv^{\emptyset} \exists^{\text{sim}}(\wp(\mathcal{I}), X)$ if $\emptyset \neq X \subseteq \text{Dom}(\mathcal{I})$.

Since the simulations only “look forward” along the RNs, here starting from X , it suffices to take the sub-interpretation \mathcal{P} of $\wp(\mathcal{I})$ consisting of all elements reachable from X — then the MMSCD $X^{\mathcal{I}}$ is already equivalent to $\exists^{\text{sim}}(\mathcal{P}, X)$.

It is interesting to remark that Lemma IX yields $C^{\wp(\mathcal{I})} = \wp(C^{\mathcal{I}})$ for each \mathcal{EL}_{si} CD C .

3.2 Axiomatization of CIs by means of FCA

The MMSCDs allow us to restrict attention to CIs of a particular form, viz. the set $\{C \sqsubseteq C^{\mathcal{I}\mathcal{I}} \mid C \text{ is an } \mathcal{EL}_{\text{si}} \text{ CD}\}$ would already be a CI base if it was finite. To see this, consider a CI $C \sqsubseteq D$ satisfied in \mathcal{I} . Then $C^{\mathcal{I}\mathcal{I}} \sqsubseteq^{\emptyset} D$ by (M2) and thus $C \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ entails $C \sqsubseteq D$. The Galois property (G7) further ensures that every CI $C \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ is satisfied in \mathcal{I} .

Each MMSCD $X^{\mathcal{I}}$ is either \perp or, according to Proposition 8, a conjunction of CNs and existential restrictions $\exists r.Y^{\mathcal{I}}$. For this reason, we let \mathbf{M} consist of \perp , all CNs, and all $\exists r.Y^{\mathcal{I}}$ where r is a RN and Y is a non-empty subset of $\text{Dom}(\mathcal{I})$.

$$\mathbf{M} := \{\perp\} \cup \{A \mid A \text{ is a CN}\}$$

$$\cup \{\exists r.X^{\mathcal{I}} \mid r \text{ is a RN and } X \subseteq \text{Dom}(\mathcal{I}), X \neq \emptyset\}$$

This definition is up to equivalence, i.e. if $Y^{\mathcal{I}} \equiv^{\emptyset} Z^{\mathcal{I}}$ for two subsets $Y, Z \subseteq \text{Dom}(\mathcal{I})$, then it suffices that \mathbf{M} contains the attributes $\exists r.Y^{\mathcal{I}}$ for all RNs r .

Due to our choice of \mathbf{M} , we can now represent the conclusion $C^{\mathcal{I}\mathcal{I}}$ of any CI $C \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ as a conjunction of atoms in \mathbf{M} , but this is not always possible for the premise C . We instead use the *partial closure* $C^{[\mathcal{I}\mathcal{I}]}$ which is closed everywhere above the root: $\perp^{[\mathcal{I}\mathcal{I}]} := \perp$ and, if $C \neq \perp$, then

$$\begin{aligned} C^{[\mathcal{I}\mathcal{I}]} &:= \prod \{A \mid A \in \text{Conj}(C)\} \\ &\quad \prod \prod \{\exists r.D^{\mathcal{I}\mathcal{I}} \mid \exists r.D \in \text{Conj}(C)\}. \end{aligned}$$

All top-level conjuncts of $C^{[\mathcal{I}\mathcal{I}]}$ are contained in \mathbf{M} .

There are only finitely many CIs of the form $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ since their premises and conclusions are conjunctions over the finite set \mathbf{M} . We will show that the TBox consisting of all these CIs is already a CI base. We therefore use the following slightly modified notion from (Baader, Distel, 2008).

Definition X. Given sets \mathcal{P} , \mathcal{Q} of CDs and a function $f: \mathcal{P} \rightarrow \mathcal{Q}$, we say that f *dominates* \mathcal{P} if $P \sqsubseteq^{\emptyset} f(P)$ and $P^{\mathcal{I}} = f(P)^{\mathcal{I}}$ for each $P \in \mathcal{P}$. If the function f is irrelevant, then we also say that \mathcal{Q} *dominates* \mathcal{P} .

Lemma XI. *If \mathcal{B} is a complete set of CIs that has model \mathcal{I} and f dominates $\{C \mid C \sqsubseteq D \in \mathcal{B}\}$, then also the set $\{f(C) \sqsubseteq D \mid C \sqsubseteq D \in \mathcal{B}\}$ is complete and has model \mathcal{I} .*

Proof. Let $\mathcal{B}_f := \{f(C) \sqsubseteq D \mid C \sqsubseteq D \in \mathcal{B}\}$ and consider a CI $C \sqsubseteq D$ in \mathcal{B} . Since $C \sqsubseteq^{\emptyset} f(C)$ and \mathcal{B}_f contains $f(C) \sqsubseteq D$, it follows that \mathcal{B}_f entails $C \sqsubseteq D$. Thus, \mathcal{B}_f is complete since it entails a complete set.

Next, consider a CI $f(C) \sqsubseteq D$ in \mathcal{B}_f . Since \mathcal{I} is a model of \mathcal{B} and the CI $C \sqsubseteq D$ is in \mathcal{B} , we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. With $C^{\mathcal{I}} = f(C)^{\mathcal{I}}$ we conclude that \mathcal{I} satisfies $f(C) \sqsubseteq D$. \square

Lemma 9. $\{C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}} \mid C \text{ is an } \mathcal{EL}_{\text{si}} \text{ CD}\}$ is a CI base.

Proof. The above TBox is denoted by \mathcal{B} . We first show that \mathcal{I} satisfies each CI $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ in \mathcal{B} . Recall that \mathcal{I} satisfies $C \sqsubseteq C^{\mathcal{I}\mathcal{I}}$. By applying (G6) to the fillers of the ERs we infer that $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq^{\emptyset} C$ and thus $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C$ is satisfied in \mathcal{I} . We conclude that \mathcal{I} satisfies $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}}$.

It remains to prove that \mathcal{B} is complete. To this end, we show that \mathcal{B} entails the set $\{C \sqsubseteq C^{\mathcal{I}\mathcal{I}} \mid C \text{ is an } \mathcal{EL} \text{ CD}\}$, which is complete since according to Lemma II all \mathcal{EL} CDs dominate all \mathcal{EL}_{si} CDs (Baader, Distel, 2008).

We prove by induction on C that \mathcal{B} entails each CI $C \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ where C is an \mathcal{EL} CD. The induction base where C has role depth³ zero is trivial since then $C = C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq^{\mathcal{B}} C^{\mathcal{I}\mathcal{I}}$. Now assume that $\text{rd}(C) > 0$. Since $\text{rd}(D) < \text{rd}(C)$ for each $\exists r.D \in \text{Conj}(C)$, the induction hypothesis yields that C is subsumed by $\prod\{A \mid A \in \text{Conj}(C)\} \sqcap \prod\{\exists r.D^{\mathcal{I}\mathcal{I}} \mid \exists r.D \in \text{Conj}(C)\}$ w.r.t. \mathcal{B} . The latter equals $C^{[\mathcal{I}\mathcal{I}]}$ and is thus subsumed by $C^{\mathcal{I}\mathcal{I}}$ w.r.t. \mathcal{B} . \square

Since the latter CI base consists of CIs between conjunctions over \mathbf{M} , Lemma 6 implies that we can obtain other, usually smaller CI bases by rewriting an implication base \mathcal{B} of the induced context $\mathbb{K}_{\mathcal{I}}$ into the TBox $\prod\mathcal{B}$. We will prove this in the following.

Moreover, we take the TBox \mathcal{T} into account by transforming it into the set $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ consisting of the implications

- $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]}) \rightarrow \{E \mid E \in \mathbf{M} \text{ and } C \sqsubseteq^{\mathcal{T}} E\}$ for each CI $C \sqsubseteq D$ in \mathcal{T}
- $\{E\} \rightarrow \{F\}$ for each two $E, F \in \mathbf{M}$ with $E \sqsubseteq^{\emptyset} F$.

The induced context $\mathbb{K}_{\mathcal{I}}$ is a model of $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ as \mathcal{I} is a model of \mathcal{T} .

Lemma XII. *If \mathcal{B} is an implication set over \mathbf{M} with $\mathcal{B} \cup \mathcal{L}_{\mathcal{I},\mathcal{T}} \models \mathbf{C} \rightarrow \mathbf{D}$, then $(\prod\mathcal{B}) \cup \mathcal{T} \models \prod\mathbf{C} \sqsubseteq \prod\mathbf{D}$.*

Proof. Let \mathcal{J} be a model of $(\prod\mathcal{B}) \cup \mathcal{T}$. We define the formal context $\mathbb{J} := (\text{Dom}(\mathcal{J}), \mathbf{M}, J)$ with incidence $(x, C) \in J$ iff. $x \in C^{\mathcal{J}}$. Note that \mathbb{J} and $\mathbb{K}_{\mathcal{I}}$ have the same attribute set.

³The *role depth* of an \mathcal{EL} CD C is the maximal number of nestings of ERs in C , denoted by $\text{rd}(C)$. In particular, $\text{rd}(C) = 0$ iff. C is \perp , \top , or a conjunction of CNs iff. C does not contain any ER.

For each subset $\mathbf{U} \subseteq \mathbf{M}$ we have $(\prod\mathbf{U})^{\mathcal{J}} = \mathbf{U}^{\mathcal{J}}$. Thus \mathcal{J} satisfies a CI $\prod\mathbf{U} \sqsubseteq \prod\mathbf{V}$ iff. \mathbb{J} satisfies the implication $\mathbf{U} \rightarrow \mathbf{V}$. We conclude that \mathbb{J} is a model of \mathcal{B} .

We show that \mathbb{J} is also a model of $\mathcal{L}_{\mathcal{I},\mathcal{T}}$. Consider an implication $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]} \rightarrow \{E \mid E \in \mathbf{M} \text{ and } C \sqsubseteq^{\mathcal{T}} E\})$ where $C \sqsubseteq D \in \mathcal{T}$. Since $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq^{\emptyset} C$ and \mathcal{J} is a model of \mathcal{T} , the CI $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq E$ is satisfied in \mathcal{J} for each atom E in the conclusion, and thus the implication is satisfied in \mathbb{J} . Furthermore, if $E \sqsubseteq^{\emptyset} F$, then \mathcal{J} satisfies the CI $E \sqsubseteq F$ and thus \mathbb{J} satisfies the implication $\{E\} \rightarrow \{F\}$.

It follows that $\mathbf{C} \rightarrow \mathbf{D}$ is satisfied in \mathbb{J} , and thus $\prod\mathbf{C} \sqsubseteq \prod\mathbf{D}$ is satisfied in \mathcal{J} . \square

Proposition XIII. *If \mathcal{B} is an implication base of $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I},\mathcal{T}}$, then the TBox $\prod\mathcal{B}$ is a CI base of \mathcal{I} relative to \mathcal{T} .*

Proof. Recall from Lemma 9 that \mathcal{I} satisfies all CIs $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ where C is an \mathcal{EL}_{si} CD. With Lemma 6 we infer that all implications $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]}) \rightarrow \text{Conj}(C^{\mathcal{I}\mathcal{I}})$ are satisfied in the induced context $\mathbb{K}_{\mathcal{I}}$. Since $\mathcal{B} \cup \mathcal{L}_{\mathcal{I},\mathcal{T}}$ is complete for $\mathbb{K}_{\mathcal{I}}$, this union entails all these implications. According to Lemma XII the union $(\prod\mathcal{B}) \cup \mathcal{T}$ entails all CIs $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ and is thus complete for \mathcal{I} by Lemma 9. That all CIs in $\prod\mathcal{B}$ are satisfied in \mathcal{I} follows from Lemma 6. \square

As next step, we show that also in the opposite direction every CI base of \mathcal{I} relative to \mathcal{T} can be transformed into an implication base of the induced context $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I},\mathcal{T}}$, provided that it is in a certain normal form. This transformation helps us with proving that a CI base $\prod\mathcal{B}$ obtained from a minimal implication base \mathcal{B} is also minimal.

Lemma XIV. *For each subset \mathbf{X} of \mathbf{M} and for each CD C , if $\prod\mathbf{X} \sqsubseteq^{\emptyset} C$, then $\prod\mathbf{X} \sqsubseteq^{\emptyset} C^{[\mathcal{I}\mathcal{I}]}$.*

Proof. Recall that C and its partial closure $C^{[\mathcal{I}\mathcal{I}]}$ have the same CNs in the top-level conjunction. It is thus only interesting to consider ERs. So let $\exists r.D^{\mathcal{I}\mathcal{I}} \in \text{Conj}(C^{[\mathcal{I}\mathcal{I}]})$, i.e. we have $\exists r.D \in \text{Conj}(C)$. Since $\prod\mathbf{X} \sqsubseteq^{\emptyset} \exists r.D$, there is $\exists r.E^{\mathcal{I}\mathcal{I}} \in \mathbf{X}$ with $E^{\mathcal{I}\mathcal{I}} \sqsubseteq^{\emptyset} D$. It follows that $E^{\mathcal{I}\mathcal{I}} \sqsubseteq^{\emptyset} D^{\mathcal{I}\mathcal{I}}$ by Properties (G1) and (G2), and thus $\prod\mathbf{X} \sqsubseteq^{\emptyset} \exists r.D^{\mathcal{I}\mathcal{I}}$. \square

Lemma XV. *If \mathcal{S} is a CI base of \mathcal{I} relative to \mathcal{T} and every CI in \mathcal{S} has the form $C \sqsubseteq D^{[\mathcal{I}\mathcal{I}]}$, then $\mathcal{B}_{\mathcal{S}} := \{\text{Conj}(C^{[\mathcal{I}\mathcal{I}]} \rightarrow \text{Conj}(D^{\mathcal{I}\mathcal{I}}) \mid C \sqsubseteq D \in \mathcal{S}\}$ is an implication base of $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I},\mathcal{T}}$.*

Proof. Note that $\prod\mathcal{B}_{\mathcal{S}} = \{C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}} \mid C \sqsubseteq D \in \mathcal{S}\}$. In general, \mathcal{S} and $\prod\mathcal{B}_{\mathcal{S}}$ do not entail each other, but due to Lemma XIV $\prod\mathcal{B}_{\mathcal{S}}$ allows us to draw the same consequences at the root of a conjunction $\prod\mathbf{X}$ where $\mathbf{X} \subseteq \mathbf{M}$ (and sometimes even more, since we replace the conclusion D of each CI in \mathcal{S} by $C^{\mathcal{I}\mathcal{I}}$, which is subsumed by D by (G1)). Further note that $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ entails the implication $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]}) \rightarrow \text{Conj}(D^{[\mathcal{I}\mathcal{I}]})$ for each $C \sqsubseteq D^{[\mathcal{I}\mathcal{I}]}$ in \mathcal{T} .

First, we show that $\mathbb{K}_{\mathcal{I}}$ satisfies all implications in $\mathcal{B}_{\mathcal{S}}$. Recall from the proof of Lemma 9 that \mathcal{I} satisfies each CI $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq C^{\mathcal{I}\mathcal{I}}$. By Lemma 6, $\mathbb{K}_{\mathcal{I}}$ satisfies each implication $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]} \rightarrow \text{Conj}(C^{\mathcal{I}\mathcal{I}}))$.

Next, we show that $\mathcal{B}_S \cup \mathcal{L}_{\mathcal{I}, \mathcal{T}}$ is complete for $\mathbb{K}_{\mathcal{I}}$. Let $\mathbf{X} \rightarrow \mathbf{Y}$ be satisfied in $\mathbb{K}_{\mathcal{I}}$. Then $\sqcap \mathbf{X} \sqsubseteq \sqcap \mathbf{Y}$ is satisfied in \mathcal{I} and thus entailed by $\mathcal{S} \cup \mathcal{T}$. We consider the TBox $\mathcal{S}' := \{C \sqsubseteq C^{II} \mid C \sqsubseteq D \in \mathcal{S}\}$, which entails \mathcal{S} by (G1). It follows that $\mathcal{S}' \cup \mathcal{T}$ entails $\sqcap \mathbf{X} \sqsubseteq \sqcap \mathbf{Y}$. According to Proposition 3, saturating $\sqcap \mathbf{X}$ by means of the \sqsubseteq_+ -Rule and the \sqsubseteq_{\perp} -Rule for $\mathcal{S}' \cup \mathcal{T}$ yields a CD that is subsumed by $\sqcap \mathbf{Y}$.

A crucial observation is that, for each conjunction $\sqcap \mathbf{X}$ with $\mathbf{X} \subseteq \mathbf{M}$, one rule application yields again a conjunction $\sqcap \mathbf{X}'$ for some $\mathbf{X}' \subseteq \mathbf{M}$. We illustrate this as follows. Since \mathcal{I} is a model of $\mathcal{S}' \cup \mathcal{T}$ and each filler of an ER in \mathbf{M} is a MMSCD of \mathcal{I} , a rule application is only possible the root of $\sqcap \mathbf{X}$.

- If the \sqsubseteq_+ -Rule is applied for a CI $C \sqsubseteq C^{II}$ in \mathcal{S}' , then we have $\sqcap \mathbf{X} \sqsubseteq^{\emptyset} C$ and obtain the CD $\sqcap \mathbf{X} \sqcap C^{II}$. Since \mathbf{M} contains all top-level conjuncts of C^{II} , the obtained CD equals $\sqcap \mathbf{X}'$ where $\mathbf{X}' := \mathbf{X} \cup \text{Conj}(C^{II})$.
- If the \sqsubseteq_+ -Rule is applied for a CI $C \sqsubseteq D^{[II]}$ in \mathcal{T} , then we have $\sqcap \mathbf{X} \sqsubseteq^{\emptyset} C$ and obtain the CD $\sqcap \mathbf{X} \sqcap D^{[II]}$. Since \mathbf{M} contains all top-level conjuncts of $D^{[II]}$, the obtained CD equals $\sqcap \mathbf{X}'$ where $\mathbf{X}' := \mathbf{X} \cup \text{Conj}(D^{[II]})$.
- If the \sqsubseteq_{\perp} -Rule is applied for a CI $C \sqsubseteq C^{II}$ in \mathcal{S}' with $C^{II} = \perp$, then we have $\sqcap \mathbf{X} \sqsubseteq^{\emptyset} C$ and obtain the CD \perp , which equals $\sqcap \mathbf{X}'$ where $\mathbf{X}' := \{\perp\}$. Further rule applications are then not possible.
- If the \sqsubseteq_{\perp} -Rule is applied for a CI $C \sqsubseteq D^{[II]}$ in \mathcal{T} with $D^{[II]} = \perp$, then we have $\sqcap \mathbf{X} \sqsubseteq^{\emptyset} C$ and obtain the CD \perp , which equals $\sqcap \mathbf{X}'$ where $\mathbf{X}' := \{\perp\}$. Further rule applications are then not possible.

Now recall that saturating $\sqcap \mathbf{X}$ w.r.t. $\mathcal{S}' \cup \mathcal{T}$ yields a CD that is subsumed by $\sqcap \mathbf{Y}$. We infer that there is a sequence $\mathbf{X}_0, \dots, \mathbf{X}_n$ of subsets of \mathbf{M} such that $\mathbf{X}_0 := \mathbf{X}$, each \mathbf{X}_{i+1} is obtained from \mathbf{X}_i by one rule application, and $\sqcap \mathbf{X}_n \sqsubseteq^{\emptyset} \sqcap \mathbf{Y}$. We define the sequence $\mathbf{X}'_0, \dots, \mathbf{X}'_n$ of subsets of \mathbf{M} where each \mathbf{X}'_i is the closure of \mathbf{X}_i under all implications $\{E\} \rightarrow \{F\}$ with $E \sqsubseteq^{\emptyset} F$. Then $\sqcap \mathbf{X}_i \sqsubseteq^{\emptyset} C$ iff. $\sqcap \mathbf{X}_i \sqsubseteq^{\emptyset} C^{[II]}$ iff. $\text{Conj}(C^{[II]}) \subseteq \mathbf{X}'_i$, where the first equivalence holds by Lemma XIV and the second by definition of \mathbf{X}'_i . This means that the implication $\text{Conj}(C^{[II]}) \rightarrow \text{Conj}(C^{II})$ contained in \mathcal{B}_S or the implication $\text{Conj}(C^{[II]}) \rightarrow \text{Conj}(D^{[II]})$ entailed by $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ can be used to obtain \mathbf{X}'_{i+1} from \mathbf{X}'_i , i.e. $\mathbf{X}'_{i+1} \subseteq (\mathbf{X}'_i)^{\mathcal{B}_S \cup \mathcal{L}_{\mathcal{I}, \mathcal{T}}}$. By induction we conclude that $\mathbf{Y} \subseteq \mathbf{X}'_n \subseteq \mathbf{X}^{\mathcal{B}_S \cup \mathcal{L}_{\mathcal{I}, \mathcal{T}}}$, i.e. $\mathcal{B}_S \cup \mathcal{L}_{\mathcal{I}, \mathcal{T}} \models \mathbf{X} \rightarrow \mathbf{Y}$. \square

Proposition XVI. *In addition to Proposition XIII: if \mathcal{B} contains the fewest implications among all implication bases of $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ and if further each CI in \mathcal{T} has the form $C \sqsubseteq D^{[II]}$, then also the CI base $\sqcap \mathcal{B}$ contains the fewest CIs among all CI bases of \mathcal{I} relative to \mathcal{T} .*

Proof. Assume that \mathcal{B} is minimal, and consider another CI base \mathcal{S} of \mathcal{I} . Lemma XV yields that \mathcal{B}_S is an implication base of $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$. Since $|\mathcal{B}_S| \leq |\mathcal{S}|$, minimality of \mathcal{B} implies that $|\mathcal{B}| \leq |\mathcal{B}_S|$. Since $|\sqcap \mathcal{B}| \leq |\mathcal{B}|$, it follows

that $|\sqcap \mathcal{B}| \leq |\mathcal{S}|$. Since this holds for all CI bases \mathcal{S} , we conclude that $\sqcap \mathcal{B}$ is minimal. \square

As last step, we analyze the computational complexity of computing a CI base. The attribute set \mathbf{M} of the induced context $\mathbb{K}_{\mathcal{I}}$ has exponential size, and thus an implication base of $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ could, in principle, contain a double-exponential number of implications. We use the below lemma to show that the canonical implication base $\text{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}})$ is not so large and can be computed in exponential time.

Lemma XVII. *Let $\mathbb{K} := (G, M, I)$ be a formal context and \mathcal{L} be a set of implications over M . Then the canonical implication base of \mathbb{K} relative to \mathcal{L} can be computed in time exponential in G , polynomial in M , and polynomial in \mathcal{L} .*

Proof. We employ the algorithm NextClosures (Kriegel, Borchmann, 2017) to show the upper bound on the time complexity. Our background implication set \mathcal{L} is there denoted by \mathcal{B} , while the canonical implication base under construction is there denoted by \mathcal{L} . Furthermore, we are in the case where all background implications are satisfied in \mathbb{K} , and thus $C^{II \vee \mathcal{B}} = C^{II}$ for all subsets $C \subseteq M$.

The algorithm NextClosures maintains a candidate set \mathbf{C} that initially contains only the set $\emptyset^{\mathcal{B}}$. Further candidates are only added in Line 7, and the candidates are only updated in Line 9. We first show that the number of fresh candidates (added in Line 7) is exponential in G and polynomial in M . To this end, we exploit the close relationship between extents and intents: if A is an extent, then A^I is an intent, and every intent is of this form (i.e. if B is an intent, then there is an extent A such that $A^I = B$). Since every extent is a subset of G , we conclude that the number of intents is exponential in G . Whenever NextClosures recognizes an intent C^{II} , then it adds the fresh candidates $C^{II} \cup \{m\}$ for all attributes $m \in M \setminus C^{II}$. It follows that the number of fresh candidates is exponential in G and polynomial in M .

Since every pseudo-intent (and thus every implication in the canonical implication base) is obtained from a candidate, the size of the canonical implication base (denoted by \mathcal{L} within NextClosures) is also exponential in G and polynomial in M .

Each candidate (and its updated versions) is considered at most $|M|$ times in the outer for-loop in Lines 1–10. When a candidate is processed, then the closures $C^{\mathcal{L}^* \vee \mathcal{B}}$ and C^{II} are computed and checked for equality with C , which needs polynomial time w.r.t. \mathcal{L} and \mathcal{B} and w.r.t. G and M , respectively.

In summary, NextClosures runs in time exponential in G and polynomial in M and \mathcal{L} and returns the canonical implication base of \mathbb{K} w.r.t. \mathcal{L} . \square

We obtain our first main result by putting all lemmas together.

Theorem 10. *For each finite interpretation \mathcal{I} and each $\mathcal{E}\mathcal{L}_{\text{SI}}^{\perp}$ TBox \mathcal{T} of which \mathcal{I} is a model, the TBox $\text{Can}(\mathcal{I}, \mathcal{T}) := \sqcap \text{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}})$ is a CI base of \mathcal{I} relative to \mathcal{T} . It is called canonical CI base and can be computed in time that is exponential in $\text{Dom}(\mathcal{I})$ and polynomial in \mathcal{T} . If all CIs in \mathcal{T} have*

The implication set $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ consists of all $\{E\} \rightarrow \{F\}$ where $E \sqsubseteq^\emptyset F$ and of $\{A\} \rightarrow \{\exists r. \{w, x, y, z\}^{\mathcal{I}}\}$. Note that the latter evaluates to $\{A\} \rightarrow \{\exists r. \top\}$ and thus does not fully capture the CI $A \sqsubseteq \exists r. B$. By transforming the canonical implication base of $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$, we obtain the canonical CI base of \mathcal{I} relative to \mathcal{T} with the following CIs:

- $B \sqsubseteq C$
- $A \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \exists r. \{z\}^{\mathcal{I}} \sqcap \exists r. \{y, z\}^{\mathcal{I}}$
- $C \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \exists r. \{z\}^{\mathcal{I}} \sqcap \exists r. \{y, z\}^{\mathcal{I}}$
- $\exists r. \{y, z\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \exists r. \{z\}^{\mathcal{I}}$
- $\exists r. \{w, x, y\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \exists r. \{x\}^{\mathcal{I}} \sqcap \exists r. \{x, y\}^{\mathcal{I}}$
- $A \sqcap C \sqcap \exists r. \{z\}^{\mathcal{I}} \sqcap \exists r. \{y, z\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \perp$
- $B \sqcap C \sqcap \exists r. \{z\}^{\mathcal{I}} \sqcap \exists r. \{y, z\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \perp$
- $\exists r. \{w\}^{\mathcal{I}} \sqcap \exists r. \{x\}^{\mathcal{I}} \sqcap \exists r. \{x, y\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \perp$
- $\exists r. \{x\}^{\mathcal{I}} \sqcap \exists r. \{z\}^{\mathcal{I}} \sqcap \exists r. \{x, y\}^{\mathcal{I}} \sqcap \exists r. \{y, z\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y\}^{\mathcal{I}} \sqcap \exists r. \{w, x, y, z\}^{\mathcal{I}} \sqsubseteq \perp$

Specifically the second CI is superfluous as it can be deduced from the others. To see this, first note that it can be simplified to $A \sqcap \exists r. \top \sqsubseteq \exists r. (B \sqcap C)$. Now, the premise $A \sqcap \exists r. \top$ is subsumed by $\exists r. B$ (since \mathcal{T} contains $A \sqsubseteq \exists r. B$) and thus subsumed by $\exists r. (B \sqcap C)$ (since the CI base contains $B \sqsubseteq C$), which is the conclusion of the second CI.

To obtain a minimal CI base, we could replace the conclusion of the CI $A \sqsubseteq \exists r. B$ in \mathcal{T} with $(\exists r. B)^{[\mathcal{I}]} = \exists r. (B \sqcap C)$. Then, the implication set $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ would contain $\{A\} \rightarrow \{\exists r. (B \sqcap C)\}$ in place of $\{A\} \rightarrow \{\exists r. \top\}$ and so the replaced CI could be fully captured.

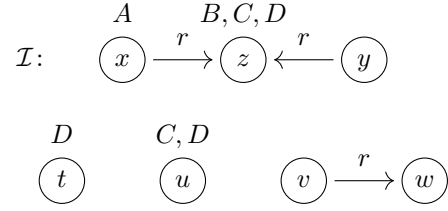
One might be tempted to think that it were sufficient to add all top-level conjuncts of CIs in \mathcal{T} to the attribute set \mathbf{M} in order to obtain a minimal CI base since \mathcal{T} can then be fully captured by implications. The below example shows that this is wrong.

Example XVIII. We use the same interpretation \mathcal{I} but where z additionally is an instance of the CN D , and the TBox \mathcal{T} is extended with the CI $B \sqsubseteq C$. We would add to \mathbf{M} the additional attribute $\exists r. B$ and can therefore fully capture $A \sqsubseteq \exists r. B$ as $\{A\} \rightarrow \{\exists r. B\}$ and $B \sqsubseteq C$ as $\{B\} \rightarrow \{C\}$. Compared to the previous example, $\mathbb{K}_{\mathcal{I}}$ would then contain the additional attributes D and $\exists r. B$, and the MMSCDs are the same except that $B \sqcap C$ is replaced by $B \sqcap C \sqcap D$.

The CI base would then contain the CI $B \sqcap C \sqsubseteq D$, but also the superfluous CI $A \sqcap \exists r. B \sqsubseteq \exists r. (B \sqcap C \sqcap D)$, i.e. the base is not minimal. The problem here is that \mathcal{T} entails $A \sqsubseteq \exists r. (B \sqcap C)$ but this cannot be captured by an implication over \mathbf{M} (neither directly nor entailed by other implications).

In the last example we illustrate that a minimal CI base is neither guaranteed when we use the “saturated atoms” $\exists r. C^{\mathcal{T}}$ for every ER $\exists r. C$ occurring as a top-level conjunct in \mathcal{T} , where $C^{\mathcal{T}}$ is defined in Section 2.3.

Example XIX. As interpretation \mathcal{I} we take the following.



We further consider the TBox $\mathcal{T} := \{A \sqsubseteq \exists r. B, B \sqsubseteq C\}$. Now we would add the “saturated atom” $\exists r. (B \sqcap C)$ to the attribute set \mathbf{M} , so that we can take the implications $\{A\} \rightarrow \{\exists r. (B \sqcap C)\}$ and $\{B\} \rightarrow \{C\}$ as background knowledge. The implication set $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ additionally contains all $\{E\} \rightarrow \{F\}$ where $E \sqsubseteq^\emptyset F$.

It is easy to verify that the implication base of the induced context $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ contains $\{C\} \rightarrow \{D\}$ and $\{\exists r. (B \sqcap C)\} \rightarrow \{\exists r. (B \sqcap C \sqcap D), \exists r. (C \sqcap D), \exists r. D\}$, which are transformed to the CIs $C \sqsubseteq D$ and $\exists r. (B \sqcap C) \sqsubseteq \exists r. (B \sqcap C \sqcap D)$. The latter is superfluous.

The problem here is that the additional atoms can be used in further implications that need not follow from the others in implication semantics (which does not look into the ERs) but that would be superfluous with the DL semantics. Proposition XVI shows that this cannot happen if all fillers in ERs are MMSCDs.

Novel Contributions. To differentiate our novel contributions from results already shown in the literature, we provide the following dissection. To begin with, the paper (Baader, Distel, 2008) shows in Theorem 15 that each finite interpretation has a finite CI base. FCA is not used explicitly therein, but key notions from FCA are translated into the DL setting, e.g. the model-based most specific concepts. Moreover, the attribute set \mathbf{M} is hidden in the proof of that Theorem 15 and is made explicit in the follow-up paper (Baader, Distel, 2009), see Definition 4. There, the computation of the CI base is done by means of FCA, see Theorem 4, where background implications are used to avoid the axiomatization of tautological CIs. That this canonical CI base contains the fewest number of CIs among all CI bases is shown in the thesis (Distel, 2011).

Our first significant contribution is to show that the canonical CI base is computable in at most exponential time. Second, we enable support for an existing TBox \mathcal{T} relative to which the input interpretation \mathcal{I} is axiomatized. This is not trivial. The key was to find a background implication set that can be constructed from \mathcal{T} in at most exponential time, see Appendix A for a remark. To sum up, Theorem 10 without a TBox \mathcal{T} and without the complexity result was already known in the literature. Analogously for Theorem 18, which expands on (Borchmann, Distel, Kriegel, 2016).

Third, we carefully revised and extended the technical argumentation leading to Theorem 10. Instead of $\mathcal{EL}_{\text{gfp}}^{\perp}$ we used the equi-expressive DL $\mathcal{EL}_{\text{si}}^{\perp}$ as it is easier to handle. We introduced the powering $\wp(\mathcal{I})$ as a joint, smaller representation of all MMSCDs; before it was necessary to compute products of different arities. (In a domain of cardinality n , the size of an MMSCD of k objects is at most $\sum_{i=1}^k \binom{n}{i}$ with the powering vs. n^k with products.) Moreover, we in-

roduced the attribute set M as needed and not let it “fall from the sky.” More specifically in this Section 3.2, all stated results and proofs but Lemma XV are revised versions from the above three references and that are extended to deal with the given TBox \mathcal{T} .

Last, we emphasize that the specific role of FCA in the result stated in Theorem 10 is two-fold: on the one hand, FCA guarantees that the canonical CI base contains the fewest possible number of CIs, especially no tautologies or CIs entailed by \mathcal{T} ; on the other hand, we can employ FCA algorithms to actually compute CI bases in practise. In contrast, the CI base \mathcal{B} in (Baader, Distel, 2008) is constructed without FCA and is of exponential size (but this was not mentioned there). With further efforts one could show that \mathcal{B} is computable in exponential time, though is not minimal and might contain tautologies or CIs entailed by \mathcal{T} , i.e. its practical computability and usability is unclear. In particular, a minimal CI base is preferred if reasoning performance with the constructed ontology is crucial.

3.3 Rewriting the CI Base

The canonical CI base contains $\mathcal{EL}_{\text{si}}^{\perp}$ CIs. We rewrite it into \mathcal{EL} to gain support by state-of-the-art reasoners, such as ELK. We subdivide the rewriting into two steps. First, since the set of all \mathcal{EL}_{si} CDs is dominated by the set of all \mathcal{EL} CDs (Baader, Distel, 2008), Lemma XI allows us to replace the premises in $\text{Can}(\mathcal{I}, \mathcal{T})$ by suitable \mathcal{EL} CDs. Minimality is preserved since such a replacement does not change the number of CIs in the base. In particular, we can replace, in every conjunction $\bigwedge C$ that occurs as a premise, each $\exists r.X^{\mathcal{I}}$ by $\exists r.(X^{\mathcal{I}} \upharpoonright_n)$ with the following choices for n .

- n is minimal such that $(X^{\mathcal{I}} \upharpoonright_n)^{\mathcal{I}} = X^{\mathcal{II}}$, which can be determined by trying non-negative integers in ascending order and picking the first for which the MMSCD $X^{\mathcal{I}}$ and its unfolding $X^{\mathcal{I}} \upharpoonright_n$ have the same extension in \mathcal{I} . Lemma II ensures that such a minimal n exists.
- $n := 2^{|\text{Dom}(\mathcal{I})|} \cdot |\text{Dom}(\mathcal{I})| + 1$ (Baader, Distel, 2008)
- n is obtained from the MVF measure based on lengths of simple paths in \mathcal{I} or powers of \mathcal{I} , seen as graphs (Guimarães, Ozaki, Persia, Sertkaya, 2021).

Next, we devise a mechanism to replicate the cyclic structures within the conclusions directly in the CI base, namely by means of variables (auxiliary CNs) that can be used in place of CNs.

Definition XX. A TBox with variables (vTBox) $\exists \mathbb{X}.\mathcal{T}$ consists of a finite set \mathbb{X} of *concept variables* that is disjoint with the signature, and of a finite set \mathcal{T} of \mathcal{EL}^{\perp} CIs in which the variables can be used in place of CNs. An interpretation \mathcal{I} is a *model* of $\exists \mathbb{X}.\mathcal{T}$ if there is a variable assignment $\mathcal{Z}: \mathbb{X} \rightarrow \wp(\text{Dom}(\mathcal{I}))$ such that the augmented interpretation $\mathcal{I}[\mathcal{Z}]$ is a model of \mathcal{T} , where $\mathcal{I}[\mathcal{Z}]$ equals \mathcal{I} but its function $\cdot^{\mathcal{I}[\mathcal{Z}]}$ additionally maps each variable $X \in \mathbb{X}$ to $\mathcal{Z}(X)$.

No new procedures or implementations are needed to decide if a vTBox entails a CI — we just discard the quantifier.

Lemma XXI. $\exists \mathbb{X}.\mathcal{T} \models C \sqsubseteq D$ iff. $\mathcal{T} \models C \sqsubseteq D$

Proof. Assume $\exists \mathbb{X}.\mathcal{T} \models C \sqsubseteq D$ and that \mathcal{I} is a model of \mathcal{T} (over the extended signature that contains all variables as additional CNs). Then the restriction of \mathcal{I} to the signature, denoted as \mathcal{I}' , is a model of $\exists \mathbb{X}.\mathcal{T}$ since for the variable assignment $\mathcal{Z}: X \mapsto X^{\mathcal{I}}$ the augmented interpretation $\mathcal{I}'[\mathcal{Z}]$ is equal to \mathcal{I} and is thus a model of \mathcal{T} . We conclude that the CI $C \sqsubseteq D$ is satisfied in \mathcal{I}' . Since the CDs C and D do not contain variables, and since the interpretation functions of \mathcal{I} and \mathcal{I}' coincide on the signature, it follows that $C \sqsubseteq D$ is also satisfied in \mathcal{I} .

Conversely, let $\mathcal{T} \models C \sqsubseteq D$ and consider a model \mathcal{I} of $\exists \mathbb{X}.\mathcal{T}$. Then $\mathcal{I}[\mathcal{Z}]$ is a model of \mathcal{T} for some variable assignment \mathcal{Z} , and we infer that $C \sqsubseteq D$ is satisfied in $\mathcal{I}[\mathcal{Z}]$. Since no variables occur in C or in D , it follows that $C \sqsubseteq D$ must already be satisfied in \mathcal{I} . \square

Proposition XXII. Every TBox consisting of CIs with \mathcal{EL} CDs as premises and $\mathcal{EL}_{\text{si}}^{\perp}$ CDs as conclusions can be rewritten into an equivalent vTBox in linear time.

Proof. Since each $\mathcal{EL}_{\text{si}}^{\perp}$ CD is equivalent either to \perp or to a CD of the form $\exists^{\text{sim}}(\mathcal{D}, y)$, we can assume w.l.o.g. that all CIs in \mathcal{T} are either of the form $C \sqsubseteq \perp$ or of the form $C \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, y)$. We further assume w.l.o.g. that all the latter CDs $\exists^{\text{sim}}(\mathcal{D}, y)$ share the same interpretation \mathcal{D} (otherwise we define \mathcal{D} as the union of all these interpretations). Since each $\exists^{\text{sim}}(\mathcal{D}, y)$ is equivalent to $\exists^{\text{sim}}(\text{Red}(\mathcal{D}), [y]_{\approx})$, we can also assume w.l.o.g. that \mathcal{D} is weakly reduced (see Definition XXV). Last, we can remove from \mathcal{D} all elements that are not reachable from y for some CI $C \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, y)$ in \mathcal{T} .

We define the vTBox $\exists \mathbb{X}.\mathcal{T}'$ where $\mathbb{X} := \text{Dom}(\mathcal{D})$ and \mathcal{T}' consists of the following CIs:

1. $C \sqsubseteq \perp$ for each CI $C \sqsubseteq \perp$ in \mathcal{T} .
2. $C \sqsubseteq y$ for each CI $C \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, y)$ in \mathcal{T} .
3. $y \sqsubseteq A$ for each CN A and each $y \in A^{\mathcal{D}}$
4. $y \sqsubseteq \exists r.z$ for each RN r and each $(y, z) \in r^{\mathcal{D}}$

Instead of the latter two instructions, we could also add the CI $y \sqsubseteq \bigwedge \{A \mid y \in A^{\mathcal{D}}\} \sqcap \bigwedge \{\exists r.z \mid (y, z) \in r^{\mathcal{D}}\}$ to \mathcal{T}' for each $y \in \text{Dom}(\mathcal{D})$. Obviously, $\exists \mathbb{X}.\mathcal{T}'$ is constructed from \mathcal{T} by mere syntactic rewriting in linear time.

We show that \mathcal{T} and $\exists \mathbb{X}.\mathcal{T}'$ have the same models and are thus equivalent. Let \mathcal{I} be a model of \mathcal{T} . We define the variable assignment \mathcal{Z} by $\mathcal{Z}(y) := (\exists^{\text{sim}}(\mathcal{D}, y))^{\mathcal{I}}$ and show that $\mathcal{I}[\mathcal{Z}]$ is a model of \mathcal{T}' .

1. Assume that $C \sqsubseteq \perp$ is in \mathcal{T}' , i.e. the same CI is in \mathcal{T} and thus $C^{\mathcal{I}} = \perp^{\mathcal{I}} = \emptyset$. Since no concept variables occur in C , we have that $C^{\mathcal{I}[\mathcal{Z}]} = C^{\mathcal{I}}$. It follows that $C \sqsubseteq \perp$ is satisfied in $\mathcal{I}[\mathcal{Z}]$.
2. Consider a CI $C \sqsubseteq y$ in \mathcal{T}' , i.e. the CI $C \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, y)$ is in \mathcal{T} . Further let $x \in C^{\mathcal{I}[\mathcal{Z}]}$. Since no concept variables occur in C , we have that $C^{\mathcal{I}[\mathcal{Z}]} = C^{\mathcal{I}}$. With \mathcal{I} being a model of \mathcal{T} it follows that $x \in (\exists^{\text{sim}}(\mathcal{D}, y))^{\mathcal{I}}$. By definition of \mathcal{Z} , the latter shows that $x \in \mathcal{Z}(y) = y^{\mathcal{I}[\mathcal{Z}]}$.
3. Let $y \sqsubseteq A$ be in \mathcal{T}' , i.e. $y \in A^{\mathcal{D}}$, and further let $x \in y^{\mathcal{I}[\mathcal{Z}]}$ be in $\mathcal{I}[\mathcal{Z}]$. So there is a simulation from \mathcal{D} to \mathcal{I} containing (y, x) , and thus $y \in A^{\mathcal{D}}$ implies $x \in A^{\mathcal{I}} = A^{\mathcal{I}[\mathcal{Z}]}$.

4. Last, assume that \mathcal{T}' contains the CI $y \sqsubseteq \exists r.z$, i.e. $(y, z) \in r^{\mathcal{D}}$, and that $x \in y^{\mathcal{I}[\mathcal{Z}]} = \mathcal{Z}(y) = (\exists^{\text{sim}}(\mathcal{D}, y))^{\mathcal{I}}$. The latter implies that there is a simulation from \mathcal{D} to \mathcal{I} containing (y, x) . Thus, it follows from $(y, z) \in r^{\mathcal{D}}$ that there is x' such that $(x, x') \in r^{\mathcal{I}} = r^{\mathcal{I}[\mathcal{Z}]}$ and there is a simulation from \mathcal{D} to \mathcal{I} containing (z, x') , i.e. $x' \in \mathcal{Z}(z) = z^{\mathcal{I}[\mathcal{Z}]}$. We conclude that $x \in (\exists r.z)^{\mathcal{I}[\mathcal{Z}]}$.

Conversely, assume that \mathcal{I} is a model of $\exists \mathbb{X}.\mathcal{T}'$, i.e. there is a variable assignment \mathcal{Z} such that $\mathcal{I}[\mathcal{Z}]$ is a model of \mathcal{T}' . Each CI of the form $C \sqsubseteq \perp$ in \mathcal{T} is also contained in \mathcal{T}' , which implies that $C \sqsubseteq \perp$ is satisfied in $\mathcal{I}[\mathcal{Z}]$. Since C does not contain variables we have $C^{\mathcal{I}} = C^{\mathcal{I}[\mathcal{Z}]}$, and thus $C \sqsubseteq \perp$ is also satisfied in \mathcal{I} .

Further let $C \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, y)$ be a CI in \mathcal{T} where $x \in C^{\mathcal{I}}$. Then \mathcal{T}' contains the CI $C \sqsubseteq y$ and $x \in C^{\mathcal{I}[\mathcal{Z}]}$, which implies $x \in y^{\mathcal{I}[\mathcal{Z}]} = \mathcal{Z}(y)$. In order to show that $x \in (\exists^{\text{sim}}(\mathcal{D}, y))^{\mathcal{I}}$, we verify that the following relation \mathfrak{S} , which contains (y, x) since $x \in \mathcal{Z}(y)$, is a simulation from \mathcal{D} to \mathcal{I} .

$$\mathfrak{S} := \{ (v, u) \mid u \in \mathcal{Z}(v) \}$$

- (S1) Let $(v, u) \in \mathfrak{S}$, i.e. $u \in \mathcal{Z}(v)$, and further let $v \in A^{\mathcal{D}}$. Then \mathcal{T}' contains $v \sqsubseteq A$. Since $\mathcal{I}[\mathcal{Z}]$ is a model of \mathcal{T}' , we have that $\mathcal{Z}(v) \subseteq A^{\mathcal{I}}$. We conclude that $u \in A^{\mathcal{I}}$.
- (S2) Let $(v, u) \in \mathfrak{S}$, i.e. $u \in \mathcal{Z}(v)$, and further let $(v, v') \in r^{\mathcal{D}}$. Then \mathcal{T}' contains $v \sqsubseteq \exists r.v'$. Since $\mathcal{I}[\mathcal{Z}]$ is a model of \mathcal{T}' , we have that $\mathcal{Z}(v) \subseteq (\exists r.v')^{\mathcal{I}[\mathcal{Z}]}$, i.e. $(u, u') \in r^{\mathcal{I}}$ for some $u' \in \mathcal{Z}(v')$. The latter yields $(v', u') \in \mathfrak{S}$. \square

Last, we apply the rewriting approach to the conclusions in the canonical CI base. Assume that \mathcal{B} is obtained from $\text{Can}(\mathcal{I}, \mathcal{T})$ by replacing the premises as above. We obtain an \mathcal{EL} TBox \mathcal{B}' with auxiliary CNs that entails the same CIs as follows. For each CI $C \sqsubseteq \perp$ in \mathcal{B} , we add $C \sqsubseteq \perp$ to \mathcal{B}' . For each CI $C \sqsubseteq \prod \mathbf{D}$ in \mathcal{B} , we add the following CIs to \mathcal{B}' :

- $C \sqsubseteq \prod \{ A \mid A \in \mathbf{D} \} \sqcap \prod \{ \exists r.X \mid \exists r.X^{\mathcal{I}} \in \mathbf{D} \}$
- $Y \sqsubseteq \prod \{ A \mid Y \in A^{\wp(\mathcal{I})} \} \sqcap \prod \{ \exists r.Z \mid (Y, Z) \in r^{\wp(\mathcal{I})} \}$ for all Y reachable from any X in $\wp(\mathcal{I})$ with $\exists r.X^{\mathcal{I}} \in \mathbf{D}$

Due to *structural sharing* this transformation is considerably smaller than the known one (Baader, Distel, 2008).

3.4 Axiomatization of RRs and RIs

As \mathcal{I} satisfies a RR $\top \sqsubseteq \forall r.C$ if $\bigcup \{ r^{\mathcal{I}}(x) \mid x \in \text{Dom}(\mathcal{I}) \} \subseteq C^{\mathcal{I}}$, the most specific RR on r uses the MMSCD $Y^{\mathcal{I}}$ of $Y := \bigcup \{ r^{\mathcal{I}}(x) \mid x \in \text{Dom}(\mathcal{I}) \}$ in place of C . We thus add these RRs to the CI base, possibly after replacing $\top \sqsubseteq \forall r.Y^{\mathcal{I}}$ with $\top \sqsubseteq \forall r.Y$ and the CIs describing all Z reachable from Y .

Next, we show how RIs can be completely axiomatized. Since \mathcal{I} is finite, all RIs are regular in the following sense.

Proposition 12. *For every RN s , the language $L_{\mathcal{I}}(s) := \{ r_1 \circ \dots \circ r_n \mid \mathcal{I} \text{ satisfies the RI } r_1 \circ \dots \circ r_n \sqsubseteq s \}$ is regular and is accepted by a finite automaton \mathfrak{A}_s of exponential size.*

Proof. We view the interpretation \mathcal{I} as finite automaton over the alphabet consisting of all RNs, with state set $\text{Dom}(\mathcal{I})$, and with the transition relation $\{ (x, r, y) \mid (x, y) \in r^{\mathcal{I}} \}$.

Specifically for objects $x, y \in \text{Dom}(\mathcal{I})$ we denote by $\mathfrak{A}_{x,y}$ the automaton with initial state x and final state y .

Now consider a RN s . The automaton \mathfrak{A}_s is defined as the complement of $\bigcup \{ \mathfrak{A}_{x,y} \mid (x, y) \notin s^{\mathcal{I}} \}$, and it accepts the word $r_1 \circ \dots \circ r_n$ iff. the RI $r_1 \circ \dots \circ r_n \sqsubseteq s$ is satisfied in \mathcal{I} .

Last, we show that the automaton \mathfrak{A}_s has exponential size. The automaton $\bigcup \{ \mathfrak{A}_{x,y} \mid (x, y) \notin s^{\mathcal{I}} \}$ has polynomial size since it is a union of polynomially many copies of \mathcal{I} . To construct the complement automaton, we need to employ the powerset construction to make this automaton deterministic, which leads to an exponential blow-up, and then swap final and non-final states. We conclude that the size of the final automaton \mathfrak{A}_s is exponential in \mathcal{I} . \square

For each RN s , let \mathfrak{A}_s be a finite automaton accepting $L_{\mathcal{I}}(s)$. We convert it into the following RIs and add them to the CI base.

- $p \circ r \sqsubseteq q$ for each transition (p, r, q) in \mathfrak{A}_s
- $\varepsilon \sqsubseteq i$ for the initial state i in \mathfrak{A}_s
- $f \sqsubseteq s$ for each final state f in \mathfrak{A}_s

Note that these RIs use the automaton states as auxiliary RNs. In order to prevent interactions that could produce RIs not satisfied in \mathcal{I} , we assume that w.l.o.g. all automata \mathfrak{A}_s have disjoint state sets. By construction, this set of RIs entails $r_1 \circ \dots \circ r_n \sqsubseteq s$ iff. \mathcal{I} satisfies $r_1 \circ \dots \circ r_n \sqsubseteq s$, i.e. it is a complete axiomatization. It remains open how these RIs can be rewritten into equivalent RIs without auxiliary RNs, but we believe this is possible. However, many reasoners transform the RIs in a given ontology into finite automata anyway and for these the above RIs are advantageous since the automata can easily be read off.

Since we add the above RIs to the CI base, we can also employ modified automata. Specifically, the CI base entails the disjointness axiom $\exists r_1. \dots \exists r_n. \top \sqsubseteq \perp$ iff. the role chain $r_1 \circ \dots \circ r_n$ does not connect any objects in \mathcal{I} , i.e. $(r_1 \circ \dots \circ r_n)^{\mathcal{I}} = \emptyset$. In this case, all RIs $r_1 \circ \dots \circ r_n \sqsubseteq s$ for RNs s are trivially satisfied in \mathcal{I} . To avoid the derivation of such RIs in the automata, we can instead use

$$\mathfrak{A}_s := \bigcup \{ \mathfrak{A}_{x,y} \mid (x, y) \in s^{\mathcal{I}} \} \setminus \bigcup \{ \mathfrak{A}_{x,y} \mid (x, y) \notin s^{\mathcal{I}} \},$$

which accepts the word $r_1 \circ \dots \circ r_n$ iff. the RI $r_1 \circ \dots \circ r_n \sqsubseteq s$ is satisfied in \mathcal{I} and $(r_1 \circ \dots \circ r_n)^{\mathcal{I}} \neq \emptyset$. These automata can be constructed in exponential time as well.

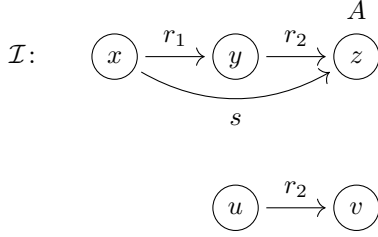
We can now formulate our second main result.

Theorem 13. *For each finite interpretation \mathcal{I} , a complete TBox of \mathcal{EL} CIs, RRs, and RIs satisfied in \mathcal{I} can be computed in exponential time. There are finite interpretations for which such a TBox cannot be of polynomial size.*

In order to ensure polynomial-time reasoning, a syntactic restriction on the interplay of RRs and RIs is imposed on TBoxes expressed in the \mathcal{EL} family and in OWL 2 EL (Baader, Brandt, Lutz, 2008): for each RI $r_1 \circ \dots \circ r_n \sqsubseteq s$ in \mathcal{T} where $n \geq 1$, if \mathcal{T} does not entail $\top \sqsubseteq \forall r_n.C$, then \mathcal{T} neither entails $\top \sqsubseteq \forall s.C$. This restriction prevents new concept memberships for objects in the range of s .

The following example shows that this restriction need not be satisfied when RRs and RIs are constructed as above.

Example XXIII. We consider the interpretation \mathcal{I} defined as follows.



Since z is the only object in the range of s , we obtain the RR $\top \sqsubseteq \forall s.A$. The range of r_2 , however, consists of z and v and we obtain the trivial RR $\top \sqsubseteq \forall r_2.\top$. We do not go into the details regarding the automaton construction, it suffices to know that we would axiomatize RIs that entail $r_1 \circ r_2 \sqsubseteq s$. Since the RR $\top \sqsubseteq \forall r_2.A$ is not entailed, the syntactic restriction is not satisfied.

As a solution, we weaken the RRs. For each RN s , we first compute the finite automaton \mathfrak{A}_s and then identify all RNs that lead to a final state, viz. since each transition (p, r, f) in \mathfrak{A}_s where f is a final state and p is reachable from the initial state encodes a RI of the form $\dots \circ r \sqsubseteq s$. We thus obtain an admissible RR on s by taking all successors of these RNs r into account, i.e. we compute the RR $\top \sqsubseteq \forall s.Y^{\mathcal{I}}$ from the set $Y := \bigcup(\{s^{\mathcal{I}}(x) \mid x \in \text{Dom}(\mathcal{I})\} \cup \{r^{\mathcal{I}}(x) \mid x \in \text{Dom}(\mathcal{I}) \text{ and } (p, r, f) \text{ is a transition in } \mathfrak{A}_s \text{ where } p \text{ is reachable and } f \text{ is final}\})$. By construction, these are the most specific RRs that together with the RIs obtained from the automata satisfy the syntactic restriction.

4 Implementation Details

In this section we describe how the canonical CI base from Theorem 10 can be efficiently computed. We have already seen some details in proofs in Section 3, but there mainly focused on deriving upper complexity bounds.

4.1 Computing a Maximal Simulation

Several steps employ maximal simulations and therefore a performant algorithm for computing these is advantageous. We adapt an approach to computing simulations between graphs (M. R. Henzinger, T. A. Henzinger, Kopke, 1995) such that it works with interpretations (which can be seen as labelled graphs) and runs in parallel on multiple threads.

We will construct a finite sequence of relations $\mathfrak{S}_0 \supset \mathfrak{S}_1 \supset \dots$. The initial relation \mathfrak{S}_0 is the full relation $\text{Dom}(\mathcal{I}) \times \text{Dom}(\mathcal{J})$. The next relation \mathfrak{S}_1 consists of all pairs (x, y) where

- for each CN A , if $x \in A^{\mathcal{I}}$, then $y \in A^{\mathcal{J}}$, and
- for each RN r , if $(x, x') \in r^{\mathcal{I}}$ for some x' , then $(y, y') \in r^{\mathcal{J}}$ for some y' .

The first instruction ensures that \mathfrak{S}_1 and also every subsequent relation only contains pairs (x, y) that satisfy Condition (S1). The second instruction filters out all pairs (x, y) that violate Condition (S2) for x having an r -successor while y does not.

In order to compute the subsequent relations and specifically to ensure that Condition (S2) is fulfilled, we need to

Algorithm 1: Computing the Maximal Simulation

Input: Interpretations \mathcal{I} and \mathcal{J}
Output: Maximal Simulation from \mathcal{I} to \mathcal{J}

```

1  $\mathfrak{S} := \mathfrak{S}_1$  (see above)
2  $R := R_1$  (see above)
3 while  $x \in \text{Dom}(\mathcal{I})$  and  $r \in N_R$  exist with  $R(x, r) \neq \emptyset$ 
4   foreach  $x'$  where  $(x', x) \in r^{\mathcal{I}}$ 
5     foreach  $y'$  where  $y' \in R(x, r)$ 
6       if  $(x', y') \in \mathfrak{S}$ 
7          $\mathfrak{S} := \mathfrak{S} \setminus \{(x', y')\}$ 
8         foreach  $r' \in N_R$ 
9           foreach  $y''$  where  $(y'', y') \in r'^{\mathcal{J}}$ 
10            if  $r'^{\mathcal{J}}(y'') \cap \mathfrak{S}(x') = \emptyset$ 
11               $R(x', r') := R(x', r') \cup \{y''\}$ 
12  $R(x, r) := \emptyset$ 
13 return  $\mathfrak{S}$ 

```

propagate deletion of pairs backwards along the RNs. If we find that an object x in $\text{Dom}(\mathcal{I})$ cannot be simulated by an object y in $\text{Dom}(\mathcal{J})$ and thus delete the pair (x, y) from \mathfrak{S}_{i-1} , then we must examine all pairs $(x', y') \in \mathfrak{S}_i$ where $(x', x) \in r^{\mathcal{I}}$ and $(y', y) \in r^{\mathcal{J}}$ — if there is no object $z \neq y$ such that $(y', z) \in r^{\mathcal{J}}$ and $(x, z) \in \mathfrak{S}_i$, then y' cannot simulate x' and the pair (x', y') must be deleted from \mathfrak{S}_i , for violating Condition (S2).

Now, we formally construct the subsequent relations in the following manner, starting with $i := 1$. If $\mathfrak{S}_{i-1} = \mathfrak{S}_i$, then return \mathfrak{S}_i . Otherwise there is an element $x \in \text{Dom}(\mathcal{I})$ for which $\mathfrak{S}_{i-1}(x) \neq \mathfrak{S}_i(x)$, where $\mathfrak{S}(x)$ denotes the set $\{y \mid (x, y) \in \mathfrak{S}\}$. For each element $x' \in \text{Dom}(\mathcal{I})$, we define

$$\mathfrak{S}_{i+1}(x') := \mathfrak{S}_i(x') \setminus \bigcup \{R_i(x, r) \mid (x', x) \in r^{\mathcal{I}}\} \text{ where}$$

$$R_i(x, r) := \left\{ y' \mid \begin{array}{l} (y', y) \in r^{\mathcal{J}} \text{ for some } (x, y) \in \mathfrak{S}_{i-1} \\ \text{and } (y', z) \notin r^{\mathcal{J}} \text{ for each } (x, z) \in \mathfrak{S}_i \end{array} \right\}$$

Algorithm 1 shows a possible implementation in pseudocode. It maintains the relations \mathfrak{S} and R , but does not keep previous versions in memory.

Note that, using the notation $r^{\mathcal{I}}(x) := \{y \mid (x, y) \in r^{\mathcal{I}}\}$, we have $R_i(x, r) = \{y' \mid r^{\mathcal{J}}(y') \cap \mathfrak{S}_{i-1}(x) \neq \emptyset \text{ and } r^{\mathcal{J}}(y') \cap \mathfrak{S}_i(x) = \emptyset\}$. In Line 10 only the latter condition needs to be checked, since the former one had been satisfied in the previous version of \mathfrak{S} before (x', y') was deleted.

Moreover, we can speed-up Algorithm 1 by parallelization. The data structures used to represent the simulation \mathfrak{S} and the relation R must support concurrent read/write access by multiple threads. Both the computation of \mathfrak{S}_1 in Line 1 and of R_1 in Line 2 can be easily parallelized by iterating over $\text{Dom}(\mathcal{I})$ in parallel (since processing one $x \in \text{Dom}(\mathcal{I})$ is independent from the others). The remaining while-loop in Lines 3–12 can be parallelized as follows.

(First) One first searches for an object $x_0 \in \text{Dom}(\mathcal{I})$ such that $R(x_0, r_0) \neq \emptyset$ for some RN r_0 . When found, a thread is started to process x_0 as in Lines 4–12 and Instruction (Next) is executed (without waiting for the thread to complete); otherwise the algorithm is finished.

(Next) Whenever a thread has been started, say for the object x_i , one searches for another object $x_{i+1} \in \text{Dom}(\mathcal{I})$ with $R(x_{i+1}, r_{i+1}) \neq \emptyset$ for some RN r_{i+1} and—in order to guarantee independence from the currently running computations—that has no common predecessor with any of the objects x_0, \dots, x_i (then no predecessor x' in Line 4 will be processed by different threads). When found, another thread is started to process x_{i+1} as per Lines 4–12 and Instruction (Next) is executed (without waiting for the thread to complete); otherwise one waits until *all* started threads have completed their computations and afterwards Instruction (First) is executed.

In addition, when an object x_i is processed in its own thread, then right away for *all* RNs r_i with $R(x_i, r_i) \neq \emptyset$ in a batch. This yields a further speed-up.

4.2 Reducing the Input Interpretation

Computing the canonical CI base needs exponential time in the worst case. We reduce the input interpretation \mathcal{I} to save computation time. The key observation is that we can group together all objects in $\text{Dom}(\mathcal{I})$ satisfying the same CDs. By doing so, no counterexamples against CIs satisfied in \mathcal{I} are removed, and also no new counterexamples against satisfied CIs are introduced. However, instead of checking infinitely many CIs the following characterization comes to the rescue: if \mathcal{I} and \mathcal{J} are finite, then $(\mathcal{I}, x) \preceq (\mathcal{J}, y)$ iff. $x \in C^{\mathcal{I}}$ implies $y \in C^{\mathcal{J}}$ for all C (Lutz, Wolter, 2010). Thus, in order to decide whether two objects x and y in \mathcal{I} satisfy the same CDs, we check if the maximal simulation $\mathfrak{S}_{\mathcal{I}}$ on \mathcal{I} contains (x, y) as well as (y, x) .

Now, assume that we have computed the maximal simulation $\mathfrak{S}_{\mathcal{I}}$ on \mathcal{I} . By means of it, we formalize the (weak) reduction of \mathcal{I} as follows.

Definition XXIV. An interpretation \mathcal{I} is *weakly reduced* if the following conditions are fulfilled for all $x, y, z \in \text{Dom}(\mathcal{I})$:

- (R1) If $x \approx y$, then $x = y$.
- (R2) If $(x, y) \in r^{\mathcal{I}}$, $(x, z) \in r^{\mathcal{I}}$, and $y \preceq z$, then $y = z$.

Condition (R2) is from (Ecke, Peñaloza, Turhan, 2015).

Definition XXV. The *weak reduction* of \mathcal{I} is denoted as $\text{Red}(\mathcal{I})$, its domain $\text{Dom}(\text{Red}(\mathcal{I}))$ consists of all equivalence classes $[x]_{\approx}$ where $x \in \text{Dom}(\mathcal{I})$, and the interpretation function $\cdot^{\text{Red}(\mathcal{I})}$ is defined by

$$A^{\text{Red}(\mathcal{I})} := \{ [x]_{\approx} \mid x \in A^{\mathcal{I}} \}$$

$$r^{\text{Red}(\mathcal{I})} := \left\{ ([x]_{\approx}, [y]_{\approx}) \mid \begin{array}{l} x \xrightarrow{r} y \text{ and there is no } z \\ \text{with } y \prec z \text{ and } x \xrightarrow{r} z \end{array} \right\}$$

for each CN A and for each RN r , respectively, where $x \xrightarrow{r} y$ indicates that $(x, y') \in r^{\mathcal{I}}$ and $y \preceq y'$ for some y' .

We need to show that the above definition is independent of representatives.

- If $x \approx u$, then $x \in A^{\mathcal{I}}$ implies $u \in A^{\mathcal{I}}$.
- Let $x \approx u$ and $y \approx v$. We first show that $x \xrightarrow{r} y$ implies $u \xrightarrow{r} v$. From $x \xrightarrow{r} y$ it follows that $(x, y') \in r^{\mathcal{I}}$ and $y \preceq y'$ for some y' . Since $x \approx u$ and $(x, y') \in r^{\mathcal{I}}$, there

is some v' with $(u, v') \in r^{\mathcal{I}}$ and $y' \preceq v'$. The latter together with $v \approx y \preceq y'$ yields $v \preceq v'$, which together with $(u, v') \in r^{\mathcal{I}}$ implies $u \xrightarrow{r} v$.

Now, assume that $v \prec w$ and $u \xrightarrow{r} w$ for some w . Since $w \approx v$, the above yields $x \xrightarrow{r} w$. From $y \approx v$ and $v \prec w$ we infer that $y \prec w$. As contraposition we obtain: if there is no z with $y \prec z$ and $x \xrightarrow{r} z$, then there is no w with $v \prec w$ and $u \xrightarrow{r} w$.

Next, we show that the weak reduction $\text{Red}(\mathcal{I})$ is minimal and can be computed in polynomial time. More importantly, we verify that \mathcal{I} and $\text{Red}(\mathcal{I})$ satisfy the same CIs. In order to show the latter as well as to specify the relationship between \mathcal{I} and $\text{Red}(\mathcal{I})$ we introduce the following notion.

Definition XXVI. We say that interpretations \mathcal{I} and \mathcal{J} are *fully similar*, denoted as $\mathcal{I} \approx \mathcal{J}$, if there are mappings $f: \text{Dom}(\mathcal{I}) \rightarrow \text{Dom}(\mathcal{J})$ and $g: \text{Dom}(\mathcal{J}) \rightarrow \text{Dom}(\mathcal{I})$ such that

1. $x \approx f(x)$ for each $x \in \text{Dom}(\mathcal{I})$, and
2. $y \approx g(y)$ for each $y \in \text{Dom}(\mathcal{J})$.

If two interpretations \mathcal{I} and \mathcal{J} are isomorphic, then they are fully similar as well. The following example shows that the converse does not hold.

Example XXVII. The interpretations \mathcal{I} and \mathcal{J} with $r^{\mathcal{I}} := \{(x, x)\}$ and $r^{\mathcal{J}} := \{(y, z), (z, z)\}$ are fully similar but not isomorphic. The maximal simulation from \mathcal{I} to \mathcal{J} is $\{(x, y), (x, z)\}$, and the maximal simulation from \mathcal{J} to \mathcal{I} is $\{(y, x), (z, x)\}$. There is no homomorphism from \mathcal{I} to \mathcal{J} that is surjective, and thus no isomorphism.

The next lemma shows that fully similar interpretations satisfy the same CIs.

Lemma XXVIII. If \mathcal{I} and \mathcal{J} are fully similar, then both satisfy the same $\mathcal{EL}_{\text{si}}^{\perp}$ CIs.

Proof. Let $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, d)$ be a CI satisfied in \mathcal{I} . Further let $y \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{J}}$, i.e. $(\mathcal{C}, c) \preceq (\mathcal{J}, y)$. Since \mathcal{I} and \mathcal{J} are fully similar, $(\mathcal{J}, y) \preceq (\mathcal{I}, g(y)) \preceq (\mathcal{J}, y)$, and thus $(\mathcal{C}, c) \preceq (\mathcal{I}, g(y))$, i.e. $g(y) \in (\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}}$. The assumption yields $g(y) \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\mathcal{I}}$, i.e. $(\mathcal{D}, d) \preceq (\mathcal{I}, g(y))$, and thus $(\mathcal{D}, d) \preceq (\mathcal{J}, y)$, i.e. $y \in (\exists^{\text{sim}}(\mathcal{D}, d))^{\mathcal{J}}$.

Moreover, CIs $\perp \sqsubseteq \exists^{\text{sim}}(\mathcal{D}, d)$ need no special treatment as they are satisfied in all interpretations, and it remains to consider CIs $\exists^{\text{sim}}(\mathcal{C}, c) \sqsubseteq \perp$. Assume that such a CI is satisfied in \mathcal{I} , i.e. $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{I}} = \emptyset$ and so $(\mathcal{C}, c) \not\preceq (\mathcal{I}, x)$ for each $x \in \text{Dom}(\mathcal{I})$. Let $y \in \text{Dom}(\mathcal{J})$. It follows that $(\mathcal{J}, y) \approx (\mathcal{I}, g(y))$ and $(\mathcal{C}, c) \not\preceq (\mathcal{I}, g(y))$, and thus $(\mathcal{C}, c) \not\preceq (\mathcal{J}, y)$. We conclude that $(\exists^{\text{sim}}(\mathcal{C}, c))^{\mathcal{J}} = \emptyset$.

By swapping \mathcal{I} and \mathcal{J} , the converse direction follows as well. \square

Proposition XXIX. $\text{Red}(\mathcal{I})$ can be computed in polynomial time and it is the smallest weakly reduced interpretation that is fully similar to \mathcal{I} (modulo renaming of domain elements).

Proof. Since the maximal simulation $\mathfrak{S}_{\mathcal{I}}$ on \mathcal{I} can be computed in polynomial time, we can determine from it all equivalence classes (in the domain of $\text{Red}(\mathcal{I})$) in polynomial time as follows. We maintain a set X of remaining objects, which is initialized as $\text{Dom}(\mathcal{I})$. As long we can find

an object $x \in X$, we determine the equivalence class $[x]_{\approx}$ as $\{y \mid (x, y) \in \mathfrak{S}_{\mathcal{I}} \text{ and } (y, x) \in \mathfrak{S}_{\mathcal{I}}\}$, add it to the domain of $\text{Red}(\mathcal{I})$, and remove from X all objects in $[x]_{\approx}$.

Note that each equivalence class can also be written as $[x]_{\approx} = \mathfrak{S}_{\mathcal{I}}(x) \cap (\mathfrak{S}_{\mathcal{I}})^{-1}(x)$. If $\mathfrak{S}_{\mathcal{I}}$ is represented in form of a matrix, then $\mathfrak{S}_{\mathcal{I}}(x)$ is a row and $(\mathfrak{S}_{\mathcal{I}})^{-1}(x)$ is a column. It further holds that $x \xrightarrow{\mathcal{I}} y$ iff. $r^{\mathcal{I}}(x) \cap \mathfrak{S}_{\mathcal{I}}(y) \neq \emptyset$, and so all RNs are interpreted as

$$r^{\text{Red}(\mathcal{I})} = \left\{ \left([x]_{\approx}, [y]_{\approx} \right) \mid \begin{array}{l} r^{\mathcal{I}}(x) \cap \mathfrak{S}_{\mathcal{I}}(y) \neq \emptyset \text{ and} \\ r^{\mathcal{I}}(x) \cap (\mathfrak{S}_{\mathcal{I}}(y) \setminus \mathfrak{S}_{\mathcal{I}}^{-1}(y)) = \emptyset \end{array} \right\}.$$

With that, we can determine the extensions of RNs in polynomial time. For CNs this is trivial.

Now, we show that \mathcal{I} and $\text{Red}(\mathcal{I})$ are fully similar. To this end, we first show that the relation

$$\mathfrak{T}_1 := \{ (x, [u]_{\approx}) \mid x \preceq u \}$$

is a simulation from \mathcal{I} to $\text{Red}(\mathcal{I})$.

- (S1) Let $x \in A^{\mathcal{I}}$ and $(x, [u]_{\approx}) \in \mathfrak{T}_1$. Then $x \preceq u$ and thus $u \in A^{\mathcal{I}}$. By definition of $\text{Red}(\mathcal{I})$ it follows that $[u]_{\approx} \in A^{\text{Red}(\mathcal{I})}$.
- (S2) Let $(x, y) \in r^{\mathcal{I}}$ and $(x, [u]_{\approx}) \in \mathfrak{T}_1$, i.e. $x \preceq u$. So there is v such that $(u, v) \in r^{\mathcal{I}}$ and $y \preceq v$. It follows that $u \xrightarrow{\mathcal{I}} y$. If there is no z with $y \prec z$ and $u \xrightarrow{\mathcal{I}} z$, then $([u]_{\approx}, [y]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ by definition and $(y, [y]_{\approx}) \in \mathfrak{T}_1$. Otherwise, there exists a \preceq -maximal z with $y \prec z$ and $u \xrightarrow{\mathcal{I}} z$. Then $([u]_{\approx}, [z]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ and $(y, [z]_{\approx}) \in \mathfrak{T}_1$.

In the converse direction we show that the relation

$$\mathfrak{T}_2 := \{ ([x]_{\approx}, u) \mid x \preceq u \}$$

is a simulation from $\text{Red}(\mathcal{I})$ to \mathcal{I} .

- (S1) Let $[x]_{\approx} \in A^{\text{Red}(\mathcal{I})}$ and $([x]_{\approx}, u) \in \mathfrak{T}_2$, i.e. $x \in A^{\mathcal{I}}$, and $x \preceq u$. We infer that $u \in A^{\mathcal{I}}$.
- (S2) Let $([x]_{\approx}, [y]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ and $([x]_{\approx}, u) \in \mathfrak{T}_2$. The former implies $x \xrightarrow{\mathcal{I}} y$ and the latter implies $x \preceq u$. Thus there is y' such that $(x, y') \in r^{\mathcal{I}}$ and $y \preceq y'$. From $(x, y') \in r^{\mathcal{I}}$ and $x \preceq u$ it follows that $(u, v) \in r^{\mathcal{I}}$ and $y' \preceq v$ for some v . We infer $y \preceq v$ and thus $([y]_{\approx}, v) \in \mathfrak{T}_2$.

Define the mapping $f: \text{Dom}(\mathcal{I}) \rightarrow \text{Dom}(\text{Red}(\mathcal{I}))$ by $f(x) := [x]_{\approx}$, and choose a mapping $g: \text{Dom}(\text{Red}(\mathcal{I})) \rightarrow \text{Dom}(\mathcal{I})$ such that $g([x]_{\approx}) \in [x]_{\approx}$. (Since equivalence classes are non-empty, at least one such g exists.) We then have $(x, f(x)) \in \mathfrak{T}_1$ and $(f(x), x) \in \mathfrak{T}_2$, and thus $x \approx f(x)$. Similarly, it holds that $([x]_{\approx}, g([x]_{\approx})) \in \mathfrak{T}_2$ and $(g([x]_{\approx}), [x]_{\approx}) \in \mathfrak{T}_1$, which implies $[x]_{\approx} \approx g([x]_{\approx})$.

As next step, we show that $\text{Red}(\mathcal{I})$ is weakly reduced.

- (R1) Assume that $[x]_{\approx}$ and $[y]_{\approx}$ are similar, i.e. there is a simulation on $\text{Red}(\mathcal{I})$ containing $([x]_{\approx}, [y]_{\approx})$ and another one containing $([y]_{\approx}, [x]_{\approx})$. Composing the first with \mathfrak{T}_1 on the left as well as with \mathfrak{T}_2 on the right yields a simulation on \mathcal{I} containing (x, y) , i.e. $x \preceq y$. Similarly, we obtain that $y \preceq x$. It follows that $x \approx y$, which means that $[x]_{\approx}$ and $[y]_{\approx}$ are actually equal.

- (R2) Let $([x]_{\approx}, [y]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ and $([x]_{\approx}, [z]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$, and assume that there is a simulation on $\text{Red}(\mathcal{I})$ containing $([y]_{\approx}, [z]_{\approx})$. Composing the simulation with \mathfrak{T}_1 on the left as well as with \mathfrak{T}_2 on the right yields a simulation on \mathcal{I} containing (y, z) , i.e. $y \preceq z$. Furthermore, $([x]_{\approx}, [y]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ implies that there is no z' with $y \preceq z'$ and $x \xrightarrow{\mathcal{I}} z'$, and $([x]_{\approx}, [z]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ implies $x \xrightarrow{\mathcal{I}} z$. Together with $y \preceq z$, the latter yields a contradiction.

Regarding the final step, assume that \mathcal{J} is another weakly reduced interpretation that is fully similar to \mathcal{I} . Since full similarity is transitive, we have that $\text{Red}(\mathcal{I}) \approx \mathcal{J}$. Let $f: \text{Dom}(\text{Red}(\mathcal{I})) \rightarrow \text{Dom}(\mathcal{J})$ and $g: \text{Dom}(\mathcal{J}) \rightarrow \text{Dom}(\text{Red}(\mathcal{I}))$ be the accompanying functions as per Definition XXVI.

From Definition XXVI it follows that $[x]_{\approx} \approx g(f([x]_{\approx}))$, and thus Condition (R1) yields $[x]_{\approx} = g(f([x]_{\approx}))$. It similarly follows that $f(g(u)) = u$. Thus, f and g are inverses of each other and thus are bijective. It follows that that $\text{Red}(\mathcal{I})$ and \mathcal{J} contain the same number of domain elements.

Next, consider a CN A . The bijection f sends each $[x]_{\approx}$ in $A^{\text{Red}(\mathcal{I})}$ to another $f([x]_{\approx})$ in $A^{\mathcal{J}}$ (since $[x]_{\approx} \approx f([x]_{\approx})$), and thus $|A^{\text{Red}(\mathcal{I})}| \leq |A^{\mathcal{J}}|$. (Actually, the converse inequality can be shown as well by means of g .)

Last, consider a RN r and a pair $([x]_{\approx}, [y]_{\approx}) \in r^{\text{Red}(\mathcal{I})}$. We will show that $(f([x]_{\approx}), f([y]_{\approx})) \in r^{\mathcal{J}}$. Since f is bijective, it then follows that $r^{\mathcal{J}}$ contains at least as many pairs as $r^{\text{Red}(\mathcal{I})}$.

Since $[x]_{\approx} \approx f([x]_{\approx})$ by Definition XXVI, Condition (S2) yields an element v such that $(f([x]_{\approx}), v) \in r^{\mathcal{J}}$ and $[y]_{\approx} \preceq v$. Similarly by Condition (S2), there is $[y']_{\approx}$ such that $(g(f([x]_{\approx})), [y']_{\approx}) \in r^{\text{Red}(\mathcal{I})}$ and $v \preceq [y']_{\approx}$. Recall from above that f and g are inverses of each other, which implies that $g(f([x]_{\approx})) = [x]_{\approx}$, i.e. $([x]_{\approx}, [y']_{\approx}) \in r^{\text{Red}(\mathcal{I})}$. We further have that $[y]_{\approx} \preceq v \preceq [y']_{\approx}$. Now Condition (R2) yields that $[y]_{\approx} = [y']_{\approx}$, and consequently that $[y]_{\approx} \approx v$. Since $f([y]_{\approx}) \approx [y]_{\approx}$ by Definition XXVI, we obtain $f([y]_{\approx}) \approx v$. Since \mathcal{J} is weakly reduced, this implies $f([y]_{\approx}) = v$ by Condition (R1), and so we conclude that $(f([x]_{\approx}), f([y]_{\approx})) \in r^{\mathcal{J}}$. \square

The weak reduction $\text{Red}(\mathcal{I})$ is not the smallest interpretation that satisfies the same CIs as \mathcal{I} , but its advantage is that it can be computed in polynomial time. By means of the powering, we could reduce $\text{Red}(\mathcal{I})$ even further. The first important observation is that \mathcal{I} and its powering $\wp(\mathcal{I})$ satisfy the same CIs. Therefore, we could remove from \mathcal{I} every object x that is represented by a subset Y in $\wp(\mathcal{I})$, i.e. $x \notin Y$ and $(\mathcal{I}, x) \approx (\wp(\mathcal{I}), Y)$. The remaining objects are called *strongly irreducible* and constitute the domain of the *strong reduction*. However, determining the strongly irreducible objects needs exponential time. Experiments with the test datasets have shown that, in the cases where determining them was possible, not much can be saved since at least half of the objects in the weak reduction are strongly irreducible (often more than 80 %).

Algorithm 2: Computing all MMSCDs with FCbO

Input: Interpretation \mathcal{I} where $\text{Dom}(\mathcal{I}) = \{x_1, \dots, x_n\}$ **Output:** All MMSCDs in \mathcal{I} (modulo equivalence)

```
1 Closures( $\mathcal{I}$ ) :=  $\emptyset$ 
2 FCbO( $\emptyset, 1, \emptyset, \dots, \emptyset$ )
3 def FCbO( $X, i, N_i, \dots, N_n$ )
4    $\mathcal{Q} := \emptyset$ 
5   foreach  $j \in \{i, \dots, n\}$  in ascending order
6      $M_j := N_j$ 
7     if  $x_j \notin X$ 
8       if  $N_j \cap \{x_1, \dots, x_{j-1}\} \subseteq X \cap \{x_1, \dots, x_{j-1}\}$ 
9          $Y := (X \cup \{x_j\})^{\mathcal{I}\mathcal{I}}$ 
10        if  $X \cap \{x_1, \dots, x_{j-1}\} = Y \cap \{x_1, \dots, x_{j-1}\}$ 
11           $\mathcal{Q} := \mathcal{Q} \cup \{(Y, j)\}$ 
12        else
13           $M_j := Y$ 
14   foreach  $j \in \{i, \dots, n\}$  in ascending order
15     if  $(Y, j) \in \mathcal{Q}$  for some  $Y$ 
16       Closures( $\mathcal{I}$ ) := Closures( $\mathcal{I}$ )  $\cup \{Y\}$ 
17       FCbO( $Y, j + 1, M_{j+1}, \dots, M_n$ )
18 return  $\{Y^{\mathcal{I}} \mid Y \in \text{Closures}(\mathcal{I})\}$ 
```

4.3 Computing all MMSCDs

Recall that the induced context $\mathbb{K}_{\mathcal{I}}$ has object set $\text{Dom}(\mathcal{I})$ and its attribute set \mathbf{M} consists of the bottom CD \perp , all CNs A , and all ERs $\exists r. X^{\mathcal{I}}$ where r is a RN and $X^{\mathcal{I}}$ is a MMSCD of a non-empty subset X of $\text{Dom}(\mathcal{I})$. To compute the attribute set \mathbf{M} , we should not naïvely go through all subsets of $\text{Dom}(\mathcal{I})$ and compute their MMSCDs because there are exponentially many subsets and MMSCDs of different subsets are often equivalent. The following consequence of Properties (G1)–(G7) helps us.

Lemma 14. *The mapping $\phi_{\mathcal{I}}: X \mapsto X^{\mathcal{I}\mathcal{I}}$ is a closure operator⁴ on $\text{Dom}(\mathcal{I})$.*

In order to avoid computing duplicates, we employ an optimized FCA algorithm to enumerate all closures of $\phi_{\mathcal{I}}$. By Property (G4) and Proposition 8 all MMSCDs are then obtained from the closures $X^{\mathcal{I}\mathcal{I}}$ as $\exists^{\text{sim}}(\wp(\mathcal{I}), X^{\mathcal{I}\mathcal{I}})$.

For this purpose, Algorithm 2 is an adaptation of Fast Close-by-One (FCbO) (Krajča, Outrata, Vychodil, 2010), which is an optimized version of Close-by-One (CbO) (Kuznetsov, 1993). In a nutshell, Algorithm 2 jumps through the powerset of $\text{Dom}(\mathcal{I})$ and determines all closures of $\phi_{\mathcal{I}}$. Whenever it has found a closure X , then it recursively calls the function FCbO and generates the next closures by adding one domain element to X and applying $\phi_{\mathcal{I}}$, see Line 9. In order to narrow down the search space and to avoid duplicate computations of the same closure, it uses a so-called *canonicity test* in Lines 8 and 10. Computations involving the different values of j in Lines 6–13 are independent of each other and can thus run in parallel

⁴Definition. A *closure operator* on a set M is a mapping $\phi: \wp(M) \rightarrow \wp(M)$ that is extensive ($X \subseteq X^{\phi}$), monotone ($X \subseteq Y$ implies $X^{\phi} \subseteq Y^{\phi}$), and idempotent ($X^{\phi\phi} = X^{\phi}$). A *closure* of ϕ is a subset X of M with $X = X^{\phi}$.

on multiple threads. Since we later need the closures themselves, e.g. in Lemma XXXI, an implementation should return in Line 18 all MMSCDs together with the closure from which they are induced, i.e. all pairs $(Y, Y^{\mathcal{I}})$ where $Y \in \text{Closures}(\mathcal{I})$.

Comparisons of CbO-based and other algorithms can be found in (Konečný, Krajča, 2021; Kuznetsov, Obiedkov, 2002). There are also other, possibly faster algorithms that can compute a closure system, but require that the closure operator comes in form of a formal context, e.g. In-Close5 (Andrews, 2011, 2014, 2017, 2018) or LCM (Janošík, Konečný, Krajča, 2022a). Thus, these are either not applicable or one must find an efficient way to describe the closure operator by a formal context.

The operator $\phi_{\mathcal{I}}$ is computed with the maximal simulation $\mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}$ from the powering $\wp(\mathcal{I})$ to \mathcal{I} .

Lemma 15. $X^{\mathcal{I}\mathcal{I}} = \mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}(X)$ if $\emptyset \neq X \subseteq \text{Dom}(\mathcal{I})$.

Proof. Consider a non-empty subset X of $\text{Dom}(\mathcal{I})$. Proposition 8 shows that the MMSCD $X^{\mathcal{I}}$ is equivalent to the \mathcal{EL}_{si} CD $\exists^{\text{sim}}(\wp(\mathcal{I}), X)$. It follows that $X^{\mathcal{I}\mathcal{I}} = (\exists^{\text{sim}}(\wp(\mathcal{I}), X))^{\mathcal{I}}$. By Lemma 1, the latter extension equals $\mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}(X)$, where $\mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}$ is the maximal simulation from $\wp(\mathcal{I})$ to \mathcal{I} . \square

To avoid fully constructing the exponentially-large powering, we lazily build only the part reachable from X when a closure $X^{\mathcal{I}\mathcal{I}}$ is computed.

By means of generators, these closures and their MMSCDs can be computed more efficiently. Specifically, in Line 9 each closure $Y := (X \cup \{x_j\})^{\mathcal{I}\mathcal{I}}$ is computed from its *direct generator* $X \cup \{x_j\}$. It is cheaper to compute its MMSCD $Y^{\mathcal{I}}$ from this generator rather than from the closure since the generator is usually smaller than the closure. In particular, the MMSCD $Y^{\mathcal{I}}$ is equivalent to $\exists^{\text{sim}}(\wp(\mathcal{I}), X \cup \{x_j\})$ by Property (G4) and Proposition 8.

In a direct generator $X \cup \{x_j\}$, the set X is another, smaller closure since X comes from the first argument in the recursive call in Line 17. By subsequently replacing these closures with their direct generators, we obtain even smaller generators. To this end, an implementation maintains a mapping g from closures to their direct generators. Now, given a closure Y we compute its *generator* $g^*(Y)$ as follows. First, let $g^1(Y) := g(Y) = X_1 \cup \{x_{j_1}\}$. For the inductive step, assume $g^k(Y) = X_k \cup \{x_{j_1}, \dots, x_{j_k}\}$.

- If $X_k = \emptyset$, then set $g^*(Y) := \{x_{j_1}, \dots, x_{j_k}\}$.
- Otherwise, define $g^{k+1}(Y) := X_{k+1} \cup \{x_{j_1}, \dots, x_{j_{k+1}}\}$ where $g(X_k) = X_{k+1} \cup \{x_{j_{k+1}}\}$.

Lemma XXX. $Y = g^*(Y)^{\mathcal{I}\mathcal{I}}$

Proof. We show by induction that $Y = g^k(Y)^{\mathcal{I}\mathcal{I}}$. We already know that $Y = g(Y)^{\mathcal{I}\mathcal{I}}$, which is the induction base since $g^1(Y) = g(Y)$. Now let $k > 1$ for the induction step, and assume $g^k(Y) = X_k \cup \{x_{j_1}, \dots, x_{j_k}\}$.

- If $X_k = \emptyset$, then $g^*(Y) = g^k(Y)$ and thus the induction hypothesis yields $Y = g^*(Y)^{\mathcal{I}\mathcal{I}}$.

- Otherwise, we have $g^{k+1}(Y) := X_{k+1} \cup \{x_{j_1}, \dots, x_{j_{k+1}}\}$ where $g(X_k) = X_{k+1} \cup \{x_{j_{k+1}}\}$. We infer the following.

$$\begin{aligned}
& g^{k+1}(Y)^{\mathcal{I}\mathcal{I}} \\
&= (X_{k+1} \cup \{x_{j_1}, \dots, x_{j_{k+1}}\})^{\mathcal{I}\mathcal{I}} \\
&= (g(X_k) \cup \{x_{j_1}, \dots, x_{j_k}\})^{\mathcal{I}\mathcal{I}} \\
&= (g(X_k)^{\mathcal{I}} \vee \{x_{j_1}, \dots, x_{j_k}\}^{\mathcal{I}})^{\mathcal{I}} \quad \text{by } (*) \\
&= (g(X_k)^{\mathcal{I}\mathcal{I}} \vee \{x_{j_1}, \dots, x_{j_k}\}^{\mathcal{I}})^{\mathcal{I}} \quad \text{by (G4)} \\
&= (X_k^{\mathcal{I}} \vee \{x_{j_1}, \dots, x_{j_k}\}^{\mathcal{I}})^{\mathcal{I}} \quad \text{by I.H.} \\
&= (X_k \cup \{x_{j_1}, \dots, x_{j_k}\})^{\mathcal{I}\mathcal{I}} \quad \text{by } (*) \\
&= g^k(Y)^{\mathcal{I}\mathcal{I}} \\
&= Y \quad \text{by I.H.}
\end{aligned}$$

It remains to specify (*). Given two \mathcal{EL}_{si} CDs C and D , their *least common subsumer* (LCS, supremum) $C \vee D$ is an \mathcal{EL}_{si} CD that is uniquely defined up to equivalence by two conditions:

1. $C \sqsubseteq^{\emptyset} C \vee D$ and $D \sqsubseteq^{\emptyset} C \vee D$
2. for each \mathcal{EL}_{si} CD E , if $C \sqsubseteq^{\emptyset} E$ and $D \sqsubseteq^{\emptyset} E$, then $C \vee D \sqsubseteq^{\emptyset} E$.

LCS always exist and can be computed by products: $\exists^{\text{sim}}(\mathcal{C}, c) \vee \exists^{\text{sim}}(\mathcal{D}, d) \equiv^{\emptyset} \exists^{\text{sim}}(\mathcal{C} \times \mathcal{D}, (c, d))$. The above used Property (*) is $(X \cup Y)^{\mathcal{I}} \equiv^{\emptyset} X^{\mathcal{I}} \vee Y^{\mathcal{I}}$ for all subsets $X, Y \subseteq \text{Dom}(\mathcal{I})$ (Baader, Distel, 2008), and it follows immediately from the definition of MMSCDs and LCSs. \square

Specifically, to compute a closure $Y := (X \cup \{x_j\})^{\mathcal{I}\mathcal{I}}$ in Line 9 of Algorithm 2, we determine its generator $g^*(Y) = g^*(X) \cup \{x_j\}$, build the sub-interpretation \mathcal{P} of $\wp(\mathcal{I})$ consisting of $g^*(Y)$ and its descendants, compute the maximal simulation $\mathfrak{S}_{\mathcal{P}, \mathcal{I}}$ from \mathcal{P} to \mathcal{I} , and then return the row $\mathfrak{S}_{\mathcal{P}, \mathcal{I}}(g^*(Y))$. In addition, this sub-interpretation \mathcal{P} represents the MMSCD $Y^{\mathcal{I}}$, which is equivalent to $\exists^{\text{sim}}(\wp(\mathcal{I}), g^*(Y))$ by Property (G4) and Proposition 8 and thus also to $\exists^{\text{sim}}(\mathcal{P}, g^*(Y))$. We therefore add to the attribute set \mathbf{M} the ERs $\exists r. \exists^{\text{sim}}(\mathcal{P}, g^*(Y))$ for all RNs r and all closures Y . In addition, we memorize that Y is the closure of $g^*(Y)$ to avoid recomputing this when needed later, e.g. by means of a mapping c with $c(g^*(Y)) := Y$. By Lemma XXX, $c(Z) = Z^{\mathcal{I}\mathcal{I}}$ for each generator $Z := g^*(Y)$.

For some datasets even these lazily constructed parts of $\wp(\mathcal{I})$ are so large that they cannot be computed within reasonable time limits. In order to detect such cases beforehand and to not waste computation time, the prototype approximates, for the current object set at which the powering is to be expanded, the number of successors — if it is larger than 10,000,000, the computation will be aborted. In order to fulfil the requirements of a closure operator, we then return the set $\text{Dom}(\mathcal{I})$, which is the largest closure. The resulting CI base will, however, not be complete anymore since some attributes required in the set \mathbf{M} could not be computed. But, if completeness comes for the price of extremely large CIs, which might not have practical relevance or suffer from overfitting, then one can probably dispense with this goal. Moreover, we also allowed to manually specify a smaller

limit on the number of successors and thereby to further restrict the size of CIs in the base. We turned this bound into a *conjunction size limit* by also counting the CNs that label the particular object set in $\wp(\mathcal{I})$. It remains unclear to which extent completeness is lost, and we leave the investigation as future research. We expect that completeness is still guaranteed for all CIs that obey the conjunction size limit, but modifications to the method might be needed to achieve this. Another way to limit the size of MMSCDs is by restricting their role depth, i.e. by looking into the powering only up to a pre-defined depth, see Section 5.3.

4.4 Computing the Induced Context

Next, we are concerned with efficiently computing the incidence relation I of the induced context $\mathbb{K}_{\mathcal{I}}$. Recall that it consists of all pairs $(x, C) \in I$ where $x \in C^{\mathcal{I}}$. Since $\perp^{\mathcal{I}}$ is empty, I does not contain any pair (x, \perp) . For each object x and each CN A , determining if I contains the pair (x, A) is a simple look-up in the interpretation function (check if $A^{\mathcal{I}}$ contains x). The following lemma shows that also the other incidence pairs are easy to determine, viz. because we compute the ERs $\exists r. X^{\mathcal{I}}$ in \mathbf{M} as explained in Section 4.3 and thus we always have $c(X) = X^{\mathcal{I}\mathcal{I}}$.

Lemma XXXI. $(x, \exists r. Y^{\mathcal{I}}) \in I$ iff. $r^{\mathcal{I}}(x) \cap c(Y) \neq \emptyset$ for each $x \in \text{Dom}(\mathcal{I})$ and each $\exists r. Y^{\mathcal{I}} \in \mathbf{M}$.

Proof. Consider an object $x \in \text{Dom}(\mathcal{I})$ and an ER $\exists r. Y^{\mathcal{I}}$ in \mathbf{M} . Then $(x, \exists r. Y^{\mathcal{I}}) \in I$ iff. there is an object y such that $(x, y) \in r^{\mathcal{I}}$ and $(\wp(\mathcal{I}), Y) \preceq (\mathcal{I}, y)$. Recall that the latter holds iff. $y \in \mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}(Y)$, where $\mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}$ is the maximal simulation. Since $c(Y) = Y^{\mathcal{I}\mathcal{I}}$, and $Y^{\mathcal{I}\mathcal{I}} = \mathfrak{S}_{\wp(\mathcal{I}), \mathcal{I}}(Y)$ by Lemma 15, we conclude that $(x, \exists r. Y^{\mathcal{I}}) \in I$ iff. $r^{\mathcal{I}}(x) \cap c(Y) \neq \emptyset$. \square

4.5 Computing the Background Implications

Recall that the background implication set $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ consists of the implications

- $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]}) \rightarrow \{E \mid E \in \mathbf{M} \text{ and } C \sqsubseteq^{\mathcal{T}} E\}$ for each CI $C \sqsubseteq D$ in \mathcal{T}
- $\{E\} \rightarrow \{F\}$ for each two $E, F \in \mathbf{M}$ with $E \sqsubseteq^{\emptyset} F$.

For those of the first kind, we go through all CIs $C \sqsubseteq D$ in \mathcal{T} and transform them into implications as follows.

1. According to the definition of the partial closure $C^{[\mathcal{I}\mathcal{I}]}$, the premise $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]})$ contains, up to equivalence, all CNs in the top-level conjunction of C as well as all ERs $\exists r. D^{\mathcal{I}\mathcal{I}}$ where $\exists r. D$ is in the top-level conjunction of C . All CNs are attributes in \mathbf{M} . Furthermore, each ER $\exists r. D^{\mathcal{I}\mathcal{I}}$ is equivalent to an attribute in \mathbf{M} . To find this attribute, we first compute the extension $D^{\mathcal{I}}$, which is a closure of the operator $\phi_{\mathcal{I}}$ by Property (G7). Then we determine its generator $Z := g^*(D^{\mathcal{I}})$. According to our construction of \mathbf{M} in Section 4.3, the ER $\exists r. D^{\mathcal{I}\mathcal{I}}$ is represented by the attribute $\exists r. Z^{\mathcal{I}}$ in \mathbf{M} , and so the premise $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]})$ contains this attribute.
2. In order to determine the conclusion consisting of all attributes $E \in \mathbf{M}$ with $C \sqsubseteq^{\mathcal{T}} E$, we first construct the

most specific consequence $C^{\mathcal{T}}$ and, by Proposition 3, we then check using maximal simulations which $E \in \mathbf{M}$ satisfy $C^{\mathcal{T}} \sqsubseteq^{\emptyset} E$. If \mathcal{T} is an \mathcal{EL} TBox, then $C^{\mathcal{T}}$ can be efficiently obtained from the classification or rather the canonical model computed by the reasoner ELK.

To determine the implications of the second kind, we use the recursive characterization of subsumption in Section 2.2. The only non-trivial such implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ are the implications $\{\perp\} \rightarrow \{F\}$ for all $F \in \mathbf{M}$ as well as the implications $\{\exists r. X^{\mathcal{I}}\} \rightarrow \{\exists r. Y^{\mathcal{I}}\}$ where $X^{\mathcal{I}} \sqsubseteq^{\emptyset} Y^{\mathcal{I}}$. Recall that, since X and Y are generators, $c(X) = X^{\mathcal{I}\mathcal{I}}$ and $c(Y) = Y^{\mathcal{I}\mathcal{I}}$ by Lemma XXX.

- If $X^{\mathcal{I}} \sqsubseteq^{\emptyset} Y^{\mathcal{I}}$, then $X^{\mathcal{I}\mathcal{I}} \subseteq Y^{\mathcal{I}\mathcal{I}}$ by Property (G5), and thus $c(X) \subseteq c(Y)$.
- Conversely, $c(X) \subseteq c(Y)$ implies $X^{\mathcal{I}} \sqsubseteq^{\emptyset} Y^{\mathcal{I}}$ by Properties (G2) and (G4).

Thus, we go through all ERs in \mathbf{M} and add the implication $\{\exists r. X^{\mathcal{I}}\} \rightarrow \{\exists r. Y^{\mathcal{I}}\}$ to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ whenever $c(X) \subseteq c(Y)$.

4.6 Computing the CI Base

Our next goal is to compute the canonical implication base of the induced context $\mathbb{K}_{\mathcal{I}}$ relative to $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$, which we afterwards transform into the canonical CI base $\sqcap \text{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}})$ from Theorem 10.

First of all, the set of all pseudo-intents and all intents can be described by a closure operator (Ganter, 1984; Stumme, 1996). As in Section 4.3, we can employ an FCA algorithm to enumerate all these closures. This not only yields all pseudo-intents, from which the canonical implication base is built, but also all intents as a by-product. It is currently unclear whether all pseudo-intents can be efficiently computed without them. The number of pseudo-intents can be exponential in the size of the formal context as well as in the number of intents, and several decision problems related to pseudo-intents are intractable (Babin, Kuznetsov, 2010, 2013; Distel, 2010; Distel, Sertkaya, 2011; Kuznetsov, 2004; Kuznetsov, Obiedkov, 2006, 2008; Sertkaya, 2009a,b).

Currently, LinCbO is the fastest (single-threaded) algorithm for computing canonical implication bases (Janoščík, Konečný, Krajča, 2021a,b, 2022b). It is based on Close-by-One (CbO) (Kuznetsov, 1993) and closures w.r.t. implications are computed with an improved version of LinClosure (Beerli, Bernstein, 1979) that reuses counters. Like all CbO-based algorithms, it uses the *canonicity test* to avoid duplicate computations of the same closure. This test is integrated into the modified LinClosure sub-routine, which enables *early stop* of unnecessary computation branches, and is additionally supported by *pruning rules*. First experiments with the C++-implementation [8] of LinCbO showed a satisfying performance.

However, we need to extend LinCbO with support for background implications. The only difference between pseudo-intents with and without a background implication set \mathcal{L} is that the former additionally are closed under \mathcal{L} . We can easily ensure this by initially adding all background implications to the implication set maintained by LinCbO.

Referring to Algorithm 5 in (Janoščík, Konečný, Krajča, 2022b), we initialize its variables as follows:

- add every background implication in \mathcal{L} to the maintained implication set \mathcal{T} ,
- set $list[i] := \{U \rightarrow V \mid U \rightarrow V \in \mathcal{L} \text{ and } i \in U\}$ for each attribute i ,
- compute the set $F := \bigcup \{V \mid \emptyset \rightarrow V \in \mathcal{L}\}$,
- initialize the first counter as $count[U \rightarrow V] := |U \setminus F|$ for each $U \rightarrow V \in \mathcal{L}$.

Afterwards, we start the recursive computation by calling `LinCbO_P_Step(F, 0, F, count)` in place of the call `LinCbO_P_Step($\emptyset, 0, \emptyset, \emptyset$)`.

5 Variations

The canonical CI base from Theorem 10 is complete for all CIs satisfied in the given interpretation \mathcal{I} , even for all disjointness axioms as well as for huge CIs that might not have practical relevance. We will describe variations that dispense with the computation of such CIs.

5.1 Not Computing Disjointness Axioms

Some CIs in the canonical CI base $\text{Can}(\mathcal{I}, \mathcal{T})$ are disjointness axioms $C \sqsubseteq \perp$, which express that no objects in \mathcal{I} are described by C . Sometimes only the other CIs $C \sqsubseteq D$ are desired as they describe the implications between CDs that are satisfied and also witnessed in \mathcal{I} . We have seen in experiments that more than half of the computation time is required for generating disjointness axioms. It is cheaper to compute only the witnessed CIs since some intermediate computation steps can be stopped early.

Definition 16. A CI $C \sqsubseteq D$ is *witnessed* in \mathcal{I} if $C^{\mathcal{I}} \neq \emptyset$ and $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. A TBox is *witnessed complete* for \mathcal{I} if it entails all CIs that are witnessed in \mathcal{I} . A *witnessed CI base* of \mathcal{I} relative to \mathcal{T} is a TBox \mathcal{B} that consists of witnessed CIs and for which $\mathcal{B} \cup \mathcal{T}$ is witnessed complete.

We partition the canonical CI base from Theorem 10 into two sub-TBoxes: $\text{Can}_+(\mathcal{I}, \mathcal{T})$ consists of all witnessed CIs and $\text{Can}_\perp(\mathcal{I}, \mathcal{T})$ consists of the remaining CIs.

Lemma XXXII. $\text{Can}_\perp(\mathcal{I}, \mathcal{T})$ consists of disjointness axioms.

Proof. Let $\sqcap C \sqsubseteq \sqcap C^{II}$ be a non-witnessed CI in the canonical CI base $\text{Can}(\mathcal{I}, \mathcal{T})$, i.e. $(\sqcap C)^{\mathcal{I}} = \emptyset$. It follows that, in the induced context $\mathbb{K}_{\mathcal{I}}$, there is no object that has *all* attributes in C , i.e. $C^{\mathcal{I}} = \emptyset$. Thus $C^{II} = \mathbf{M}$, which specifically means that $\perp \in C^{II}$ and so $\sqcap C^{II}$ is equal to \perp (modulo equivalence). \square

Proposition 17. $\text{Can}_+(\mathcal{I}, \mathcal{T})$ is a witnessed CI base of \mathcal{I} relative to \mathcal{T} . Among all witnessed CI bases of \mathcal{I} relative to \mathcal{T} it contains the fewest CIs.

Proof. We first show witnessed completeness. Recall from the proof of Proposition 3 that $\mathcal{EL}_{\mathcal{I}}^\perp$ is invariant under direct products. Assume that $C \sqsubseteq D$ is satisfied in \mathcal{I} and $C^{\mathcal{I}} \neq \emptyset$. Consider a model \mathcal{J} of $\text{Can}_+(\mathcal{I}, \mathcal{T}) \cup \mathcal{T}$. We need to show that $C \sqsubseteq D$ is satisfied in \mathcal{J} .

Product invariance yields that the direct product $\mathcal{I} \times \mathcal{J}$ is a model of $\text{Can}_+(\mathcal{I}, \mathcal{T}) \cup \mathcal{T}$. We show that $\mathcal{I} \times \mathcal{J}$

also a model of the other CIs in $\text{Can}(\mathcal{I}, \mathcal{T})$. Let $E \sqsubseteq F \in \text{Can}(\mathcal{I}, \mathcal{T})$ with $E^{\mathcal{I}} = \emptyset$. It follows that $E^{\mathcal{I} \times \mathcal{J}} = \emptyset$ and thus $\mathcal{I} \times \mathcal{J}$ satisfies $E \sqsubseteq F$.

Completeness of $\text{Can}(\mathcal{I}, \mathcal{T})$ yields that $C \sqsubseteq D$ is entailed by $\text{Can}(\mathcal{I}, \mathcal{T}) \cup \mathcal{T}$. We conclude that $C^{\mathcal{I} \times \mathcal{J}} \subseteq D^{\mathcal{I} \times \mathcal{J}}$. Since $C^{\mathcal{I}} \neq \emptyset$, there is some $x \in C^{\mathcal{I}}$. In order to verify that $C^{\mathcal{J}} \subseteq D^{\mathcal{J}}$, we consider an object $y \in C^{\mathcal{J}}$. Then $(x, y) \in C^{\mathcal{I} \times \mathcal{J}}$ by product invariance, and thus $(x, y) \in D^{\mathcal{I} \times \mathcal{J}}$. By product invariance, we infer that $y \in D^{\mathcal{J}}$.

Next, we show minimality. Consider another witnessed CI base \mathcal{B} . Due to witnessed completeness, $\mathcal{B} \cup \mathcal{T} \models \text{Can}_+(\mathcal{I}, \mathcal{T})$ and thus $\mathcal{B} \cup \text{Can}_\perp(\mathcal{I}, \mathcal{T}) \cup \mathcal{T} \models \text{Can}(\mathcal{I}, \mathcal{T})$. It follows that the union of \mathcal{B} and $\text{Can}_\perp(\mathcal{I}, \mathcal{T})$ is a CI base. According to Theorem 10, this union cannot contain fewer CIs than $\text{Can}(\mathcal{I}, \mathcal{T})$, which implies that \mathcal{B} must contain at least as many CIs as $\text{Can}_+(\mathcal{I}, \mathcal{T})$. \square

To compute $\text{Can}_+(\mathcal{I}, \mathcal{T})$, we should exclude from the attribute set \mathbf{M} all CDs not describing any object in \mathcal{I} . Specifically, denote by $\mathbb{K}_{\mathcal{I}} \upharpoonright_{\text{None}}$ the sub-context of $\mathbb{K}_{\mathcal{I}}$ with attribute set $\mathbf{M} \upharpoonright_{\text{None}} := \{C \mid C \in \mathbf{M} \text{ and } C^{\mathcal{I}} \neq \emptyset\}$, and let $\mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}$ be the subset of $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ consisting of all implications in which only attributes from $\mathbf{M} \upharpoonright_{\text{None}}$ occur. The witnessed CI base $\text{Can}_+(\mathcal{I}, \mathcal{T})$ can be computed from this sub-context.

Lemma XXXIII. $\text{Can}_+(\mathcal{I}, \mathcal{T})$ is equal to $\{\sqcap \mathbf{C} \sqsubseteq \sqcap \mathbf{D} \mid \mathbf{C} \rightarrow \mathbf{D} \in \text{Can}(\mathbb{K}_{\mathcal{I}} \upharpoonright_{\text{None}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}) \text{ and } (\sqcap \mathbf{C})^{\mathcal{I}} \neq \emptyset\}$.

Proof. According to Theorem 10 and Section 2.4, the witnessed CI base $\text{Can}_+(\mathcal{I}, \mathcal{T})$ consists of the CIs $\sqcap \mathbf{C} \sqsubseteq \sqcap \mathbf{D}$ where \mathbf{C} is a pseudo-intent of the induced context $\mathbb{K}_{\mathcal{I}}$ relative to the background implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ with $(\sqcap \mathbf{C})^{\mathcal{I}} \neq \emptyset$, and $\mathbf{D} = \mathbf{C}^{II}$.

We denote by J the incidence relation of $\mathbb{K}_{\mathcal{I}} \upharpoonright_{\text{None}}$, i.e. $J := I \cap (\text{Dom}(\mathcal{I}) \times \mathbf{M} \upharpoonright_{\text{None}})$. The following two statements hold.

(α) If $\mathbf{C} \subseteq \mathbf{M} \upharpoonright_{\text{None}}$, then $\mathbf{C}^I = \mathbf{C}^J$.

Recall from Section 2.4 that \mathbf{C}^I is the set of all elements in $\text{Dom}(\mathcal{I})$ that satisfy every atom in \mathbf{C} . Since I and J coincide on $\text{Dom}(\mathcal{I}) \times \mathbf{M} \upharpoonright_{\text{None}}$ and \mathbf{C} is a subset of $\mathbf{M} \upharpoonright_{\text{None}}$, also \mathbf{C}^J is the set of all elements in $\text{Dom}(\mathcal{I})$ that satisfy every atom in \mathbf{C} , i.e. $\mathbf{C}^I = \mathbf{C}^J$.

(β) If $X \subseteq \text{Dom}(\mathcal{I})$ with $X \neq \emptyset$, then $X^I = X^J$.

The definition of J yields $X^J = X^I \cap \mathbf{M} \upharpoonright_{\text{None}}$. Recall from Section 2.4 that X^I is the set of atoms in \mathbf{M} that are satisfied by each domain element in X . Since X is non-empty, X^I must be a subset of $\mathbf{M} \upharpoonright_{\text{None}}$, and thus $X^J = X^I$.

By means of an induction along the subset inclusion \subseteq , we show that a set $\mathbf{C} \subseteq \mathbf{M}$ with $(\sqcap \mathbf{C})^{\mathcal{I}} \neq \emptyset$ is a pseudo-intent of $\mathbb{K}_{\mathcal{I}}$ w.r.t. $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ iff. it is a pseudo-intent of the sub-context $\mathbb{K}_{\mathcal{I}} \upharpoonright_{\text{None}}$ w.r.t. $\mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}$. Consider some such \mathbf{C} .

(PI1) From $(\sqcap \mathbf{C})^{\mathcal{I}} \neq \emptyset$ we infer that $C^{\mathcal{I}} \neq \emptyset$ for each $C \in \mathbf{C}$, and thus $\mathbf{C} \subseteq \mathbf{M} \upharpoonright_{\text{None}}$. Statement (α) yields $\mathbf{C}^J = \mathbf{C}^I$. Since $\mathbf{C}^I = (\sqcap \mathbf{C})^{\mathcal{I}} \neq \emptyset$, it follows with Statement (β) that $\mathbf{C}^{II} = \mathbf{C}^{JJ}$. So $\mathbf{C} \neq \mathbf{C}^{II}$ iff. $\mathbf{C} \neq \mathbf{C}^{JJ}$.

(PI2) If \mathbf{C} is closed under the background implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$, then it is also closed under the implications in the subset $\mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}$.

Conversely, let \mathbf{C} be closed under $\mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}$. We need to show that \mathbf{C} is also closed under $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$. We proceed with a case distinction on the implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$.

- First, consider an implication $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]}) \rightarrow \{E \mid E \in \mathbf{M} \text{ and } C \sqsubseteq^{\mathcal{T}} E\}$ for some CI $C \sqsubseteq D$ in \mathcal{T} , and assume $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]}) \subseteq \mathbf{C}$. The latter implies $C^{[\mathcal{I}\mathcal{I}]} \supseteq^{\emptyset} \sqcap \mathbf{C}$ and thus $(C^{[\mathcal{I}\mathcal{I}]})^{\mathcal{I}} \supseteq (\sqcap \mathbf{C})^{\mathcal{I}}$ by Property (G5). Since $(\sqcap \mathbf{C})^{\mathcal{I}}$ is non-empty, also $(C^{[\mathcal{I}\mathcal{I}]})^{\mathcal{I}}$ is non-empty, and thus $\text{Conj}(C^{[\mathcal{I}\mathcal{I}]})$ is a subset of $\mathbf{M} \upharpoonright_{\text{None}}$.

Now consider an attribute E in the conclusion of the implication. Recall from Section 3.2 that $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq^{\emptyset} C$, and thus $C^{[\mathcal{I}\mathcal{I}]} \sqsubseteq^{\mathcal{T}} E$. Since \mathcal{I} is a model of \mathcal{T} , we have $(C^{[\mathcal{I}\mathcal{I}]})^{\mathcal{I}} \subseteq E^{\mathcal{I}}$, which implies that also $E^{\mathcal{I}}$ is non-empty and thus $E \in \mathbf{M} \upharpoonright_{\text{None}}$. We conclude that the considered implication is contained in $\mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}$, and therefore its conclusion is a subset of \mathbf{C} .

- It remains to consider an implication $\{E\} \rightarrow \{F\}$ where $E, F \in \mathbf{M}$ and $E \sqsubseteq^{\emptyset} F$, for which we assume $E \in \mathbf{C}$. From the latter we infer that $E \supseteq^{\emptyset} \sqcap \mathbf{C}$ and so $E^{\mathcal{I}} \supseteq (\sqcap \mathbf{C})^{\mathcal{I}}$ by Property (G5). Thus $E^{\mathcal{I}}$ is non-empty, i.e. $E \in \mathbf{M} \upharpoonright_{\text{None}}$. We further have $E^{\mathcal{I}} \subseteq F^{\mathcal{I}}$, which implies that also $F^{\mathcal{I}}$ is non-empty and thus $F \in \mathbf{M} \upharpoonright_{\text{None}}$. We conclude that the considered implication $\{E\} \rightarrow \{F\}$ is contained in $\mathcal{L}_{\mathcal{I}, \mathcal{T}} \upharpoonright_{\text{None}}$, and so $F \in \mathbf{C}$.

(PI3) The induction hypothesis shows that the pseudo-intents \mathbf{Q} that are strict subsets of \mathbf{C} are the same in $\mathbb{K}_{\mathcal{I}}$ and $\mathbb{K}_{\mathcal{I}} \upharpoonright_{\text{None}}$. Let $\mathbf{Q} \subset \mathbf{C}$ be such a pseudo-intent. Then $(\sqcap \mathbf{Q})^{\mathcal{I}} \supseteq (\sqcap \mathbf{C})^{\mathcal{I}}$ and thus $(\sqcap \mathbf{Q})^{\mathcal{I}} \neq \emptyset$. With similar arguments as above we infer that $\mathbf{Q}^{II} = \mathbf{Q}^{JJ}$. So $\mathbf{Q}^{II} \subseteq \mathbf{C}$ iff. $\mathbf{Q}^{JJ} \subseteq \mathbf{C}$. \square

Implementation Details. We have seen in Section 4.3 how the attribute set \mathbf{M} can be efficiently computed, and we will now describe modifications regarding the subset $\mathbf{M} \upharpoonright_{\text{None}}$. Of course, we always exclude the bottom CD \perp from \mathbf{M} . It is easy to determine which CNs must be excluded from \mathbf{M} (just check if $A^{\mathcal{I}}$ is empty). However, we should not filter the ERs $\exists r. Y^{\mathcal{I}}$ after they have been computed but rather avoid their computation (since computing closures is expensive). According to Lemma XXXI, $\exists r. Y^{\mathcal{I}}$ must be excluded from \mathbf{M} iff. $r^{\mathcal{I}}(x) \cap c(Y) = \emptyset$ for each $x \in \text{Dom}(\mathcal{I})$. Thus, we should compute only those closures of $\phi_{\mathcal{I}}$ that contain an object with a predecessor.

In order to achieve this with FCbO, we need to carefully modify it. Recall from Algorithm 2 that an enumeration $\{x_1, \dots, x_n\}$ of the domain $\text{Dom}(\mathcal{I})$ is expected. This enumeration influences the order in which subsets X of $\text{Dom}(\mathcal{I})$ occur during the algorithm's run.

The first call of the function FCbO is with arguments $X = \emptyset$ and $i = 1$. For each $j \in \{1, \dots, n\}$, the closure $\{x_j\}^{\mathcal{I}\mathcal{I}}$ is computed in Line 9. If $\{x_j\}^{\mathcal{I}\mathcal{I}}$ does not contain an object x_k where $k < j$, then FCbO is recursively called with arguments $X = \{x_j\}^{\mathcal{I}\mathcal{I}}$ and $i = j + 1$. If we ensure

that all objects with a predecessor come first in the enumeration $\{x_1, \dots, x_n\}$, then we can skip the recursive calls for all $\{x_j\}^{\mathcal{I}\mathcal{T}}$ where x_j has no predecessor—because in the subsequent recursive calls only further objects without predecessor could be added to it.

Furthermore, we only need to compute pseudo-intents \mathbf{C} that are premises of witnessed CIs, i.e. for which $(\prod \mathbf{C})^{\mathcal{I}} = \mathbf{C}^{\mathcal{I}}$ is non-empty. This is a monotonous property: if $\mathbf{C}^{\mathcal{I}}$ is empty, then also $\mathbf{E}^{\mathcal{I}}$ is empty for each superset $\mathbf{E} \supseteq \mathbf{C}$. Since the algorithm LinCbO enumerates the pseudo-intents in a sub-order of set inclusion \subseteq , we can stop a computation branch as soon as such a pseudo-intent \mathbf{C} with $\mathbf{C}^{\mathcal{I}} = \emptyset$ has been found.

In a similar way, we incorporate a conjunction size limit (here for the top-level conjunctions in the CI base, whereas for inner conjunctions the limit is taken into account during the computation of all MMSCDs in Section 4.3). Such a limit is monotonous as well: if \mathbf{C} contains more than ℓ attributes, then also every superset. Thus, we dispense with processing supersets and stop the respective computation branch as soon as a pseudo-intent \mathbf{C} with $|\mathbf{C}| > \ell$ has been found.

5.2 Fast Disjointness Axioms

If, instead, we want a CI base that is still complete for disjointness axioms but which need not be minimal, then we compute the attribute set \mathbf{M} as usual, but before building the induced context $\mathbb{K}_{\mathcal{I}}$ from it we remove every CD C not satisfied in \mathcal{I} , except \perp , and store the *fast disjointness axiom* $C \sqsubseteq \perp$ in an intermediate set that we will later add to the computed CI base. Since thereby the size of $\mathbb{K}_{\mathcal{I}}$ is often significantly reduced, the computation of the canonical implication base is much faster. The downside is, however, that the final CI base is larger.

Denote by $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ the sub-context of $\mathbb{K}_{\mathcal{I}}$ with attribute set $\mathbf{M}|_{\text{Fast}} := \{\perp\} \cup \{C \mid C \in \mathbf{M} \text{ and } C^{\mathcal{I}} \neq \emptyset\}$, let $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$ consist of all implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ that use only attributes from $\mathbf{M}|_{\text{Fast}}$, and define the set of *fast disjointness axioms* as $\text{FastDA}(\mathcal{I}) := \{C \sqsubseteq \perp \mid C \in \mathbf{M} \setminus \mathbf{M}|_{\text{Fast}}\}$. From this sub-context we compute the *fast CI base* $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T}) := \prod \text{Can}(\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}, \mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}) \cup \text{FastDA}(\mathcal{I})$.

Proposition XXXIV. *$\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$ is a CI base of \mathcal{I} relative to \mathcal{T} , though it need not be minimal.*

Proof. We first show that the canonical CI base $\text{Can}(\mathcal{I}, \mathcal{T})$ from Theorem 10 and the fast CI base $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$ are equivalent.

Each CI in the fast CI base is satisfied in \mathcal{I} and thus follows from the canonical CI base, for its completeness.

In the converse direction, we show that the fast CI base entails each CI in the canonical CI base. We denote by L the incidence relation of the sub-context $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$, i.e. $L := I \cap (\text{Dom}(\mathcal{I}) \times \mathbf{M}|_{\text{Fast}})$.

Similar as in the proof of Lemma XXXIII, a set $\mathbf{C} \subseteq \mathbf{M}$ with $(\prod \mathbf{C})^{\mathcal{I}} \neq \emptyset$ is a pseudo-intent of $\mathbb{K}_{\mathcal{I}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ iff. it is a pseudo-intent of the sub-context $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$. We conclude that each CI $\prod \mathbf{C} \sqsubseteq \prod \mathbf{D}$ in $\text{Can}(\mathcal{I}, \mathcal{T})$ where $(\prod \mathbf{C})^{\mathcal{I}} \neq \emptyset$ is also contained in $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$.

Now consider a CI $\prod \mathbf{C} \sqsubseteq \prod \mathbf{D}$ in $\text{Can}(\mathcal{I}, \mathcal{T})$ where $(\prod \mathbf{C})^{\mathcal{I}} = \emptyset$. In particular, \mathbf{C} is a pseudo-intent of $\mathbb{K}_{\mathcal{I}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ and $\mathbf{D} = \mathbf{C}^{\mathcal{I}\mathcal{I}}$. Since $\mathbf{C}^{\mathcal{I}} = (\prod \mathbf{C})^{\mathcal{I}}$, we obtain $\mathbf{C}^{\mathcal{I}} = \emptyset$ and thus $\mathbf{C}^{\mathcal{I}\mathcal{I}} = \mathbf{M}$. It follows that $\perp \in \mathbf{C}^{\mathcal{I}\mathcal{I}}$ and so $\prod \mathbf{C} \sqsubseteq \prod \mathbf{D}$ is (equivalent to) the disjointness axiom $\prod \mathbf{C} \sqsubseteq \perp$. We distinguish the following cases.

- If \mathbf{C} is no subset of $\mathbf{M}|_{\text{Fast}}$, then there is some atom $C \in \mathbf{C}$ with $C^{\mathcal{I}} = \emptyset$. Thus $C \sqsubseteq \perp$ is a fast disjointness axiom, and we conclude that $\prod \mathbf{C} \sqsubseteq \prod \mathbf{D}$ follows from $\text{FastDA}(\mathcal{I})$ and thus also from $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$.
- We proceed with the case where \mathbf{C} is a subset of $\mathbf{M}|_{\text{Fast}}$. The Statements (α) and (β) from the proof of Lemma XXXIII hold similarly, and we will implicitly use them in the following. It follows that $\mathbf{C}^{\mathcal{I}} = \mathbf{C}^{\mathcal{L}}$. If there is a pseudo-intent $\mathbf{Q} \subset \mathbf{C}$ of $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$ with $\mathbf{Q}^{\mathcal{I}} = \emptyset$, then also $\mathbf{Q}^{\mathcal{L}}$ is empty, i.e. $\perp \in \mathbf{Q}^{\mathcal{L}\mathcal{L}}$. In this case, $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$ contains, modulo equivalence, the CI $\prod \mathbf{Q} \sqsubseteq \perp$. Since \mathbf{Q} is a subset of \mathbf{C} , the CI $\prod \mathbf{C} \sqsubseteq \perp$ follows from the latter.
- Otherwise, we will verify that \mathbf{C} is also a pseudo-intent of $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$.

(PI1) Recall that $\mathbf{C}^{\mathcal{I}} = \emptyset$, which implies $\mathbf{C}^{\mathcal{L}} = \emptyset$. Then $\mathbf{C}^{\mathcal{L}\mathcal{L}} = \mathbf{M}|_{\text{Fast}}$, i.e. \perp is contained in $\mathbf{C}^{\mathcal{L}\mathcal{L}}$. We show that \mathbf{C} cannot contain \perp , from which $\mathbf{C} \neq \mathbf{C}^{\mathcal{L}\mathcal{L}}$ follows. Assume the contrary. Since $\{\perp\} \rightarrow \{F\}$ for all $F \in \mathbf{M}$ are background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ and \mathbf{C} is closed under all background implications, it would follow that \mathbf{M} is a subset of \mathbf{C} , i.e. \mathbf{C} and \mathbf{M} are actually equal. But then \mathbf{C} would be an intent of $\mathbb{K}_{\mathcal{I}}$, namely the largest one, a contradiction to (PI1).

(PI2) By assumption, \mathbf{C} is closed under the background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}$. It follows that \mathbf{C} is also closed under the subset $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$.

(PI3) Consider a pseudo-intent $\mathbf{Q} \subset \mathbf{C}$ of $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$. Then $\mathbf{Q}^{\mathcal{I}} \neq \emptyset$ (otherwise we would be in the previous case), and so \mathbf{Q} is also a pseudo-intent of $\mathbb{K}_{\mathcal{I}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ (see above). (PI3) yields $\mathbf{Q}^{\mathcal{I}\mathcal{I}} \subseteq \mathbf{C}$. Similarly as in the proof of Lemma XXXIII we infer that $\mathbf{Q}^{\mathcal{I}\mathcal{I}} = \mathbf{Q}^{\mathcal{L}\mathcal{L}}$, and thus $\mathbf{Q}^{\mathcal{L}\mathcal{L}} \subseteq \mathbf{C}$.

We conclude that $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$ contains the CI $\prod \mathbf{C} \sqsubseteq \prod \mathbf{C}^{\mathcal{L}\mathcal{L}}$ where $\perp \in \mathbf{C}^{\mathcal{L}\mathcal{L}}$, which entails $\prod \mathbf{C} \sqsubseteq \perp$.

Last, we give an example where the fast CI base $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$ is no minimal CI base. We choose a signature with CNs A, B, C and without RNs. As interpretation, we take \mathcal{I} with $\text{Dom}(\mathcal{I}) := \{x, y\}$ and $\cdot^{\mathcal{I}} := \{x : A, y : B\}$. The existing knowledge is contained in the TBox $\mathcal{T} := \{C \sqsubseteq A \sqcap B\}$, which has \mathcal{I} as a model. In order to construct the induced context $\mathbb{K}_{\mathcal{I}}$, we do not need to compute any MMSCDs since there are not RNs. We obtain the following context.

$\mathbb{K}_{\mathcal{I}}$	A	B	C	\perp
x	×	•	•	•
y	•	×	•	•

The background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ are $\{C\} \rightarrow \{A, B\}$ and $\{\perp\} \rightarrow \{A, B, C\}$. The canonical implication base of $\mathbb{K}_{\mathcal{I}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ contains only the implication $\{A, B\} \rightarrow \{A, B, C, \perp\}$, which is transformed into the CI $A \sqcap B \sqsubseteq \perp$. We obtain the canonical CI base $\{A \sqcap B \sqsubseteq \perp\}$.

Now, the only fast disjointness axiom here is $C \sqsubseteq \perp$. The restricted context $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ is $\mathbb{K}_{\mathcal{I}}$ without the attribute C , and the restricted implication set $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$ is $\{\{\perp\} \rightarrow \{A, B\}\}$. The canonical implication base of $\mathbb{K}_{\mathcal{I}}|_{\text{Fast}}$ w.r.t. $\mathcal{L}_{\mathcal{I},\mathcal{T}}|_{\text{Fast}}$ consists of the single implication $\{A, B\} \rightarrow \{A, B, \perp\}$, which is transformed into $A \sqcap B \sqsubseteq \perp$. The fast CI base $\text{Can}_{\text{Fast}}(\mathcal{I}, \mathcal{T})$ is thus $\{A \sqcap B \sqsubseteq \perp, C \sqsubseteq \perp\}$. It is not minimal since the fast disjointness axiom $C \sqsubseteq \perp$ already follows from $A \sqcap B \sqsubseteq \perp$ and \mathcal{T} . \square

5.3 Bounding the Role Depth

Apart from bounding the conjunction size, another effective way to avoid the axiomatization of impractically huge CIs is to limit the role depth. Specifically, the *role depth* of an \mathcal{EL} CD is the maximal number of nestings of existential restrictions. By modifications to the approach in Section 3.2, we can also compute a CI base w.r.t. a role-depth bound $n \geq 0$ which is, however, only guaranteed to be complete for all CIs bounded by n . The case without a known TBox \mathcal{T} has already been considered (Borchmann, Distel, Kriegel, 2016). We show how such an existing TBox can be taken into account, yielding a minimal CI base as for the unrestricted case if \mathcal{T} also satisfies the role-depth bound n .

It is a finger exercise to verify that all results in Section 3.2 hold for the role-depth bounded case when all employed notions are replaced by their role-depth bounded variants:

- For each subset $X \subseteq \text{Dom}(\mathcal{I})$, the MMSCD of X for role-depth bound n is denoted as $X^{\mathcal{I}_n}$, and it is obtained by unfolding the unbounded MMSCD: $X^{\mathcal{I}_n} \equiv^{\emptyset} X^{\mathcal{I}}|_n$.
- We represent a bounded MMSCD $X^{\mathcal{I}_n}$ by layered copies of the powering. We therefore construct the interpretation \mathcal{J} that contains all pairs (X, k) in its domain where X is in the domain of $\wp(\mathcal{I})$ and $0 \leq k \leq n$, and its extension function labels (X, k) with the CN A if X is labelled with A in $\wp(\mathcal{I})$ and connects (X, k) and (Y, ℓ) with the RN r if X is connected to Y by r in $\wp(\mathcal{I})$ and $k+1 = \ell$. With that, the bounded MMSCD $X^{\mathcal{I}_n}$ is equivalent to the CD $\exists^{\text{sim}}(\mathcal{J}, (X, 0))$, which is exponential in $\text{Dom}(\mathcal{I})$ and polynomial in n . Moreover, as it suffices to know all pairs reachable from $(X, 0)$, the powering $\wp(\mathcal{I})$ needs to be built only up to depth n , starting with X .
- As attribute set we now use \mathbf{M}_n consisting of \perp , all CNs, and all $\exists r.X^{\mathcal{I}_{n-1}}$ if $n > 0$ (instead of $\exists r.X^{\mathcal{I}}$). The induced context with attribute set \mathbf{M}_n is denoted by $\mathbb{K}_{\mathcal{I},n}$, and the background implication set by $\mathcal{L}_{\mathcal{I},\mathcal{T},n}$. When the incidence relation I is filled, we add a pair $(x, \exists r.X^{\mathcal{I}_n})$ if $r^{\mathcal{I}}(x) \cap \wp_{\mathcal{I},\mathcal{T}}((X, 0)) \neq \emptyset$.
- Also to the partial closure we apply the role-depth bound and use the unfolding $C^{[\mathcal{I}\mathcal{I}]|_n}$.
- Only the adaptation of Lemma XV is not obvious. In the proof, reconsider the part where we use the sequence of applications of the \sqsubseteq_+ -Rule and the \sqsubseteq_{\perp} -Rule that

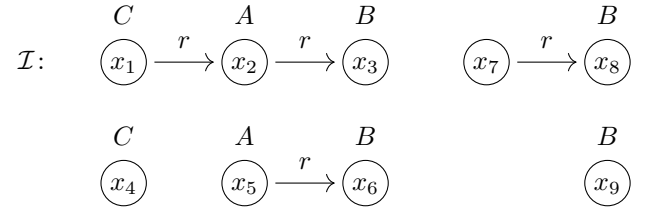
produces from $\sqcap \mathbf{X}$ a subsumee of $\sqcap \mathbf{Y}$ to verify that $\mathcal{B}_S \cup \mathcal{L}_{\mathcal{I},\mathcal{T}}$ entails $\mathbf{X} \rightarrow \mathbf{Y}$. Now, the two rules might also be applicable above the root, but these applications are irrelevant since all involved concepts are already closed w.r.t. \mathcal{I} up to depth n and thus additional information could be added only at deeper levels (as \mathcal{I} models \mathcal{T}).

Altogether we obtain the following variant of Theorem 10.

Proposition XXXV. *The TBox $\sqcap \text{Can}(\mathbb{K}_{\mathcal{I},n}, \mathcal{L}_{\mathcal{I},\mathcal{T},n})$ is a CI base of \mathcal{I} relative to \mathcal{T} for role depth n and is computable in time that is exponential in $\text{Dom}(\mathcal{I})$ and polynomial in \mathcal{T} and n . If all CIs in \mathcal{T} are of the form $C \sqsubseteq D^{[\mathcal{I}\mathcal{I}]|_n}$ and bounded by n , then it contains the fewest CIs among all CI bases of \mathcal{I} relative to \mathcal{T} for n . Furthermore, there are finite interpretations that have no polynomial-size CI base for n .*

The following example shows that $\sqcap \text{Can}(\mathbb{K}_{\mathcal{I},n}, \mathcal{L}_{\mathcal{I},\mathcal{T},n})$ need not be a minimal CI base for role depth n if not all CIs in \mathcal{T} are bounded by n , even though \mathcal{T} contains only CIs $C \sqsubseteq D$ with $D \sqsubseteq^{\emptyset} D^{[\mathcal{I}\mathcal{I}]|_n}$.

Example XXXVI. Consider the following interpretation \mathcal{I} .



We further have the TBox $\mathcal{T} := \{\exists r.\exists r.B \sqsubseteq C\}$, and we consider the role-depth bound $n := 1$. Computing the induced context $\mathbb{K}_{\mathcal{I},n}$ is rather simple since we only need to determine all MMSCDs for role depth $n-1$, which are \top , A , B , C and thus $\mathbb{K}_{\mathcal{I},n}$ has the attributes $\exists r.\top$, $\exists r.A$, $\exists r.B$, and $\exists r.C$ in addition to \perp and all CNs.

$\mathbb{K}_{\mathcal{I},n}$	\perp	A	B	C	$\exists r.\top$	$\exists r.A$	$\exists r.B$	$\exists r.C$
x_1	•	•	•	×	×	×	•	•
x_2	•	×	•	•	×	•	×	•
x_3	•	•	×	•	•	•	•	•
x_4	•	•	•	×	•	•	•	•
x_5	•	×	•	•	×	•	×	•
x_6	•	•	×	•	•	•	•	•
x_7	•	•	•	•	×	•	×	•
x_8	•	•	×	•	•	•	•	•
x_9	•	•	×	•	•	•	•	•

The background implication set $\mathcal{L}_{\mathcal{I},\mathcal{T},n}$ is empty since \mathcal{T} does not entail any non-trivial CIs between conjunctions over these attributes. Ignoring disjointness axioms, the CI base $\sqcap \text{Can}(\mathbb{K}_{\mathcal{I},n}, \mathcal{L}_{\mathcal{I},\mathcal{T},n})$ is $\{A \sqsubseteq \exists r.B, \exists r.A \sqsubseteq C\}$. The latter CI is redundant since it follows from $A \sqsubseteq \exists r.B$ and $\exists r.\exists r.B \sqsubseteq C$.

Even if we add all top-level conjuncts in CIs in \mathcal{T} to the attribute set, which here yields the additional attribute $\exists r. \exists r. B$ only, the obtained CI base need not be minimal. Although we can now capture the CI $\exists r. \exists r. B \sqsubseteq C$ in \mathcal{T} by means of the background implication $\{\exists r. \exists r. B\} \rightarrow \{C\}$, there are two problems. On the one hand, the induced context now contains attributes with a role depth exceeding the bound and these will be contained in the CIs. On the other hand, the implication base of the extended induced context $\mathbb{K}_{\mathcal{I}, n}$ would now contain the implications $\{A\} \rightarrow \{\exists r. B\}$ and $\{\exists r. A\} \rightarrow \{\exists r. \exists r. B, C\}$, but after transformation into CIs the second is still redundant. It might be a solution to update the background implication set $\mathcal{L}_{\mathcal{I}, \mathcal{T}, n}$ according to the transformation of each computed implication into a CI. Here, we would add the background implication $\{\exists r. A\} \rightarrow \{\exists r. \exists r. B\}$ when $\{A\} \rightarrow \{\exists r. B\}$ has been computed. Then $\{\exists r. A\}$ would not be a pseudo-intent anymore (and thus no premise of an implication in the base) since its closure under the updated $\mathcal{L}_{\mathcal{I}, \mathcal{T}, n}$ would be $\{\exists r. A, \exists r. \exists r. B, C\}$. We leave this as future research.

Our last main result is the following.

Theorem 18. *Given a finite interpretation \mathcal{I} , an $\mathcal{EL}_{\text{SI}}^{\perp}$ TBox \mathcal{T} of which \mathcal{I} is a model, and a number $n \geq 0$, then a CI base of \mathcal{I} relative to \mathcal{T} for role depth n can be computed in time that is exponential in $\text{Dom}(\mathcal{I})$ and polynomial in \mathcal{T} and n . If all CIs in \mathcal{T} are of the form $C \sqsubseteq D^{[\mathcal{I}]} \upharpoonright_n$ and bounded by n , then it contains the fewest CIs among all CI bases of \mathcal{I} relative to \mathcal{T} for n . Furthermore, there are finite interpretations of which no CI base for n has polynomial size.*

As an application of Theorem 18, we can keep the CIs in a base small by iteratively axiomatizing CIs from a given interpretation \mathcal{I} . We therefore increase the role-depth bound in each step (starting with 0) and take all CIs in \mathcal{T} as well as the CIs from all previous steps as background knowledge. This guarantees a CI base that is complete for all CIs when the role-depth bound $2^{|\text{Dom}(\mathcal{I})|} \cdot |\text{Dom}(\mathcal{I})| + 1$ has been reached (Baader, Distel, 2008). Alternatively, we could stop earlier and as last step compute the canonical CI base from Theorem 10 relative to \mathcal{T} and all CIs from the previous steps.

6 Experimental Evaluation

We implemented [9] the axiomatization method in the programming language Scala 3 [10] and we evaluate the prototype with the plethora of ontologies [11] from real-world applications used in the ORE 2015 Reasoner Competition (Parsia, Matentzoglou, Gonçalves, Glimm, Steigmiller, 2017). This collection is split into OWL 2 EL and OWL 2 DL ontologies. The former cannot contain any CIs not expressible in \mathcal{EL} . For the latter, we syntactically transform as many axioms as possible into \mathcal{EL} and remove the others. There is no best way to do this since optimal finite \mathcal{EL} approximations need not exist (Haga, Lutz, Marti, Wolter, 2020). Removal of unsupported axioms makes these ontologies weaker in the sense that some logical consequences are lost; however, no new, undesired consequences are thereby introduced. Each test dataset is derived from such an ontology, viz. we treat the ABox as interpretation \mathcal{I} (under closed-world assumption) and the TBox \mathcal{T} as existing knowledge.

In general, the so obtained interpretation \mathcal{I} need not be a model of the TBox \mathcal{T} . In order to fulfill the preconditions of Theorem 10, we saturate \mathcal{I} by means of the \sqsubseteq_{+} -Rule and the \sqsubseteq_{\perp} -Rule from Section 2.3 for the CIs in \mathcal{T} . This was possible for all test datasets, i.e. the \sqsubseteq_{\perp} -Rule never failed. According to Proposition IV and Lemma VI, the saturated interpretation $\text{sat}(\mathcal{I}, \mathcal{T}_{+})$ contains \mathcal{I} as a sub-interpretation and is a model of \mathcal{T} . Now, it depends on the point of view whether one wants to take the newly added domain elements into account during axiomatization. We decided to ignore these, and therefore slightly deviate from Theorem 10 in that we restrict the induced context of the saturation $\text{sat}(\mathcal{I}, \mathcal{T}_{+})$ to the object set $\text{Dom}(\mathcal{I})$. Technically, this means that we axiomatize the closure operator $C \mapsto (C^{\text{sat}(\mathcal{I}, \mathcal{T}_{+})} \cap \text{Dom}(\mathcal{I}))^{\text{sat}(\mathcal{I}, \mathcal{T}_{+})}$ instead (Kriegel, 2019a).

Altogether we obtain 614 test datasets with up to 747,998 objects, of which 446 (72.64%) are acyclic. The average number of triples per object varies from slightly over 0 up to 25.39.

The prototype supports three modes in which disjointness axioms are not computed (None) as per Section 5.1, computed in the fast way (Fast) as per Section 5.2, or computed in the canonical way (Can.) as per Theorem 10. It further allows for specifying a role depth bound and a conjunction size limit. During the experiment, we used all three modes and the settings (0, 32), (1, 8), (1, 32), (2, 32), (∞ , 32), (∞ , ∞) where the first parameter is the role depth bound and the second is the conjunction size limit. For every dataset, the prototype was executed once for each configuration (mode and parameters).

There are three types of failures that can occur during computation: timeouts (limit: 8 hours), out-of-memory errors (limit: 80 GB), and powering-too-large exceptions (conjunction size limit: 10,000,000). In order to save unnecessary computation time resulting in the same failure, we assumed the following order \leq on the configurations and skipped subsequent computations with the same dataset when a failure occurred for a smaller configuration: $\text{None}(x, y) < \text{Fast}(x, y) < \text{Can.}(x, y)$, and $M(0, 32) < M(1, 32) < M(2, 32) < M(\infty, 32) < M(\infty, \infty)$, and $M(1, 8) < M(1, 32)$. In the statistics, the failure is then inherited by all larger configurations.

The prototype is implemented according to presented details. It uses Java's Fork/Join Framework [12] to execute concurrent computation tasks and therefore all used data structures must support concurrent read/write access by multiple threads. The standard libraries in Java and Scala already contain many suitable thread-safe collections, but we also needed to implement a `ConcurrentBitSet` to represent subsets efficiently. However, the prototype differs in the following aspects from this extended version.

- During the computation of all MMSCDs as per Section 4.3, it uses the direct generators $g(Y)$ instead of the generators $g^*(Y)$. The reason is that we found the latter only later, when there was no time left to repeat the experiments. Since the generators $g^*(Y)$ are subsets of the respective direct generators $g(Y)$, we expect a performance gain when the prototype is updated.

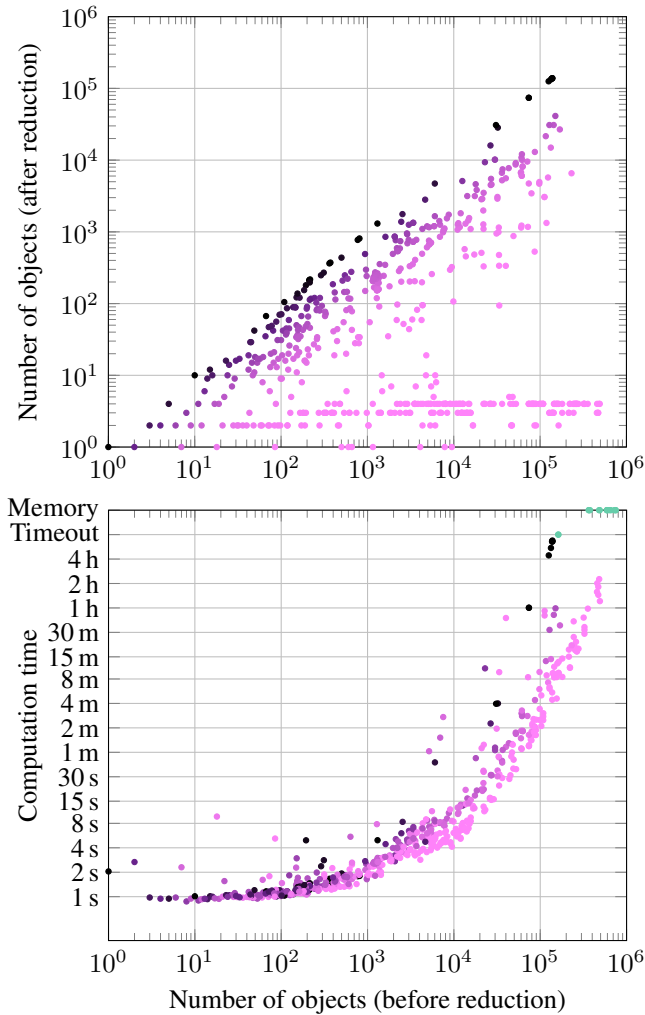


Figure 1: Computing reductions of the test datasets

- The prototype uses a background implication set different from $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ in Section 3.2. In an earlier version of this article and its accompanying report, we used a rather straightforward definition of the background implication set that was easy to handle in proofs but was not suitable in computation for its exponential size w.r.t. the attribute set M . We resolved this bottleneck by developing a polynomial-size set of background implications that can be computed efficiently, though uses auxiliary attributes and needs a rather long correctness proof, see Appendix A. The presented background implication set $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$ in Section 3.2 is both easy to handle in proofs and can be computed efficiently. We do not expect a relevant difference in performance as both implication sets have similar size and can be computed in similar ways.
- Axiomatization of RRs and RIs was not implemented.

The experiments were run on a small, old computer server with two Intel Xeon E5-2640 processors (each with 6 CPU cores, hyper-threading, 15 MB cache, 2.80 GHz frequency, boost up to 3.00 GHz) and 96 GB DDR3-SDRAM main

memory. Modern laptops have faster processors but usually only a smaller amount of main memory. As runtime environment we used Oracle GraalVM EE 22.3.0 (Java 19.0.1).

The only other prototype for axiomatizing graph datasets that we are aware of is the one by Daniel Borchmann [13]. It was used in (Borchmann, Distel, Kriegel, 2016) for a case study with a fragment of DBpedia (with only one role name). However, no resource consumption was measured. For the particular programming language used, we were not able to conduct a performance comparison. We do expect that our implementation is faster, due to many optimizations, efficient data structures, and multi-threading, but also for the modern FCA algorithms employed.

Computing the Weak Reduction. For 599 (97.56 %) of the 614 test datasets the prototype successfully computed the (weak) reduction. Figure 1 shows computation times as well as size changes. In many cases, the number of objects was significantly reduced and often by more than one order of magnitude. Several reductions contain fewer than ten objects, meaning that there is only a small variety of different types of objects. We ignored these for the subsequent experiment steps. Reductions could not be computed for 13 (2.12 %) larger datasets with more than 300,000 objects due to out-of-memory errors (limit: 80 GB), and for 2 (0.33 %) datasets due to timeouts (limit: 8 hours). Note that in these cases the maximal simulation could contain more than $300,000 \cdot 300,000$ pairs, which amounts to more than 83 GB if only one bit is used to represent containment of a pair (like in the prototype, plus some metadata).

Computing all MMSCDs. Figures I and XVIII show the numbers of MMSCDs computed for the test datasets, and Figures II, XIX and XX show the computation times. We observe that in more restricted settings fewer MMSCDs are obtained in less time. This is a consequence of the smaller and fewer CDs that can be used to differentiate the objects.

Computing the Induced Context. In Figures III and XXI we see the number of attributes in the induced contexts $\mathbb{K}_{\mathcal{I}}$ for the test datasets, and Figures IV, XXII and XXIII show the computation times for $\mathbb{K}_{\mathcal{I}}$ (from the set of all MMSCDs computed in the previous step). As expected, computing the incidence relation is cheap. Moreover, we see that $\mathbb{K}_{\mathcal{I}}$ contains significantly more attributes when disjointness axioms are to be computed in the canonical way (in order to get a minimal CI base), but for the other two modes the induced contexts do not differ in size (ignoring \perp). Of course, the fewer MMSCDs have been computed in the previous step due to more restricted parameters, the smaller $\mathbb{K}_{\mathcal{I}}$ is and the faster it can be computed.

Computing the Background Implications. Figures V and XXIV show the number of background implications computed for the test datasets. There is no clear correlation with the number of objects in the reduction, but instead with the number of CIs in the given TBox \mathcal{T} , see Figure VII, as well as with the size of the underlying signature, see Figure VIII. As shown in Figures VI, XXV and XXVI, computing $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ is quite cheap and the computation times seem to be highly correlated with the domain size of the reduction.

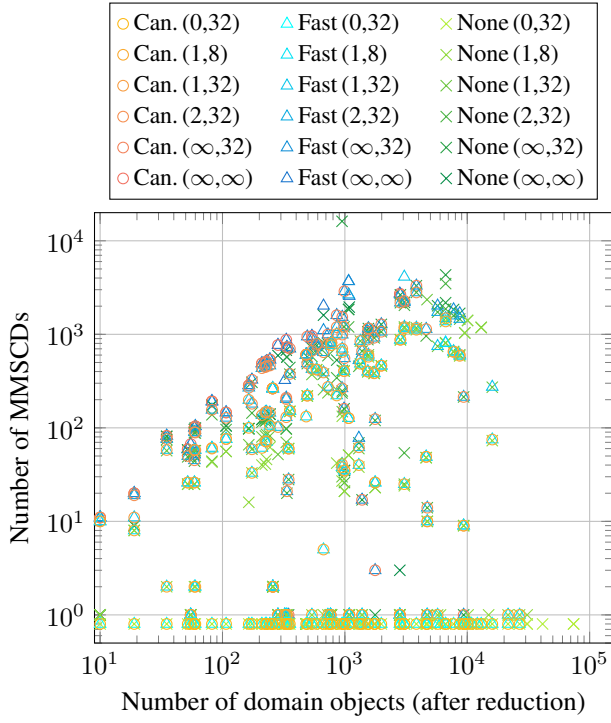


Figure I: Computing all MMSCDs of \mathcal{I}

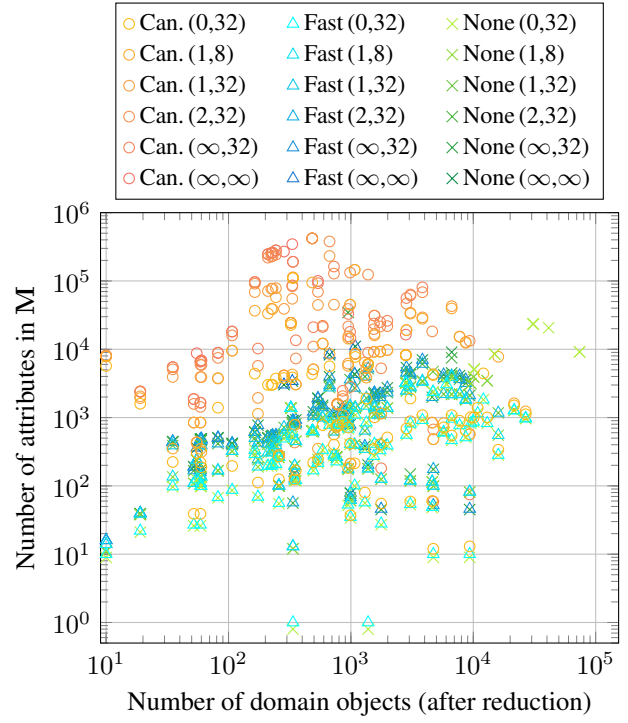


Figure III: Computing the induced context $\mathbb{K}_{\mathcal{I}}$

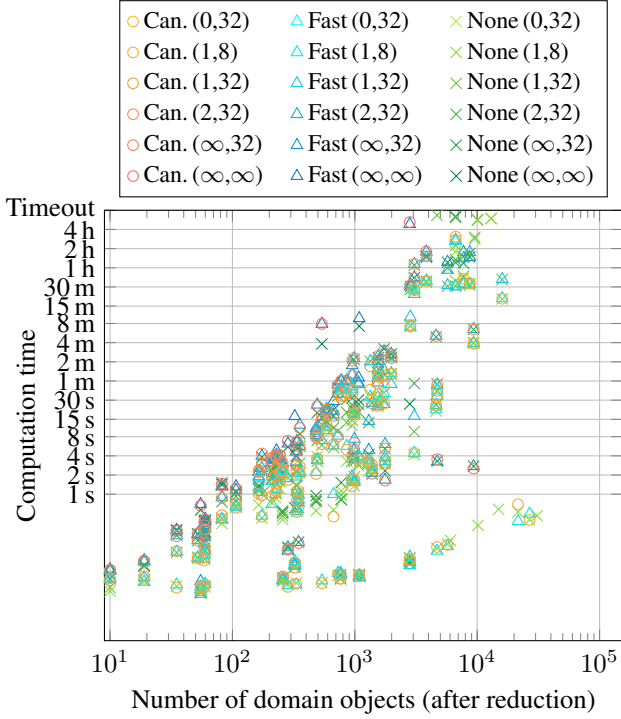


Figure II: Computing all MMSCDs of \mathcal{I}

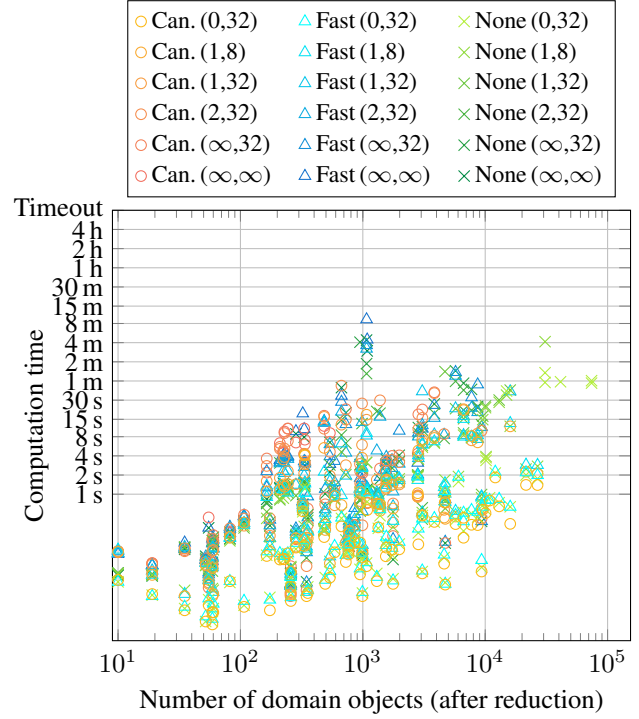


Figure IV: Computing the induced context $\mathbb{K}_{\mathcal{I}}$

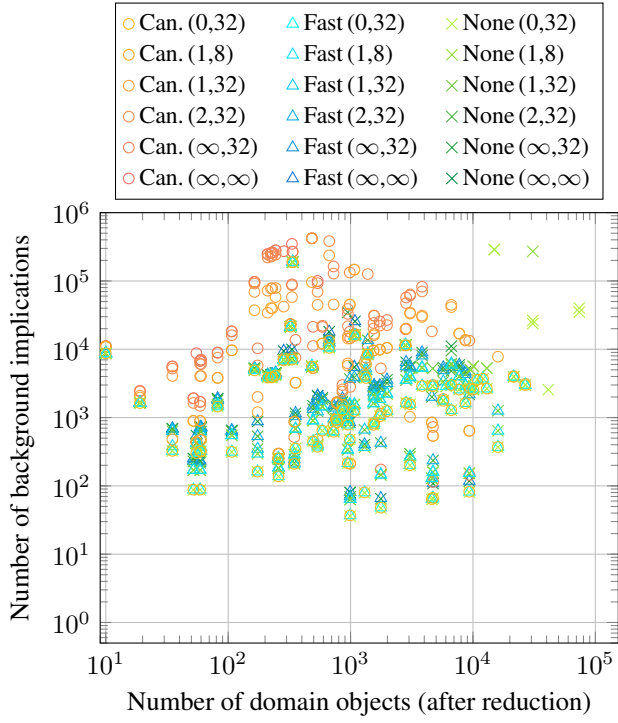


Figure V: Computing all background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

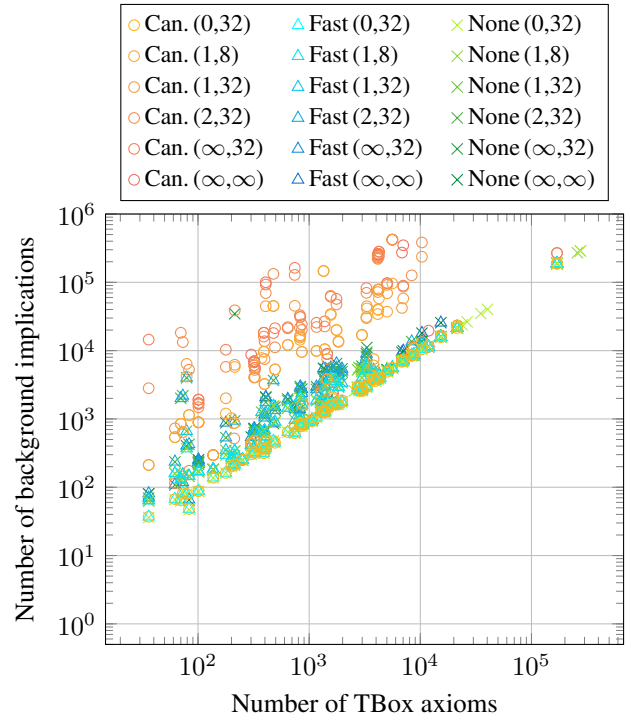


Figure VII: Computing all background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

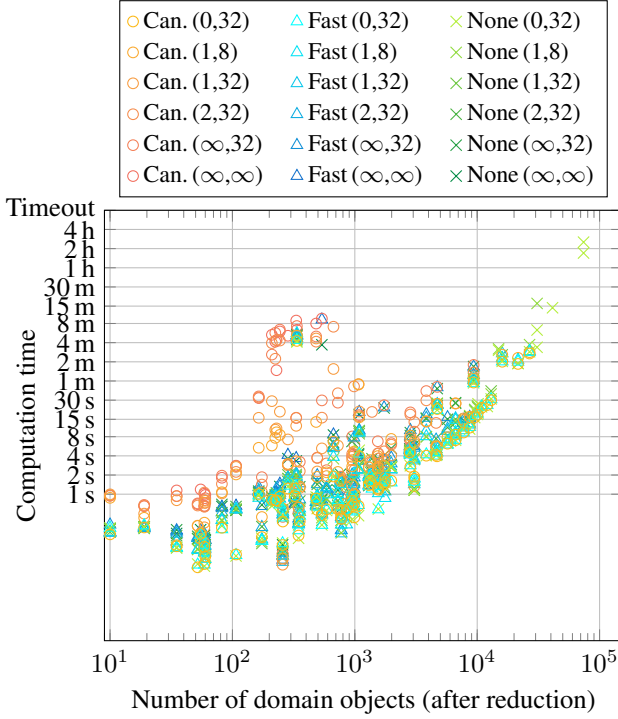


Figure VI: Computing all background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

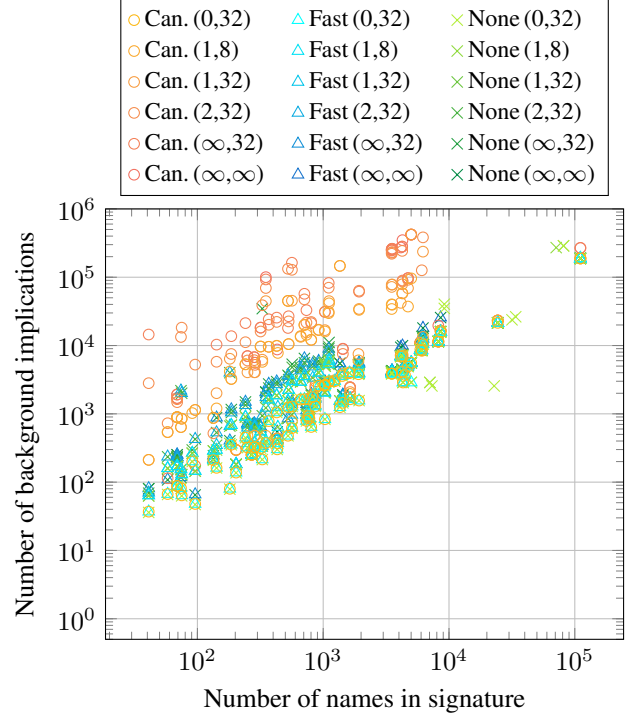


Figure VIII: Computing all background impl. in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

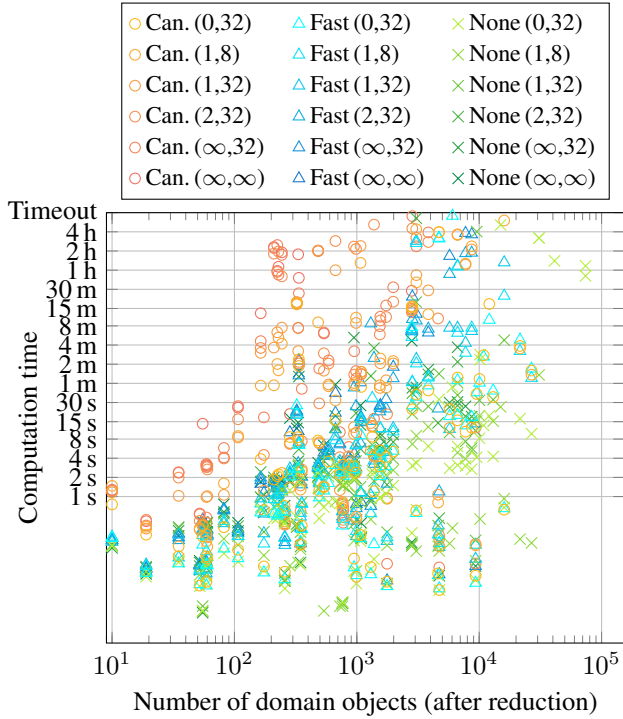


Figure IX: Computing the implication base $\text{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}}^*)$

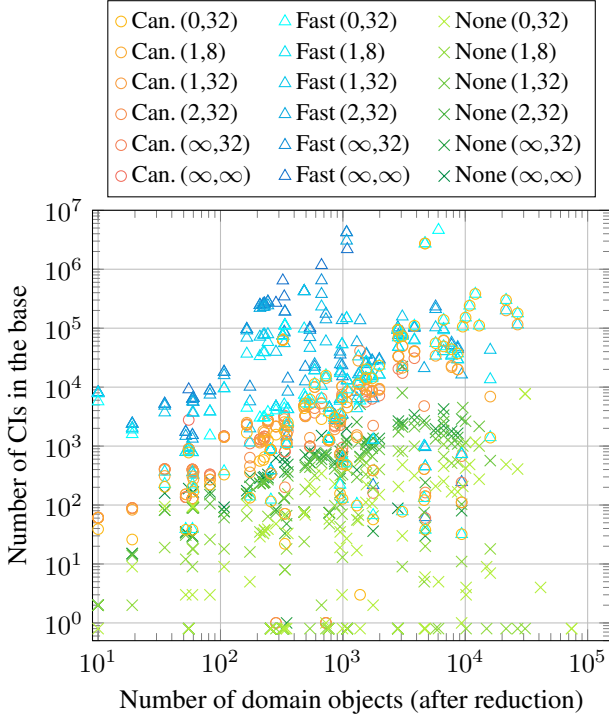


Figure X: Computing the CI base $\text{Can}(\mathcal{I}, \mathcal{T})$

Computing the CI Base. In Figures IX, XXVIII and XXIX we see the computation times needed for computing the canonical implication base. Furthermore, Figures X and XXVII show the number of implications/CIs in the base (including the fast disjointness axioms). Unsurprisingly, the mode without disjointness axioms is the fastest and produces the fewest CIs. The canonical mode without any modifications is the slowest (mainly because the induced contexts then become very large), but produces fewer CIs than the mode with fast disjointness axioms, i.e. there is a trade-off between computation time and cardinality of the base.

Total computation time. Figures XI to XIII show the total computation times for the CI bases (without reduction), including failures due to timeouts (limit: 8 hours), out-of-memory errors (limit: 80 GB), or powering-too-large exceptions (conjunction size limit: 10,000,000). The respective success rates and rates for the three failure types are gleaned in Tables XIV to XVII. However, we did not implement and measure the rewriting of the CI base into \mathcal{EL} , nor the axiomatization of RRs and RIs. Computation finished for all reduced datasets with no more than 100 objects. For reduced datasets with up to 1,000 objects, the first errors due to insufficient computing resources occurred without a role-depth bound. Between 1,000 and 10,000 objects, computations failed without restrictions, but otherwise succeeded in the majority of cases. Reduced datasets with more than 10,000 objects could only sometimes be axiomatized with very restricted settings, given 8 hours time and 80 GB memory.

In summary, we clearly see that computation resources can be saved if no disjointness axioms are wanted, or if they are computed in the fast way. Furthermore, the parameters allow us to control the overall resource consumption on the one hand, but also the size and number of the CIs in the final base on the other hand. We can avoid the computation of huge CIs that might not have any practical relevance.

Last, since the implemented method produces only correct axioms by design, all computed axioms are 100 % correct in the input dataset. It would instead make sense to measure completeness of the computed CI base since only without restrictions it is guaranteed to be complete by design. Since an ontology usually entails infinitely many axioms and thus counting does not work, it is unclear how the loss of completeness can be quantified. We leave the development of a suitable metric as future task.

7 Future Prospects

That the theoretical approach itself can be extended to more expressive DLs has already been proven, but it is unclear whether such an extended approach can still be efficiently implemented and used in practice. From the perspective of this article, this seems possible for DLs characterized by simulations, e.g. \mathcal{ELI} or Horn- \mathcal{ALC} .

Regarding the presented approach, an interesting question for future research would be whether one can give any kind of completeness guarantee if a conjunction size limit is used (e.g. every CI that also satisfies the limit is entailed). A smaller task can be to investigate how range restrictions and role inclusions can be integrated into the background know-

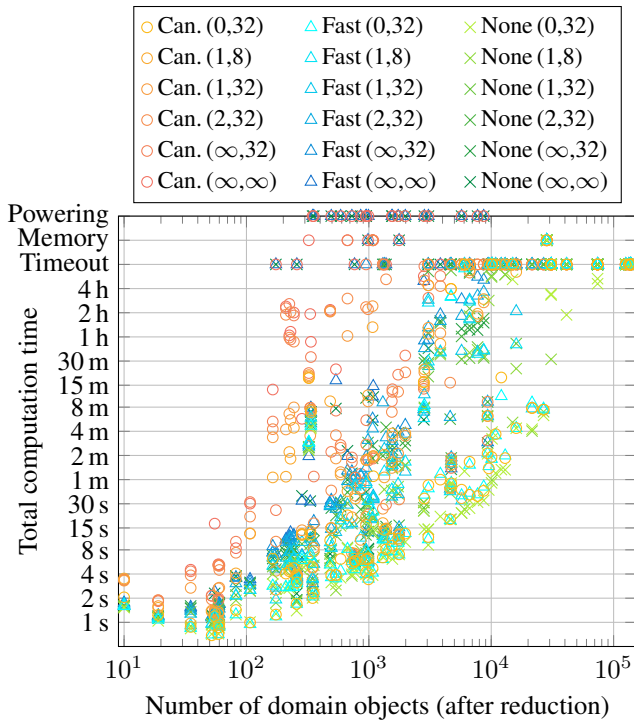


Figure XI: Total computation time (without reduction)

ledge after they have been computed but prior to axiomatizing the CIs, preferably yielding an overall minimal base.

It should be investigated whether the canonical CI base can be obtained more efficiently from the fast CI base by means of the algorithm in (Rudolph, 2007). The witnessed CI base ignores all disjointness axioms. One could restrict it even more by requiring that the number of objects satisfying a CI premise must exceed an absolute limit (e.g. at least ten objects) or relative limit (e.g. at least every twentieth object).

Furthermore, the computation can be speed-up with even faster FCA algorithms for enumerating closures. The employed LinCbO algorithm is currently the fastest algorithm for computing the canonical implication base, but it is unfortunately only single-threaded. Developing a multi-threaded variant is thus another future goal. It might already help to change its depth-first behaviour. Apart from that one could use a faster programming language (like C++), more computation time, a faster server, or optimize the prototype.

A CI $C \sqsubseteq D$ is *confident* if the ratio $|C \sqcap D|^{\mathcal{I}} / |C^{\mathcal{I}}|$ exceeds a pre-defined limit but need not be 100% (Borchmann, 2013). Since a confident CI base extends a canonical CI base by CIs of the form $X^{\mathcal{I}} \sqsubseteq Y^{\mathcal{I}}$, the prototype could be upgraded as it already computes all MMSCDs $X^{\mathcal{I}}$ and $Y^{\mathcal{I}}$.

We have not considered keys supported by the OWL 2 EL profile. Learning of keys from RDF data using FCA has been addressed in (Abbas, Bazin, David, Napoli, 2021, 2022; Atencia, David, Euzenat, Napoli, Vizzini, 2020). To apply this approach to DL and OWL it must be extended towards complex DL concepts in place of RDF classes.

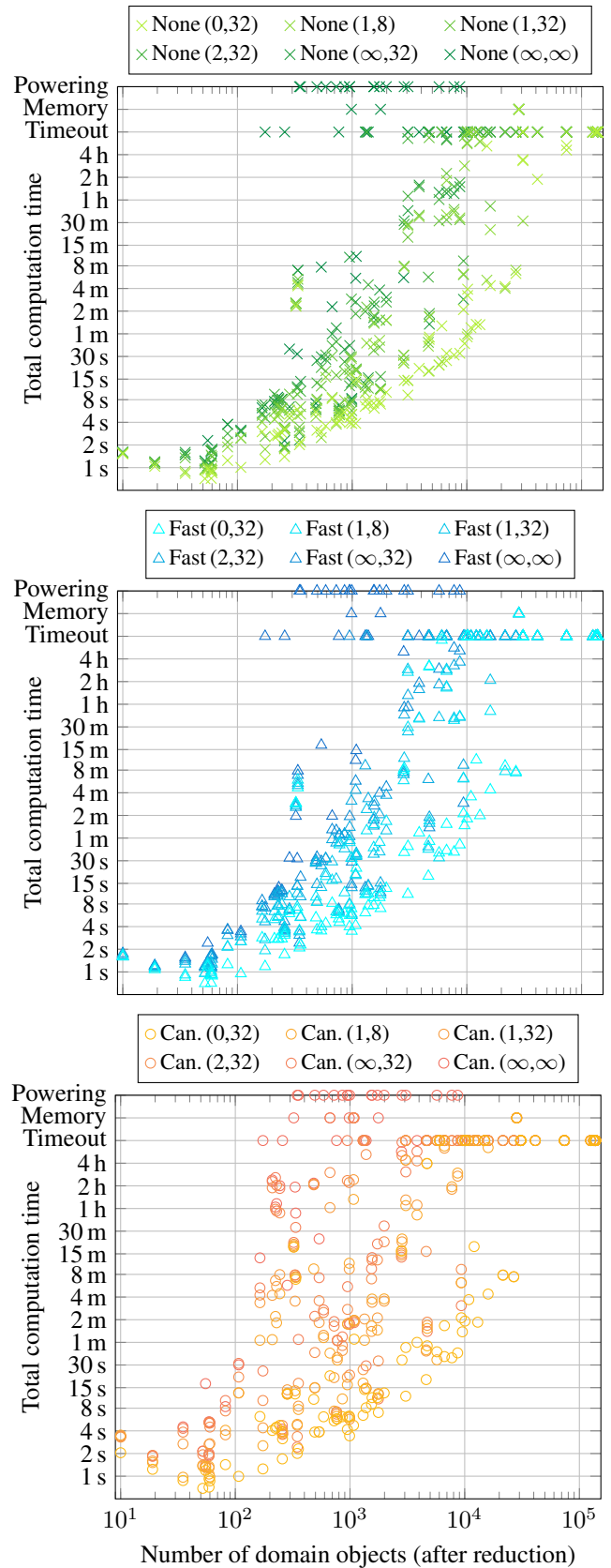


Figure XII: Total computation time (without reduction)

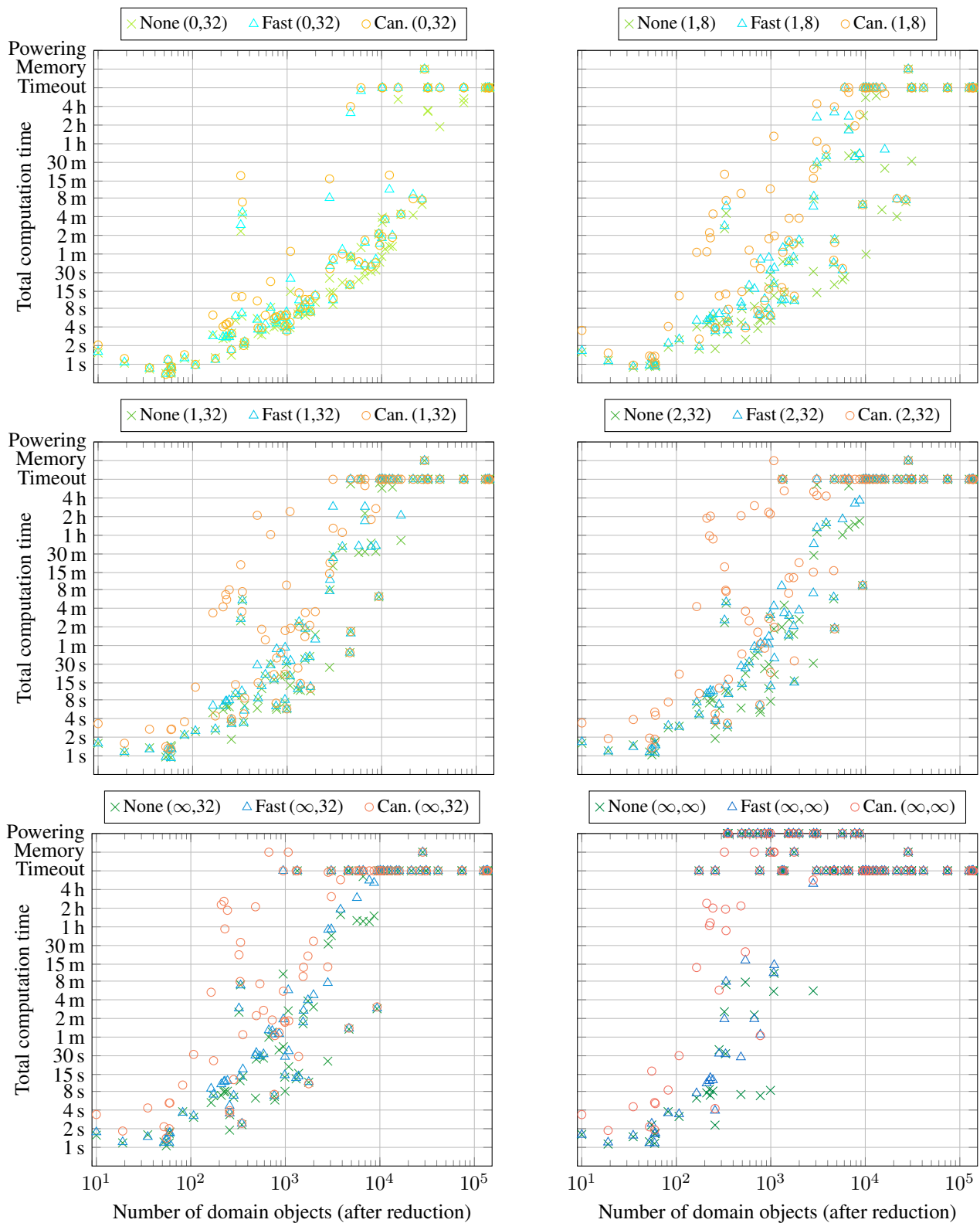


Figure XIII: Total computation time (without reduction)

	$\geq 10^1$ $< 10^2$	$\geq 10^2$ $< 10^3$	$\geq 10^3$ $< 10^4$	$\geq 10^4$
× None (0,32)	100.00 %	100.00 %	100.00 %	65.22 %
△ Fast (0,32)	100.00 %	100.00 %	100.00 %	30.43 %
○ Can. (0,32)	100.00 %	100.00 %	96.15 %	30.43 %
× None (1,8)	100.00 %	100.00 %	100.00 %	34.78 %
△ Fast (1,8)	100.00 %	100.00 %	92.31 %	13.04 %
○ Can. (1,8)	100.00 %	100.00 %	88.46 %	13.04 %
× None (1,32)	100.00 %	100.00 %	96.15 %	13.04 %
△ Fast (1,32)	100.00 %	100.00 %	88.46 %	4.35 %
○ Can. (1,32)	100.00 %	100.00 %	76.92 %	0.00 %
× None (2,32)	100.00 %	100.00 %	84.62 %	0.00 %
△ Fast (2,32)	100.00 %	100.00 %	73.08 %	0.00 %
○ Can. (2,32)	100.00 %	100.00 %	53.85 %	0.00 %
× None (∞ ,32)	100.00 %	100.00 %	76.92 %	0.00 %
△ Fast (∞ ,32)	100.00 %	96.43 %	69.23 %	0.00 %
○ Can. (∞ ,32)	100.00 %	92.86 %	50.00 %	0.00 %
× None (∞ , ∞)	100.00 %	57.14 %	11.54 %	0.00 %
△ Fast (∞ , ∞)	100.00 %	53.57 %	11.54 %	0.00 %
○ Can. (∞ , ∞)	100.00 %	46.43 %	3.85 %	0.00 %

Table XIV: Success rates

	$\geq 10^1$ $< 10^2$	$\geq 10^2$ $< 10^3$	$\geq 10^3$ $< 10^4$	$\geq 10^4$
× None (0,32)	0.00 %	0.00 %	0.00 %	30.43 %
△ Fast (0,32)	0.00 %	0.00 %	0.00 %	65.22 %
○ Can. (0,32)	0.00 %	0.00 %	3.85 %	65.22 %
× None (1,8)	0.00 %	0.00 %	0.00 %	60.87 %
△ Fast (1,8)	0.00 %	0.00 %	7.69 %	82.61 %
○ Can. (1,8)	0.00 %	0.00 %	11.54 %	82.61 %
× None (1,32)	0.00 %	0.00 %	3.85 %	82.61 %
△ Fast (1,32)	0.00 %	0.00 %	11.54 %	91.30 %
○ Can. (1,32)	0.00 %	0.00 %	23.08 %	95.65 %
× None (2,32)	0.00 %	0.00 %	15.38 %	95.65 %
△ Fast (2,32)	0.00 %	0.00 %	26.92 %	95.65 %
○ Can. (2,32)	0.00 %	0.00 %	42.31 %	95.65 %
× None (∞ ,32)	0.00 %	0.00 %	23.08 %	95.65 %
△ Fast (∞ ,32)	0.00 %	3.57 %	30.77 %	95.65 %
○ Can. (∞ ,32)	0.00 %	3.57 %	46.15 %	95.65 %
× None (∞ , ∞)	0.00 %	10.71 %	50.00 %	95.65 %
△ Fast (∞ , ∞)	0.00 %	10.71 %	50.00 %	95.65 %
○ Can. (∞ , ∞)	0.00 %	10.71 %	50.00 %	95.65 %

Table XV: Timeout rates

	$\geq 10^1$ $< 10^2$	$\geq 10^2$ $< 10^3$	$\geq 10^3$ $< 10^4$	$\geq 10^4$
× None (0,32)	0.00 %	0.00 %	0.00 %	4.35 %
△ Fast (0,32)	0.00 %	0.00 %	0.00 %	4.35 %
○ Can. (0,32)	0.00 %	0.00 %	0.00 %	4.35 %
× None (1,8)	0.00 %	0.00 %	0.00 %	4.35 %
△ Fast (1,8)	0.00 %	0.00 %	0.00 %	4.35 %
○ Can. (1,8)	0.00 %	0.00 %	0.00 %	4.35 %
× None (1,32)	0.00 %	0.00 %	0.00 %	4.35 %
△ Fast (1,32)	0.00 %	0.00 %	0.00 %	4.35 %
○ Can. (1,32)	0.00 %	0.00 %	0.00 %	4.35 %
× None (2,32)	0.00 %	0.00 %	0.00 %	4.35 %
△ Fast (2,32)	0.00 %	0.00 %	0.00 %	4.35 %
○ Can. (2,32)	0.00 %	0.00 %	3.85 %	4.35 %
× None (∞ ,32)	0.00 %	0.00 %	0.00 %	4.35 %
△ Fast (∞ ,32)	0.00 %	0.00 %	0.00 %	4.35 %
○ Can. (∞ ,32)	0.00 %	3.57 %	3.85 %	4.35 %
× None (∞ , ∞)	0.00 %	3.57 %	3.85 %	4.35 %
△ Fast (∞ , ∞)	0.00 %	3.57 %	3.85 %	4.35 %
○ Can. (∞ , ∞)	0.00 %	10.71 %	11.54 %	4.35 %

Table XVI: Out-of-memory rates

	$\geq 10^1$ $< 10^2$	$\geq 10^2$ $< 10^3$	$\geq 10^3$ $< 10^4$	$\geq 10^4$
× None (0,32)	0.00 %	0.00 %	0.00 %	0.00 %
△ Fast (0,32)	0.00 %	0.00 %	0.00 %	0.00 %
○ Can. (0,32)	0.00 %	0.00 %	0.00 %	0.00 %
× None (1,8)	0.00 %	0.00 %	0.00 %	0.00 %
△ Fast (1,8)	0.00 %	0.00 %	0.00 %	0.00 %
○ Can. (1,8)	0.00 %	0.00 %	0.00 %	0.00 %
× None (1,32)	0.00 %	0.00 %	0.00 %	0.00 %
△ Fast (1,32)	0.00 %	0.00 %	0.00 %	0.00 %
○ Can. (1,32)	0.00 %	0.00 %	0.00 %	0.00 %
× None (2,32)	0.00 %	0.00 %	0.00 %	0.00 %
△ Fast (2,32)	0.00 %	0.00 %	0.00 %	0.00 %
○ Can. (2,32)	0.00 %	0.00 %	0.00 %	0.00 %
× None (∞ ,32)	0.00 %	0.00 %	0.00 %	0.00 %
△ Fast (∞ ,32)	0.00 %	0.00 %	0.00 %	0.00 %
○ Can. (∞ ,32)	0.00 %	0.00 %	0.00 %	0.00 %
× None (∞ , ∞)	0.00 %	28.57 %	34.62 %	0.00 %
△ Fast (∞ , ∞)	0.00 %	32.14 %	34.62 %	0.00 %
○ Can. (∞ , ∞)	0.00 %	32.14 %	34.62 %	0.00 %

Table XVII: Powering-too-large rates

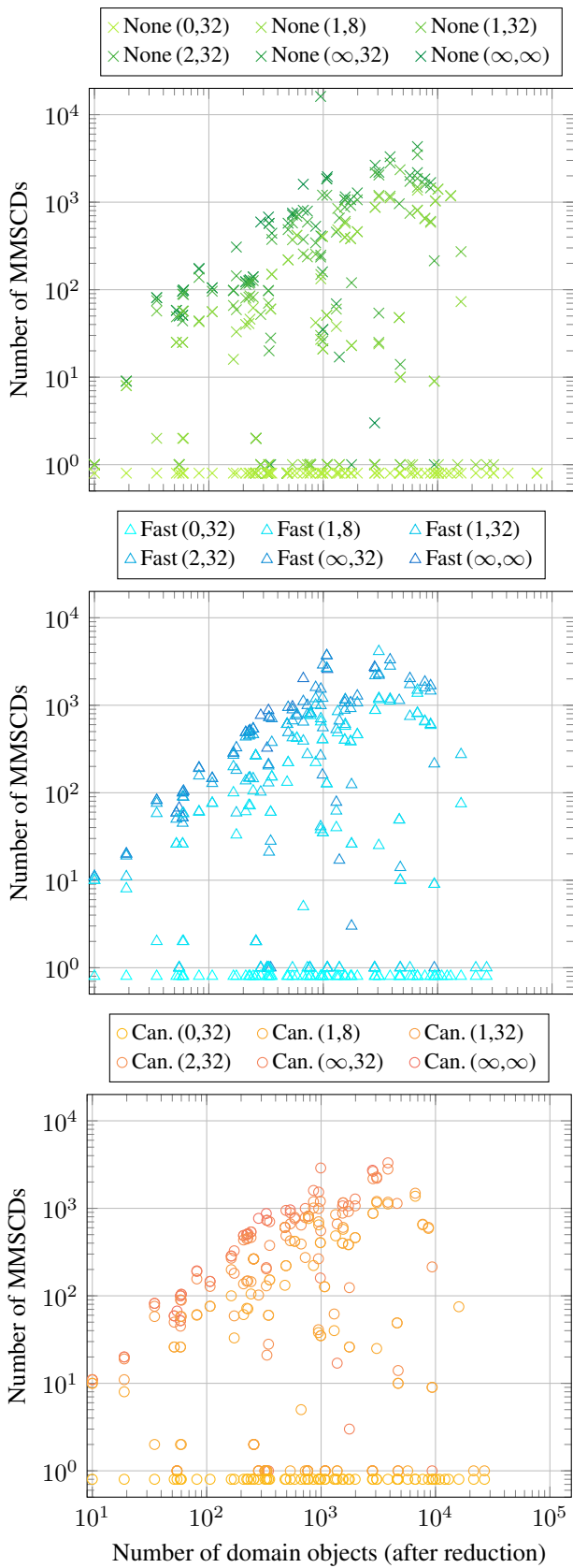


Figure XVIII: Computing all MMSCDs of \mathcal{I}

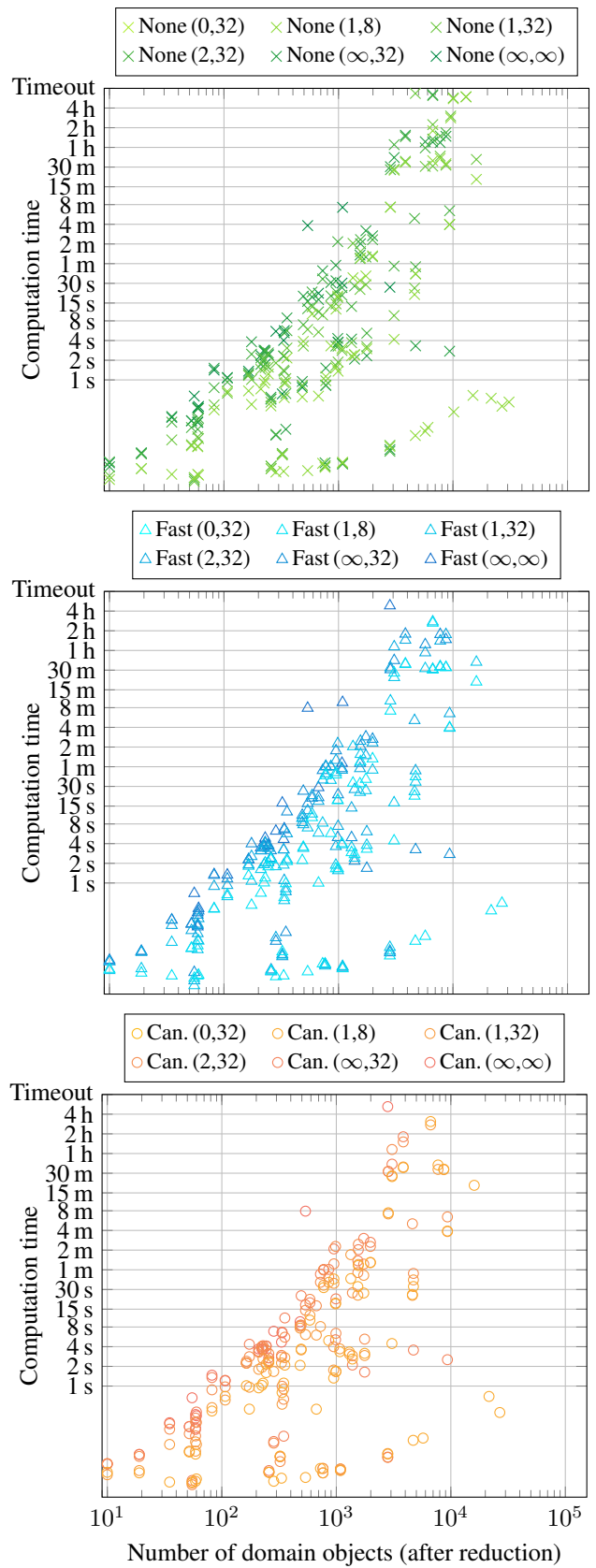


Figure XIX: Computing all MMSCDs of \mathcal{I}

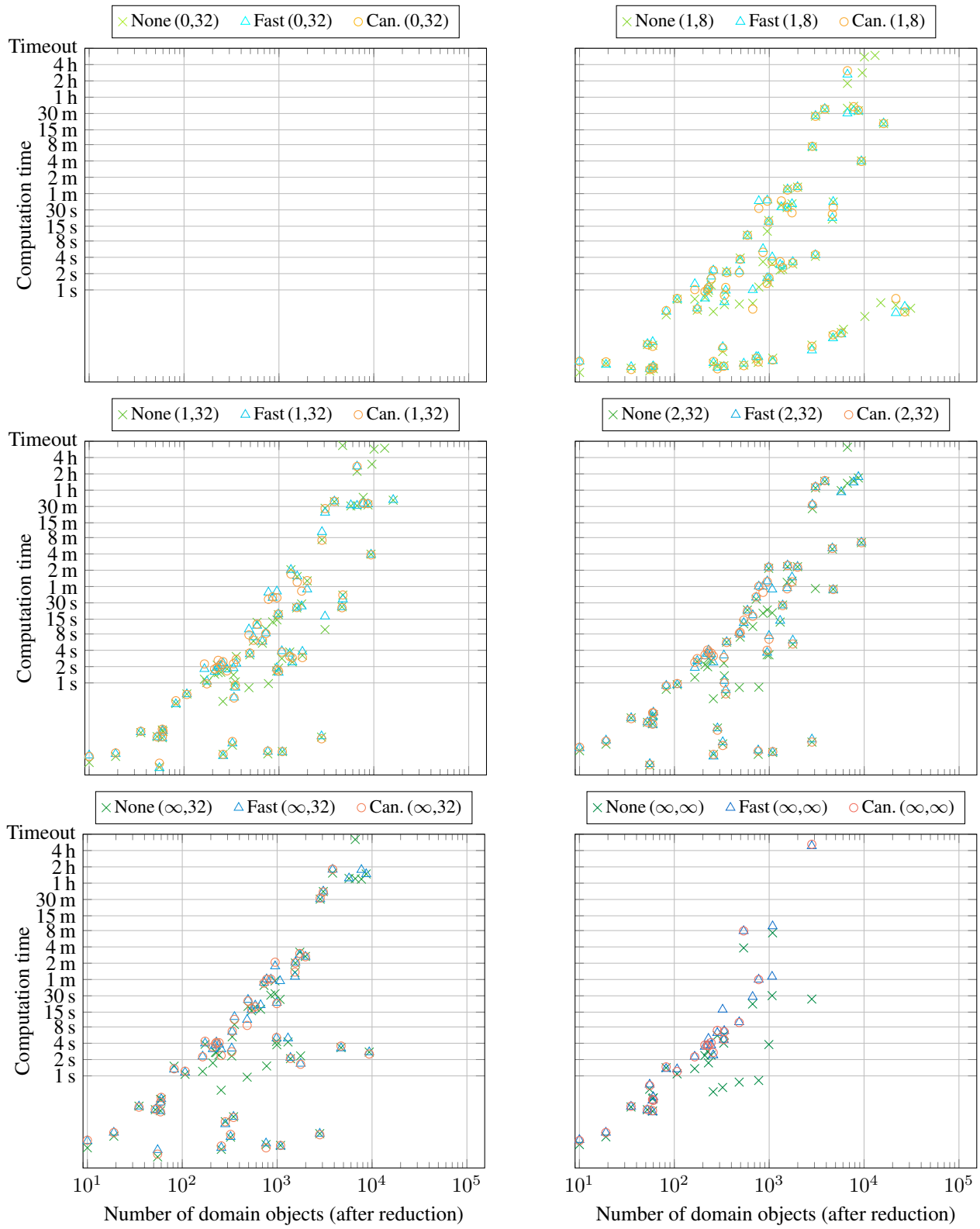


Figure XX: Computing all MMSCDs of \mathcal{I}

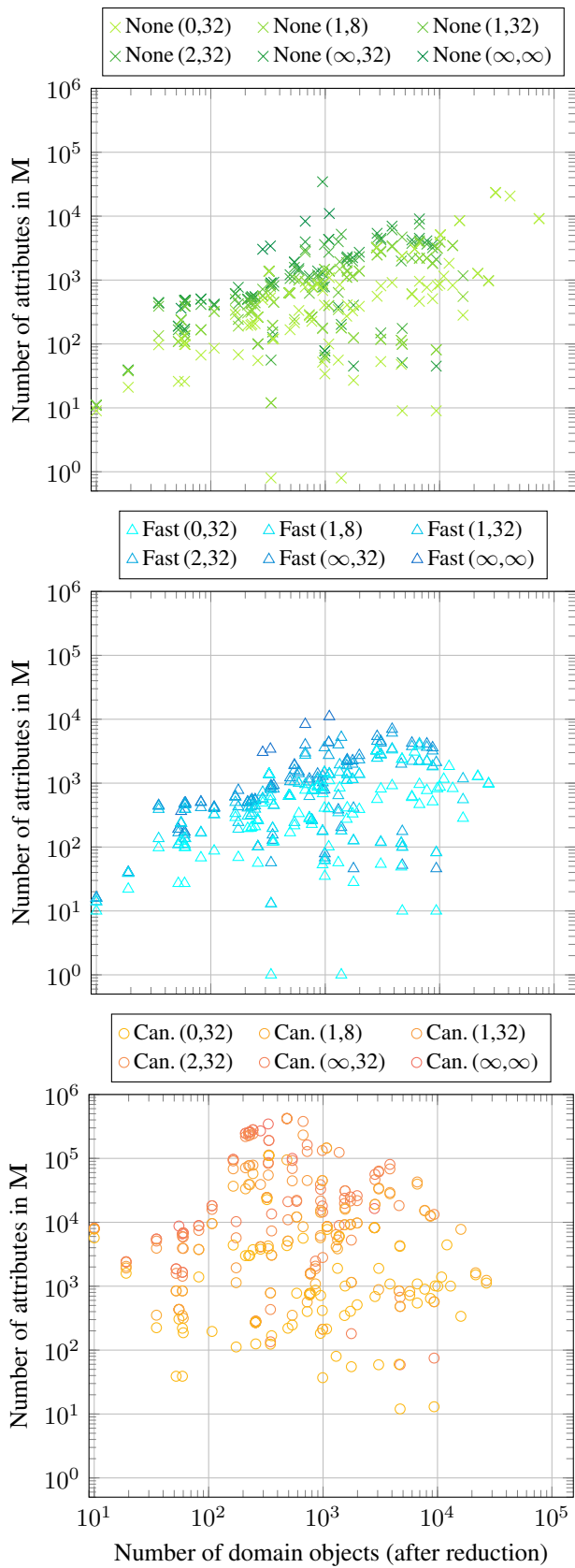


Figure XXI: Computing the induced context $\mathbb{K}_{\mathcal{T}}$

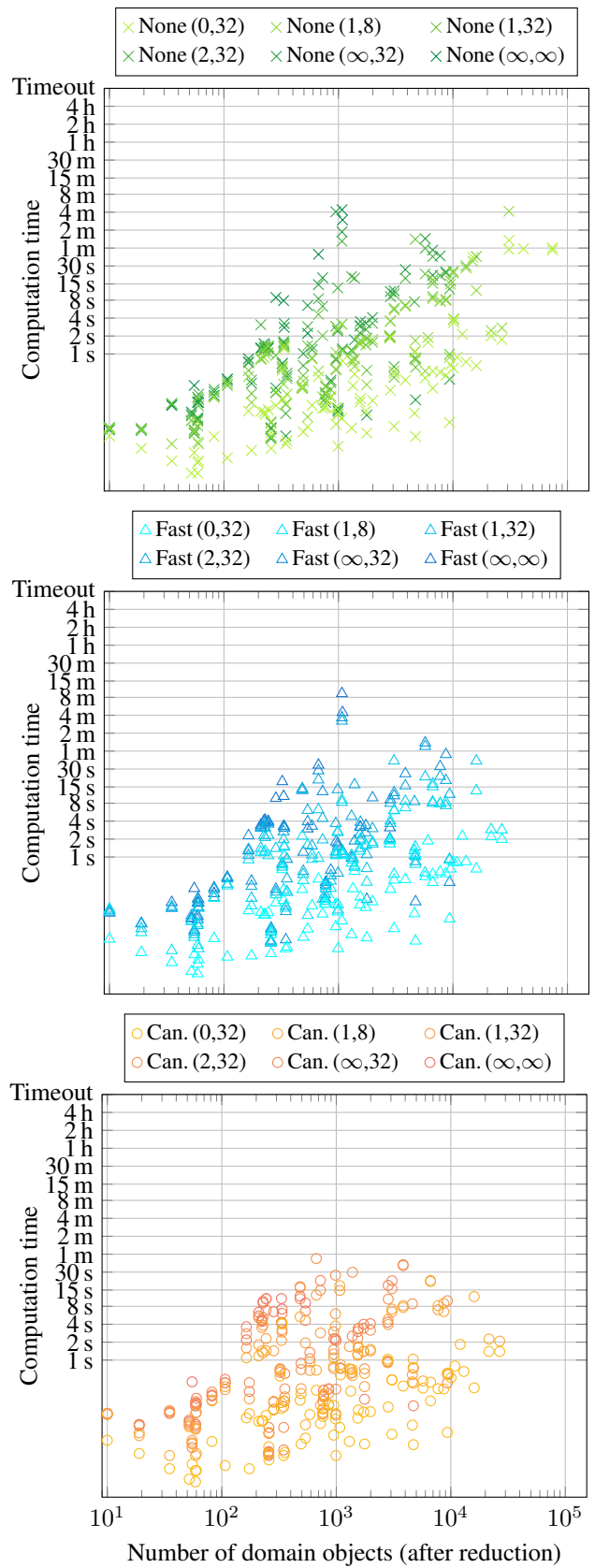


Figure XXII: Computing the induced context $\mathbb{K}_{\mathcal{T}}$

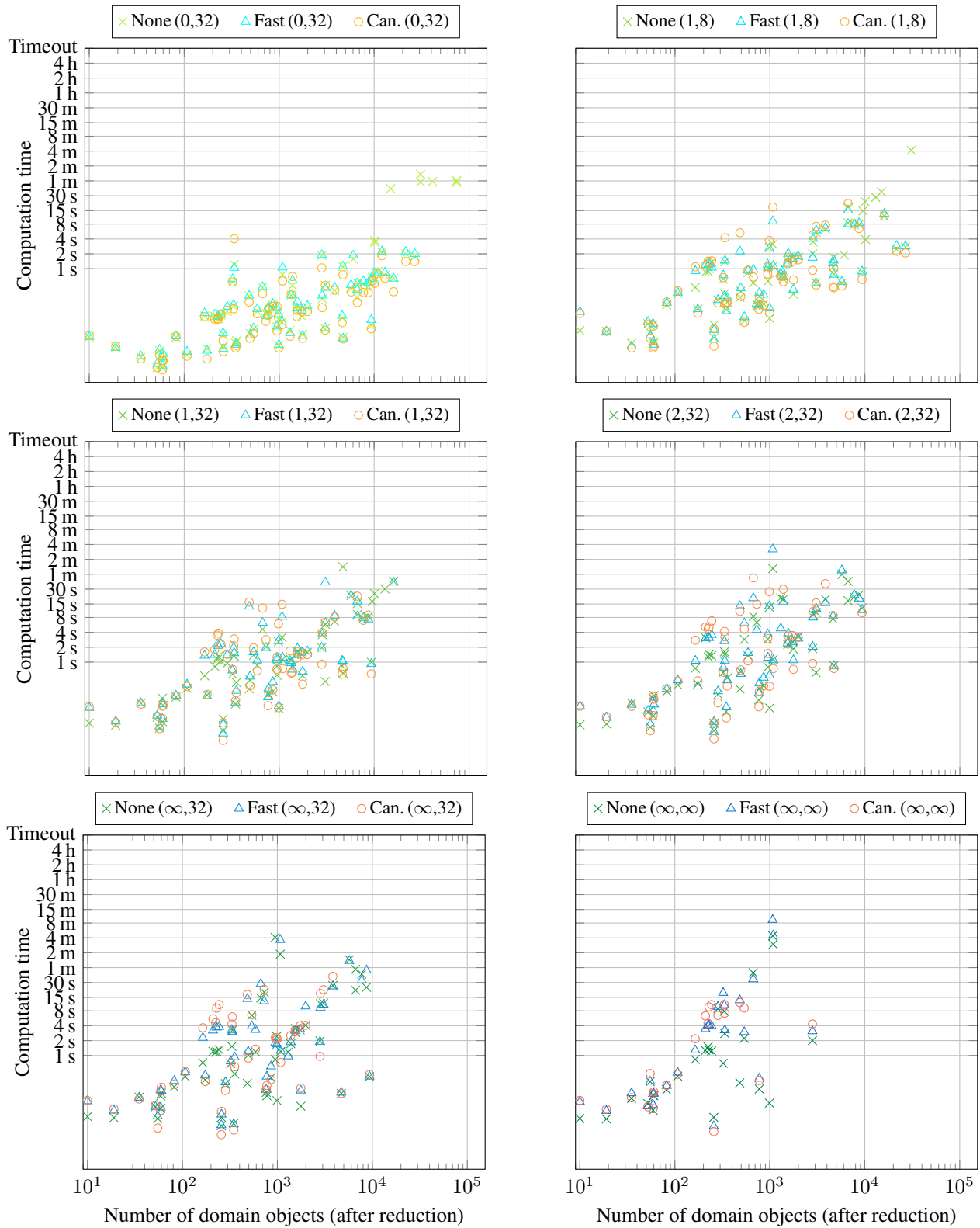


Figure XXIII: Computing the induced context $\mathbb{K}_{\mathcal{I}}$

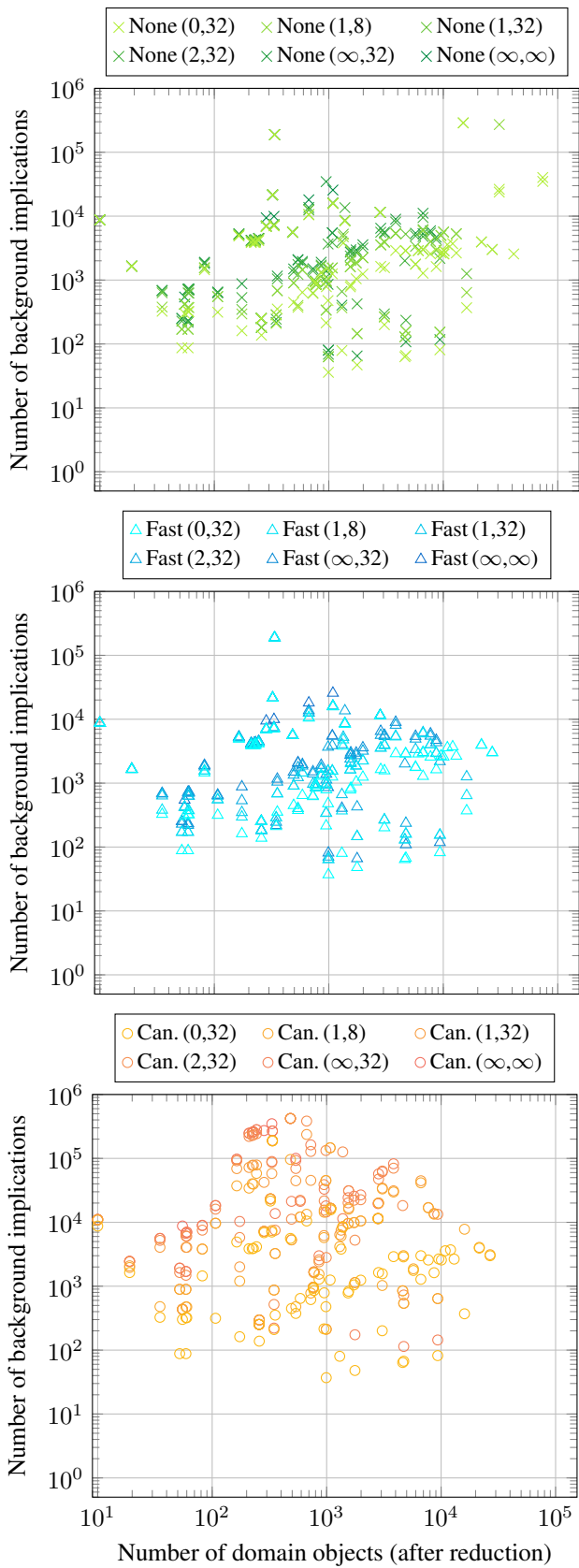


Figure XXIV: Computing all background impl. in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

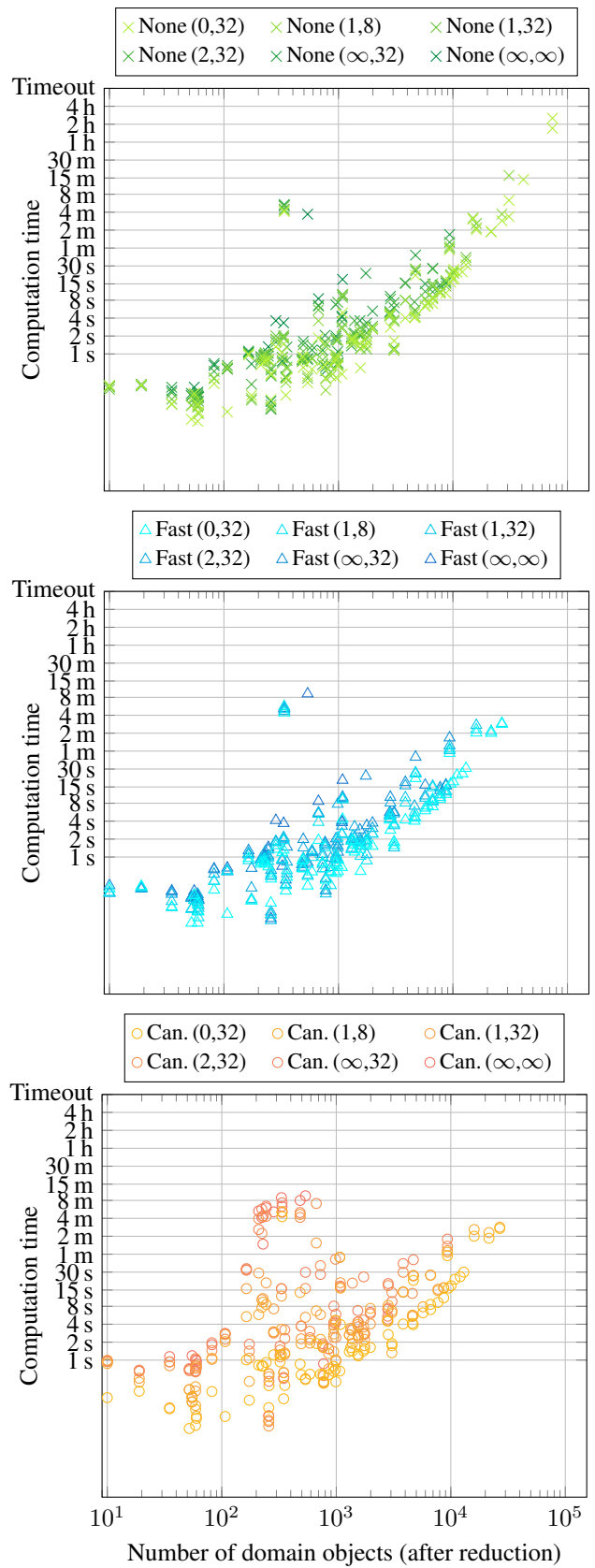


Figure XXV: Computing all background impl. in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

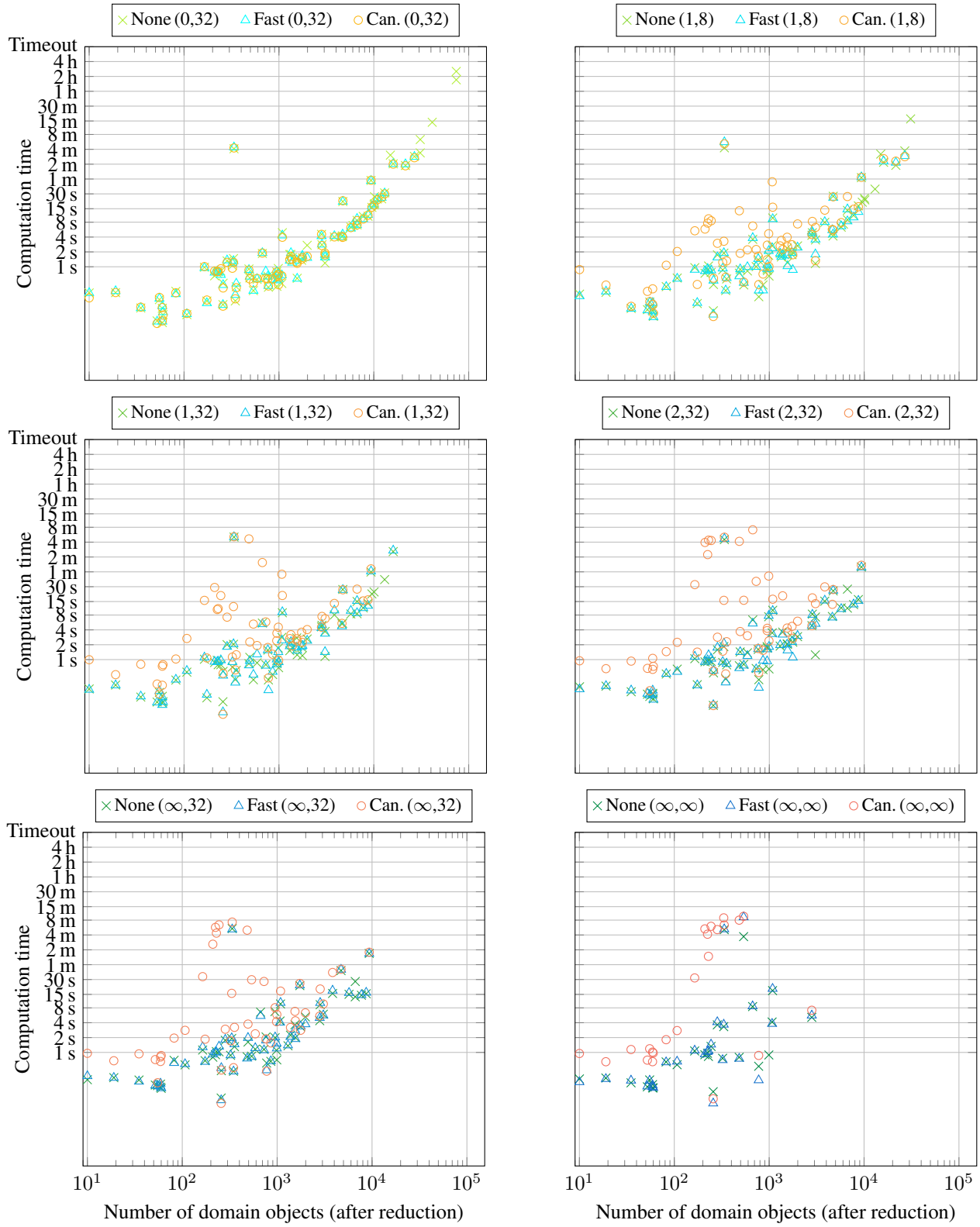


Figure XXVI: Computing all background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

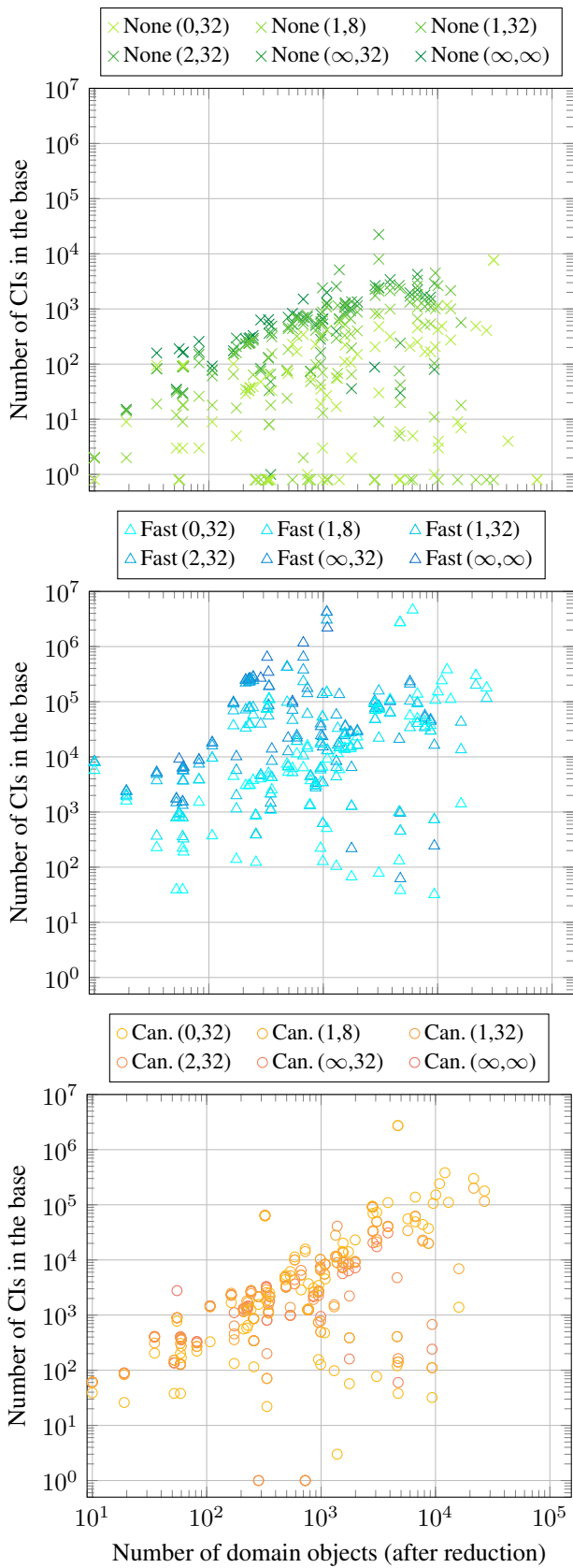


Figure XXVII: Computing the CI base $\text{Can}(\mathcal{I}, \mathcal{T})$

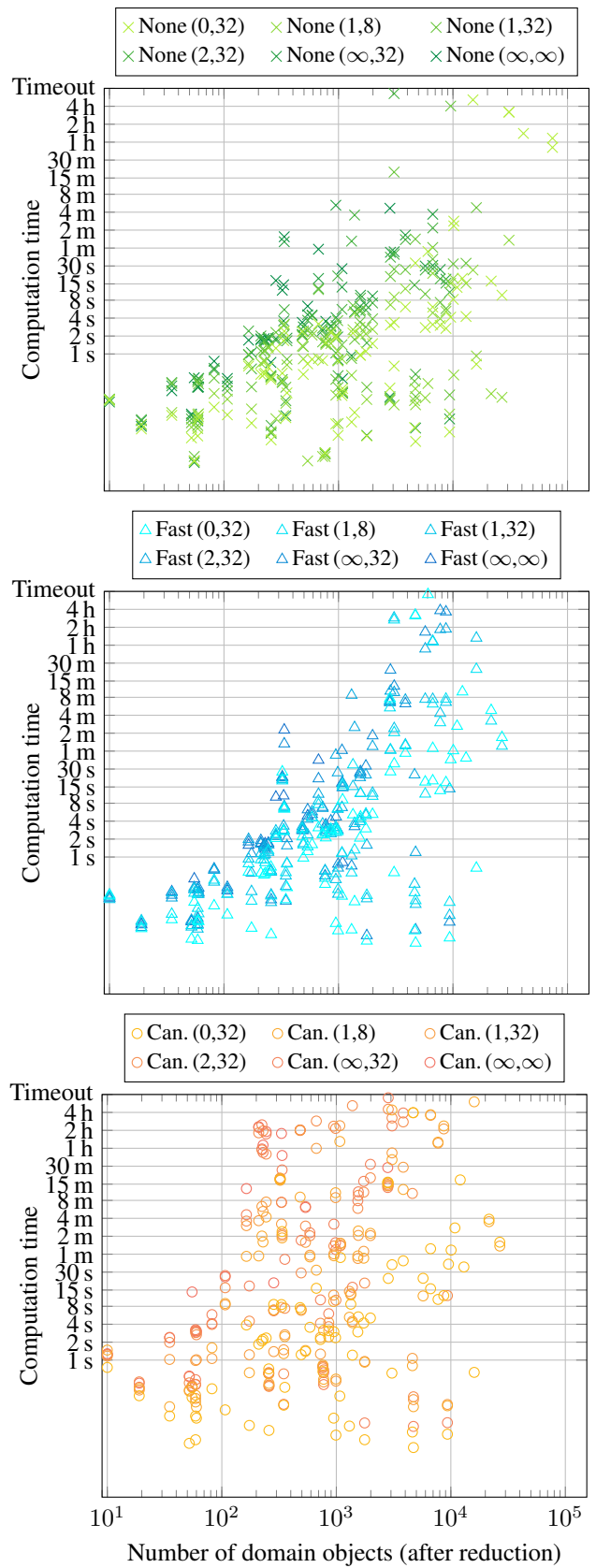


Figure XXVIII: Computing the impl. base $\text{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}}^*)$

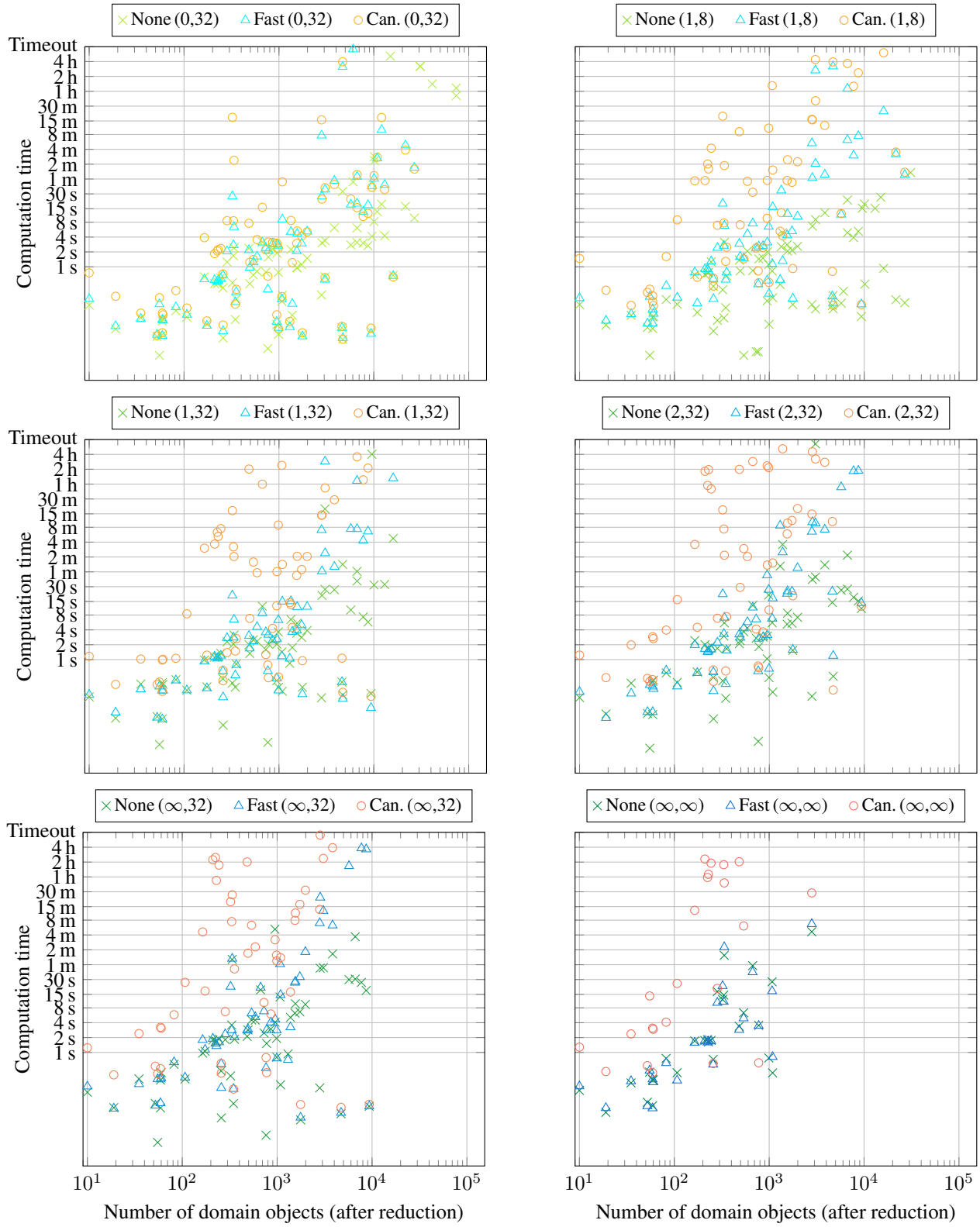


Figure XXIX: Computing the implication base $\text{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I}, \mathcal{T}}^*)$

A Alternative Background Implications

Recall that the background implications in $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ (see Page 10) are constructed from the input TBox \mathcal{T} and ensure that the canonical CI base in Theorem 10 does not contain unnecessary CIs, i.e., no tautologies and no CIs that already follow from \mathcal{T} . As mentioned in Section 6, in an earlier version of Section 3.2 we used a rather straightforward definition of the background implication set $\mathcal{L}_{\mathcal{I},\mathcal{T}}$: it consisted of all implications $\mathbf{C} \rightarrow \mathbf{D}$ where \mathbf{C} and \mathbf{D} are subsets of \mathbf{M} such that \mathcal{T} entails $\sqcap \mathbf{C} \sqsubseteq \sqcap \mathbf{D}$. In all proofs in Section 3.2 this alternative definition of $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ can be used without any issues. However, it is of exponential size w.r.t. the attribute set \mathbf{M} , which is itself exponential in $\text{Dom}(\mathcal{I})$ in the worst case, and it should thus not be directly used in the computation of the canonical CI base. We resolved this bottleneck by developing a conservative extension $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$ of the alternative $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ that can be computed in polynomial time w.r.t. \mathbf{M} . It is defined over a superset of \mathbf{M} but entails the same implications involving only attributes in \mathbf{M} . Since this conservative extension $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$ is computed and used in the prototype, we will here describe all details and provide proofs. However, we do not expect a relevant difference in performance as both implication sets (the newer $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ on Page 10 and the conservative extension $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$) have similar size and can be computed in similar ways.

In what follows, $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ always refers to the alternative, older definition of the background implications (see above).

A.1 The Conservative Extension $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$

According to Proposition 3, we can decide by means of the \sqsubseteq_+ -Rule and the \sqsubseteq_{\perp} -Rule whether the background implication set $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ entails an implication $\mathbf{C} \rightarrow \mathbf{D}$. We will show how these rules can be emulated with FCA implications that use additional attributes.

Since FCA implications can only capture conjunctions in their premises and conclusions, we take as additional attributes all top-level conjuncts in the CIs in \mathcal{T} . The so extended attribute set contains all atoms that are necessary for detecting applicability of the \sqsubseteq_+ -Rule or of the \sqsubseteq_{\perp} -Rule as well as all atoms that could be added by an application of the \sqsubseteq_+ -Rule. Formally, we extend the attribute set \mathbf{M} with the smallest set \mathbf{M}^* of additional attributes such that

- $\text{Conj}(E) \cup \text{Conj}(F) \subseteq \mathbf{M} \cup \mathbf{M}^*$ for each $E \sqsubseteq F \in \mathcal{T}_+$
- and $\text{Conj}(E) \subseteq \mathbf{M} \cup \mathbf{M}^*$ for each $E \sqsubseteq \perp \in \mathcal{T}_{\perp}$.

Over the extended attribute set $\mathbf{M} \cup \mathbf{M}^*$ we define the implication set $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$ that consists of the following implications:

- (B11) $\text{Conj}(E) \rightarrow \text{Conj}(F)$ for each $E \sqsubseteq F \in \mathcal{T}_+$
- (B12) $\{\exists r.E\} \rightarrow \{\exists r.F\}$ for each $\exists r.E \in \mathbf{M} \cup \mathbf{M}^*$ and each $\exists r.F \in \mathbf{M} \cup \mathbf{M}^*$ with $E \sqsubseteq_{\mathcal{T}_+} F$
- (B13) $\text{Conj}(E) \rightarrow \{\perp\}$ for each $E \sqsubseteq \perp \in \mathcal{T}_{\perp}$
- (B14) $\text{Conj}(F) \rightarrow \{\perp\}$ for each $E \sqsubseteq F \in \mathcal{T}_+$ with $F \sqsubseteq_{\mathcal{T}} \perp$
- (B15) $\{\perp\} \rightarrow \mathbf{M} \cup \mathbf{M}^*$

We are going to prove that $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$ is a conservative extension in the following sense.

Definition XXXVII. Consider implication sets \mathcal{L} and \mathcal{L}^* over M and M^* , respectively, where $M \subseteq M^*$. We say that

\mathcal{L}^* is a *conservative extension* of \mathcal{L} if both entail the same implications over M , i.e., $\mathcal{L} \models U \rightarrow V$ iff $\mathcal{L}^* \models U \rightarrow V$ for each implication $U \rightarrow V$ where $U, V \subseteq M$.

Lemma XXXVIII. \mathcal{L}^* is a conservative extension of \mathcal{L} iff $U^{\mathcal{L}} = U^{\mathcal{L}^*} \cap M$ for each subset $U \subseteq M$.

Proof. Recall that $\mathcal{L} \models U \rightarrow V$ iff $V \subseteq U^{\mathcal{L}}$.

We begin with the if direction. Therefore consider an implication $U \rightarrow V$ where $U, V \subseteq M$. From $U^{\mathcal{L}} = U^{\mathcal{L}^*} \cap M$ it follows that $V \subseteq U^{\mathcal{L}}$ iff $V \subseteq U^{\mathcal{L}^*}$, and thus $\mathcal{L} \models U \rightarrow V$ iff $\mathcal{L}^* \models U \rightarrow V$.

Next, we show the only-if direction, for which we consider a subset $U \subseteq M$. On the one hand, the implication $U \rightarrow U^{\mathcal{L}}$ follows from \mathcal{L} and thus also from \mathcal{L}^* , i.e., $U^{\mathcal{L}} \subseteq U^{\mathcal{L}^*}$. On the other hand, the implication $U \rightarrow U^{\mathcal{L}^*} \cap M$ follows from \mathcal{L}^* and thus also from \mathcal{L} , i.e., $U^{\mathcal{L}^*} \cap M \subseteq U^{\mathcal{L}}$. We conclude that $U^{\mathcal{L}} = U^{\mathcal{L}^*} \cap M$. \square

Lemma XXXIX. Let \mathcal{L} be an implication set over $\mathbf{M} \cup \mathbf{M}^*$ such that $\sqcap \mathbf{E} \sqsubseteq_{\mathcal{T}} \sqcap \mathbf{F}$ for each implication $\mathbf{E} \rightarrow \mathbf{F}$ in \mathcal{L} . For each subset $\mathbf{C} \subseteq \mathbf{M} \cup \mathbf{M}^*$, it holds that $\sqcap \mathbf{C} \sqsubseteq_{\mathcal{T}} \sqcap \mathbf{C}^{\mathcal{L}}$.

Proof. As explained above, there must be sets $\mathbf{C}_0, \dots, \mathbf{C}_n$ and implications $\mathbf{E}_1 \rightarrow \mathbf{F}_1, \dots, \mathbf{E}_n \rightarrow \mathbf{F}_n$ in \mathcal{L} such that

1. $\mathbf{C}_0 = \mathbf{C}$
2. $\mathbf{E}_{k+1} \subseteq \mathbf{C}_k$ but $\mathbf{F}_{k+1} \not\subseteq \mathbf{C}_k$
3. $\mathbf{C}_{k+1} = \mathbf{C}_k \cup \mathbf{F}_{k+1}$
4. $\mathbf{C}_n = \mathbf{C}^{\mathcal{L}}$.

Consider a number k with $0 \leq k < n$. From $\mathbf{E}_{k+1} \subseteq \mathbf{C}_k$ we infer $\sqcap \mathbf{C}_k \sqsubseteq^{\emptyset} \sqcap \mathbf{E}_{k+1}$, and the assumption implies $\sqcap \mathbf{E}_{k+1} \sqsubseteq_{\mathcal{T}} \sqcap \mathbf{F}_{k+1}$. It follows that $\sqcap \mathbf{C}_k \sqsubseteq_{\mathcal{T}} \sqcap (\mathbf{C}_k \cup \mathbf{F}_{k+1}) = \sqcap \mathbf{C}_{k+1}$. By induction, we obtain that $\sqcap \mathbf{C} = \sqcap \mathbf{C}_0 \sqsubseteq_{\mathcal{T}} \sqcap \mathbf{C}_n = \sqcap \mathbf{C}^{\mathcal{L}}$. \square

Proposition XL. $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$ is a conservative extension of $\mathcal{L}_{\mathcal{I},\mathcal{T}}$.

Proof. According to Lemma XXXVIII it is enough to verify that $\mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}} = \mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}^*} \cap \mathbf{M}$ for all $\mathbf{C} \subseteq \mathbf{M}$. The inclusion \supseteq comes first. Let $D \in \mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}} \cap \mathbf{M}$. Since Lemma XXXIX yields $\sqcap \mathbf{C} \sqsubseteq_{\mathcal{T}} \sqcap \mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}^*}$, we infer that $\sqcap \mathbf{C} \sqsubseteq_{\mathcal{T}} D$. Thus $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ contains $\mathbf{C} \rightarrow \{D\}$, and so the closure $\mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}}$ contains D .

Next, we turn our attention to the inclusion \subseteq . Assume that D is an attribute in the closure $\mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}}$. If $\perp \in \mathbf{C}$, then $\mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}} = \mathbf{M} \cup \mathbf{M}^*$ for the implication $\{\perp\} \rightarrow \mathbf{M} \cup \mathbf{M}^*$, and so $D \in \mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}}$. We proceed with the case where $\perp \notin \mathbf{C}$. Lemma XXXIX yields $\sqcap \mathbf{C} \sqsubseteq_{\mathcal{T}} \sqcap \mathbf{C}^{\mathcal{L}_{\mathcal{I},\mathcal{T}}}$, and thus $\sqcap \mathbf{C} \sqsubseteq_{\mathcal{T}} D$. With Proposition 3 we obtain $(\sqcap \mathbf{C})^{\mathcal{T}} \sqsubseteq^{\emptyset} D$, where $(\sqcap \mathbf{C})^{\mathcal{T}}$ is constructed from $\sqcap \mathbf{C}$ by means of the \sqsubseteq_+ -Rule and the \sqsubseteq_{\perp} -Rule.

According to Lemma I, $\sqcap \mathbf{C}$ is equivalent to a CD of the form $\exists^{\text{sim}}(\mathcal{C}, c)$. Specifically, by means of the generic construction in the proof of Lemma I we obtain (\mathcal{C}, c) by augmenting the powering $\wp(\mathcal{I})$ with a fresh object c , i.e., which is no object in the powering $\wp(\mathcal{I})$ and also does not occur in any of the CIs in the TBox \mathcal{T} . Formally, we define the

domain as $\text{Dom}(\mathcal{C}) := \{c\} \cup \text{Dom}(\wp(\mathcal{I}))$ and the interpretation function as $\cdot^{\mathcal{C}} := \{c : A \mid A \in \mathbf{C}\} \cup \{(c, X) : r \mid \exists r. X^{\mathcal{I}} \in \mathbf{C}\} \cup \cdot^{\wp(\mathcal{I})}$.

In the following we consider a sequence of applications of the \sqsubseteq_+ -Rule and the \sqsubseteq_{\perp} -Rule that produces $(\prod \mathbf{C})^{\mathcal{T}}$ from the above representation $\exists^{\text{sim}}(\mathcal{C}, c)$ of $\prod \mathbf{C}$. We will construct from it, step by step, the closure of \mathbf{C} w.r.t. the implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ and we show that it contains all attributes in \mathbf{M} that subsume $(\prod \mathbf{C})^{\mathcal{T}}$, in particular D .

Since \mathcal{I} is a model of \mathcal{T} and $X \in C^{\wp(\mathcal{I})}$ iff $X \subseteq C^{\mathcal{I}}$ by Lemma IX, also the powering $\wp(\mathcal{I})$ is a model of \mathcal{T} . This means that, within \mathcal{C} , neither the \sqsubseteq_+ -Rule nor the \sqsubseteq_{\perp} -Rule is applicable for objects in $\text{Dom}(\wp(\mathcal{I}))$; in the beginning they are only applicable to the root object c .

Now assume that $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_n$ is a sequence of interpretations such that

- i. $\mathcal{C}_0 := \mathcal{C}$
- ii. The \sqsubseteq_{\perp} -Rule does not fail for \mathcal{C}_k , and the subsequent interpretation \mathcal{C}_{k+1} is constructed from \mathcal{C}_k by
 - (a) once applying the \sqsubseteq_+ -Rule at the root object c , say for the CI $E_{k+1} \sqsubseteq F_{k+1} \in \mathcal{T}_+$ where $E_{k+1} = \exists^{\text{sim}}(\mathcal{E}_{k+1}, e_{k+1})$ and $F_{k+1} = \exists^{\text{sim}}(\mathcal{F}_{k+1}, f_{k+1})$,
 - (b) and then exhaustively applying the \sqsubseteq_+ -Rule at all objects except c .
- iii. Either the \sqsubseteq_{\perp} -Rule fails for \mathcal{C}_n (then $\prod \mathbf{C}$ is not satisfiable w.r.t. \mathcal{T}), or the \sqsubseteq_+ -Rule is not applicable to \mathcal{C}_n (then this last interpretation is the saturation $\text{sat}(\mathcal{C}, \mathcal{T}_+)$).

We set $\mathcal{C}_k := \exists^{\text{sim}}(\mathcal{C}_k, c)$ for each $k < n$. If the \sqsubseteq_{\perp} -Rule failed, then let $\mathcal{C}_n := \perp$, and otherwise $\mathcal{C}_n := \exists^{\text{sim}}(\mathcal{C}_n, c)$, i.e., $\mathcal{C}_n = (\prod \mathbf{C})^{\mathcal{T}}$.

From this sequence, we will construct a sequence of subsets $\mathbf{C}_0, \mathbf{C}_1, \dots, \mathbf{C}_n$ of $\mathbf{M} \cup \mathbf{M}^*$ in the following way:

- I. \mathbf{C}_0 is obtained as the closure of \mathbf{C} w.r.t. the implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^* \setminus \{\text{Conj}(E) \rightarrow \text{Conj}(F) \mid E \sqsubseteq F \in \mathcal{T}_+\}$.
- II. \mathbf{C}_{k+1} is constructed as the closure of \mathbf{C}_k
 - (a) first w.r.t. the implication $\text{Conj}(E_{k+1}) \rightarrow \text{Conj}(F_{k+1})$, i.e., we have $\text{Conj}(E_{k+1}) \subseteq \mathbf{C}_k$ and we set $\mathbf{C}_{k+1} := \mathbf{C}_k \cup \text{Conj}(F_{k+1})$,
 - (b) and then w.r.t. the implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^* \setminus \{\text{Conj}(E) \rightarrow \text{Conj}(F) \mid E \sqsubseteq F \in \mathcal{T}_+\}$.

We need to show that the latter sequence is well-defined, specifically that Instruction II(a) can always be carried out. To this end, we show the following claim.

Claim. For each $M \in \mathbf{M} \cup \mathbf{M}^*$, if $\mathcal{C}_k \sqsubseteq^{\emptyset} M$, then $M \in \mathbf{C}_k$.

Recall from Property ii(a) of the sequence \mathcal{C}_k that the \sqsubseteq_+ -Rule is applicable for a CI with premise E_{k+1} , i.e., $\mathcal{C}_k \sqsubseteq^{\emptyset} E_{k+1}$. The claim yields $\text{Conj}(E_{k+1}) \subseteq \mathbf{C}_k$, and thus Instruction II(a) can be executed.

Moreover, the claim allows us to finish the proof as follows. Recall that $\mathcal{C}_n = (\prod \mathbf{C})^{\mathcal{T}}$, and note that each \mathbf{C}_k is a subset of the closure $\mathbf{C}^{\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*}$. We can thus conclude from the claim (specifically for $k = n$) that $(\prod \mathbf{C})^{\mathcal{T}} \sqsubseteq^{\emptyset} M$ implies $M \in \mathbf{C}^{\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*}$ for each $M \in \mathbf{M} \cup \mathbf{M}^*$. Taking as M the considered attribute D yields $D \in \mathbf{C}^{\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*}$ as we need it.

It remains to show the claim. We begin with the induction base ($k = 0$) and therefore assume that $\mathcal{C}_0 \sqsubseteq^{\emptyset} M$. If the \sqsubseteq_{\perp} -Rule failed already for \mathcal{C}_0 , then $\mathcal{C}_0 = \perp$. Furthermore, there must be a disjointness axiom $E \sqsubseteq_{\perp} \perp \in \mathcal{T}_{\perp}$ with $E = \exists^{\text{sim}}(\mathcal{E}, e)$ and $(\mathcal{E}, e) \preceq (\mathcal{C}_0, c)$. Since $\exists^{\text{sim}}(\mathcal{C}_0, c) = \exists^{\text{sim}}(\mathcal{C}, c) \equiv^{\emptyset} \prod \mathbf{C}$, we infer with Lemma 2 that $\prod \mathbf{C} \sqsubseteq^{\emptyset} E$, i.e., for each $E' \in \text{Conj}(E)$ there is some $C' \in \mathbf{C}$ such that $C' \sqsubseteq^{\emptyset} E'$. We conclude that either $C' = E'$ or $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ contains the implication $\{C'\} \rightarrow \{E'\}$ (by Instruction (BI2)), and thus $\text{Conj}(E) \subseteq \mathbf{C}_0$. Furthermore, $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ contains the implications $\text{Conj}(E) \rightarrow \{\perp\}$ (by Instruction (BI3)) and $\{\perp\} \rightarrow \mathbf{M} \cup \mathbf{M}^*$ (by Instruction (BI5)), which implies $\mathbf{C}_0 = \mathbf{M} \cup \mathbf{M}^*$ and thus also $M \in \mathbf{C}_0$.

We proceed with the remaining case of the induction base, where the \sqsubseteq_{\perp} -Rule did not fail for \mathcal{C}_0 . Since then $\mathcal{C}_0 = \exists^{\text{sim}}(\mathcal{C}_0, c) = \exists^{\text{sim}}(\mathcal{C}, c) \equiv^{\emptyset} \prod \mathbf{C}$, we have $\prod \mathbf{C} \sqsubseteq^{\emptyset} M$. We must prove that $M \in \mathbf{C}_0$. Since M is an atom, there must be some $C \in \mathbf{C}$ such that $C \sqsubseteq^{\emptyset} M$. If $C = M$, then it immediately follows that $M \in \mathbf{C}_0$. Otherwise, $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ contains the implication $\{C\} \rightarrow \{M\}$ (by Instruction (BI2)) and thus \mathbf{C}_0 contains M .

Last, we turn our attention towards the induction step ($k \rightarrow k+1$). Recall that \mathcal{C}_{k+1} is obtained from \mathcal{C}_k as follows. First the \sqsubseteq_+ -Rule is once applied at the root object c for the CI $E_{k+1} \sqsubseteq F_{k+1}$ where $E_{k+1} = \exists^{\text{sim}}(\mathcal{E}_{k+1}, e_{k+1})$ and $F_{k+1} = \exists^{\text{sim}}(\mathcal{F}_{k+1}, f_{k+1})$, which requires that $(\mathcal{E}_{k+1}, e_{k+1}) \preceq (\mathcal{C}_k, c)$ and yields the interpretation with domain $\text{Dom}(\mathcal{C}_k) \cup \text{Dom}(\mathcal{F}_{k+1})$ and interpretation function

$$\begin{aligned} & \cdot^{\mathcal{C}_k} \cup \cdot^{\mathcal{F}_{k+1}} \cup \{c : A \mid f_{k+1} : A \in \cdot^{\mathcal{F}_{k+1}}\} \\ & \cup \{(c, x) : r \mid (f_{k+1}, x) : r \in \cdot^{\mathcal{F}_{k+1}}\}. \end{aligned}$$

Afterwards the \sqsubseteq_+ -Rule is exhaustively applied at all objects except c and for all CIs in \mathcal{T}_+ . Since this has already been done within \mathcal{C}_k in the previous induction step, this means that we saturate the part \mathcal{F}_{k+1} only. Since c is not reachable from any other object, no further assertions for c are added. It follows that (\mathcal{C}_{k+1}, x) and $(\text{sat}(\mathcal{F}_{k+1}, \mathcal{T}_+), x)$ are simulation-equivalent for each $(f_{k+1}, x) : r \in \cdot^{\mathcal{F}_{k+1}}$. We conclude that $\exists^{\text{sim}}(\mathcal{C}_{k+1}, c)$ is equivalent to

$$\begin{aligned} & \exists^{\text{sim}}(\mathcal{C}_k, c) \\ & \sqcap \sqcap \{A \mid f_{k+1} : A \in \cdot^{\mathcal{F}_{k+1}}\} \\ & \sqcap \sqcap \{\exists r. \exists^{\text{sim}}(\text{sat}(\mathcal{F}_{k+1}, \mathcal{T}_+), x) \mid (f_{k+1}, x) : r \in \cdot^{\mathcal{F}_{k+1}}\} \end{aligned}$$

Now let $\exists^{\text{sim}}(\mathcal{C}_{k+1}, c) \sqsubseteq^{\emptyset} M$ for an atom $M \in \mathbf{M} \cup \mathbf{M}^*$. (Then $M \neq \perp$.) If this subsumption already holds for \mathcal{C}_k , then the induction hypothesis yields $M \in \mathbf{C}_k$, and with $\mathbf{C}_k \subseteq \mathbf{C}_{k+1}$ we conclude that $M \in \mathbf{C}_{k+1}$. Otherwise, since M is an atom, M must subsume any of the other atoms in the above conjunction.

- If M is a CN, then $A \sqsubseteq^{\emptyset} M$ for some $f_{k+1} : A \in \cdot^{\mathcal{F}_{k+1}}$, and thus $M = A \in \text{Conj}(F_{k+1})$. According to Instruction II(a), $\text{Conj}(F_{k+1})$ is a subset of \mathbf{C}_{k+1} , and thus $M \in \mathbf{C}_{k+1}$.
- Otherwise, M is an ER, say $\exists r. M'$. Then there is some $(f_{k+1}, x) : r \in \cdot^{\mathcal{F}_{k+1}}$ such that $\exists r. \exists^{\text{sim}}(\text{sat}(\mathcal{F}_{k+1}, \mathcal{T}_+), x) \sqsubseteq^{\emptyset}$

$\exists r.M'$. It follows that $\exists^{\text{sim}}(\text{sat}(\mathcal{F}_{k+1}, \mathcal{T}_+), x) \sqsubseteq^\emptyset M'$ and Lemma V yields $\exists^{\text{sim}}(\mathcal{F}_{k+1}, x) \sqsubseteq^{\mathcal{T}_+} M'$. Since both ERs are in $\mathbf{M} \cup \mathbf{M}^*$, we infer that the implication $\{\exists r.\exists^{\text{sim}}(\mathcal{F}_{k+1}, x)\} \rightarrow \{\exists r.M'\}$ is in $\mathcal{L}_{\mathcal{T}, \mathcal{T}}^*$ (by Instruction (BI2)).

According to Instruction II, \mathbf{C}_{k+1} is obtained from \mathbf{C}_k by first adding all elements in $\text{Conj}(F_{k+1})$, and then saturating w.r.t. the other implications in $\mathcal{L}_{\mathcal{T}, \mathcal{T}}^*$ that do not correspond to a CI in \mathcal{T} . Since $\exists r.\exists^{\text{sim}}(\mathcal{F}_{k+1}, x) \in \text{Conj}(F_{k+1})$, we conclude that $\exists r.M'$ must be added by the latter, i.e., $M \in \mathbf{C}_{k+1}$.

First assume that the \sqsubseteq_{\perp} -Rule does not fail for \mathbf{C}_{k+1} , i.e., $\mathbf{C}_{k+1} = \exists^{\text{sim}}(\mathbf{C}_{k+1}, c)$. From the above we immediately conclude that $\mathbf{C}_{k+1} \sqsubseteq^\emptyset M$ implies $M \in \mathbf{C}_{k+1}$.

We finish the induction step by considering the case where the \sqsubseteq_{\perp} -Rule fails for \mathbf{C}_{k+1} . Then \mathbf{C}_{k+1} is the last interpretation in the sequence, i.e., $n = k + 1$. It follows that $\mathbf{C}_{k+1} = \mathbf{C}_n = \perp$ and further there is a disjointness axiom $E \sqsubseteq_{\perp}$ in \mathcal{T}_{\perp} with $E = \exists^{\text{sim}}(\mathcal{E}, e)$ and $(\mathcal{E}, e) \preceq (\mathbf{C}_{k+1}, y)$ for some object $y \in \text{Dom}(\mathbf{C}_{k+1})$.

- If y is the root object c , then $\exists^{\text{sim}}(\mathbf{C}_{k+1}, c) \sqsubseteq^\emptyset E$ by Lemma 2. The above yields $\text{Conj}(E) \subseteq \mathbf{C}_{k+1}$. Since $\mathcal{L}_{\mathcal{T}, \mathcal{T}}^*$ contains the implications $\text{Conj}(E) \rightarrow \{\perp\}$ (by Instruction (BI3)) and $\{\perp\} \rightarrow \mathbf{M} \cup \mathbf{M}^*$ (by Instruction (BI5)), it follows from Instruction II(b) that $\mathbf{M} \cup \mathbf{M}^* \subseteq \mathbf{C}_{k+1}$, and thus the claim holds.
- Otherwise, since the \sqsubseteq_{\perp} -Rule did not fail for \mathbf{C}_k , the object y must be reachable from an object x with $(f_{k+1}, x): r \in \mathcal{F}_{k+1}$. Recall that (\mathbf{C}_{k+1}, x) and $(\text{sat}(\mathcal{F}_{k+1}, \mathcal{T}_+), x)$ are simulation-equivalent. So the \sqsubseteq_{\perp} -Rule also fails for $\text{sat}(\mathcal{F}_{k+1}, \mathcal{T}_+)$ and thus F_{k+1} is unsatisfiable w.r.t. \mathcal{T} by Lemma VI, i.e., $F_{k+1} \sqsubseteq^{\mathcal{T}} \perp$. It follows that $\mathcal{L}_{\mathcal{T}, \mathcal{T}}^*$ contains the implication $\text{Conj}(F_{k+1}) \rightarrow \{\perp\}$ (by Instruction (BI4)). Since $\text{Conj}(F_{k+1}) \subseteq \mathbf{C}_{k+1}$ and $\mathcal{L}_{\mathcal{T}, \mathcal{T}}^*$ also contains the implication $\{\perp\} \rightarrow \mathbf{M} \cup \mathbf{M}^*$ (by Instruction (BI5)), we conclude from Instruction II(b) that $\mathbf{M} \cup \mathbf{M}^* \subseteq \mathbf{C}_{k+1}$, and thus the claim holds. \square

Lemma XLI. $\mathcal{L}_{\mathcal{T}, \mathcal{T}}^*$ is computable in polynomial time w.r.t. \mathbf{M} and \mathcal{T} .

Proof. The implications in Instructions (BI1) and (BI3) are just syntactic rewritings of the CIs in \mathcal{T} and can thus be computed in polynomial time. Instruction (BI2) yields at most polynomially many implications and since subsumption can be decided in polynomial time by Proposition 3, all these implications can be computed in polynomial time. In Instruction (BI4) one goes through the polynomially many CIs in the sub-TBox \mathcal{T}_+ and checks whether the conclusion is unsatisfiable, which needs polynomial time by Proposition 3. Instruction (BI5) yields only one implication without checking any condition. \square

The Case with a Role-Depth Bound. Regarding the proof of Proposition XL in the role-depth-bounded case, the \sqsubseteq_{+} -Rule and the \sqsubseteq_{\perp} -Rule are initially not only applicable at the root of $\sqcap \mathbf{C}$, but also elsewhere. The crucial point is: $\sqcap \mathbf{C}$ satisfies the chosen role-depth bound, and

thus applying the rules to $\sqcap \mathbf{C}$ above the root does not add anything within the role-depth bound (only at deeper levels). However, since D also satisfies the role-depth bound, these deeper levels are irrelevant. It therefore suffices to apply the rules only at the root, just like in the unbounded case. This assumes that the given TBox \mathcal{T} complies with the role-depth bound as well, since then testing applicability of the \sqsubseteq_{+} -Rule does not need the deeper levels either.

Last, if \mathcal{T} does not satisfy the role-depth bound, then it might also be necessary to apply the \sqsubseteq_{+} -Rule initially to deeper levels in $\sqcap \mathbf{C}$. W.l.o.g. we can assume that this is done right in the beginning, i.e., we exhaustively apply the \sqsubseteq_{+} -Rule everywhere below the root in $\sqcap \mathbf{C}$. This replaces each ER $\exists r.(X^{\mathcal{I}} \upharpoonright_{n-1})$ with $\exists r.((X^{\mathcal{I}} \upharpoonright_{n-1})^{\mathcal{T}_+})$. That the same atoms at the root are then deducible is already taken care of by the implications (BI2).

The only necessary change in the above proof is thus to start with \mathbf{C}_0 obtained from \mathcal{C} by exhaustively applying the \sqsubseteq_{+} -Rule everywhere but at the root. The construction of the \mathbf{C}_i is the same, no change necessary. (Then also the induction base for the claim needs to be adapted in the obvious way.)

A.2 Impl. Bases w.r.t. Conservative Extensions

In order to use the background implications over the extended attribute set we need to modify the definition of the canonical implication base.

Definition XLII. Let $\mathbb{K} := (G, M, I)$ be a formal context and \mathcal{L}^* be an implication set over M^* , where M is a subset of M^* . A *pseudo-intent* of \mathbb{K} w.r.t. \mathcal{L}^* is a subset $P \subseteq M^*$ that fulfills the following conditions:

- (PI*1) $P \cap M$ is no intent, i.e., $(P \cap M) \neq (P \cap M)^{II}$
- (PI*2) P is closed under \mathcal{L}^* , i.e., $P = P^{\mathcal{L}^*}$
- (PI*3) $Q \cap M \subset P$ implies $(Q \cap M)^{II} \subseteq P$ for each pseudo-intent Q .

The canonical implication base $\text{Can}(\mathbb{K}, \mathcal{L}^*)$ consists of all implications $(P \cap M) \rightarrow (P \cap M)^{II}$ where P is a pseudo-intent.

The following lemma shows how the pseudo-intents of \mathbb{K} w.r.t. an implication set \mathcal{L} and w.r.t. a conservative extension \mathcal{L}^* are related.

Lemma XLIII. Consider a formal context $\mathbb{K} := (G, M, I)$, an implication set \mathcal{L} over M , and an implication set \mathcal{L}^* over M^* , where $M \subseteq M^*$. Further let \mathcal{L}^* be a conservative extension of \mathcal{L} . The following statements hold.

1. If P is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* , then $P \cap M$ is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L} .
2. If P is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L} , then $P^{\mathcal{L}^*}$ is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* such that $P = P^{\mathcal{L}^*} \cap M$.

Proof. We show the two statements simultaneously by an induction along the partial order \subseteq_M where $Q \subseteq_M P$ if $Q \cap M \subseteq P \cap M$. Note that \subseteq_M is well-founded.

First of all, since \mathcal{L}^* is a conservative extension of \mathcal{L} , we have $P^{\mathcal{L}} = P^{\mathcal{L}^*} \cap M$ for all subsets $P \subseteq M$.

1. Let P be a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* .

(PI1) (PI*1) already yields that $P \cap M$ is no intent, i.e., (PI1) is satisfied by $P \cap M$.

(PI2) Since \mathcal{L}^* is a conservative extension of \mathcal{L} , we have $(P \cap M)^\mathcal{L} = (P \cap M)^{\mathcal{L}^*} \cap M$. Since $P \cap M \subseteq P$, we infer that $(P \cap M)^{\mathcal{L}^*} \subseteq P^{\mathcal{L}^*}$. By (PI*2), $P = P^{\mathcal{L}^*}$. It follows that $(P \cap M)^{\mathcal{L}^*} \subseteq P$. Since \mathcal{L} is an implication set over M and $P \cap M$ is a subset of M , the closure $(P \cap M)^\mathcal{L}$ must be a subset of M as well. We conclude that $(P \cap M)^\mathcal{L} \subseteq P \cap M$. The converse subset inclusion trivially holds, and thus (PI2) is fulfilled by $P \cap M$.

(PI3) Finally, consider a pseudo-intent Q of \mathbb{K} w.r.t. \mathcal{L} with $Q \subseteq P \cap M$. We need to verify that $Q^{II} \subseteq P \cap M$. We have $Q^{\mathcal{L}^*} \cap M = Q^\mathcal{L} = Q \subseteq P \cap M$, and thus $Q^{\mathcal{L}^*} \subset_M P$. The induction hypothesis yields that $Q^{\mathcal{L}^*}$ is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* and $Q^{\mathcal{L}^*} \cap M = Q$. Then $Q^{\mathcal{L}^*} \cap M \subset P$, and by (PI*3) we infer that $(Q^{\mathcal{L}^*} \cap M)^{II} \subseteq P$, i.e., $Q^{II} \subseteq P$. Since Q^{II} is a subset of M , we conclude that $Q^{II} \subseteq P \cap M$, which verifies (PI3) for $P \cap M$.

2. Assume that P is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L} .

(PI*1) According to (PI1), $P \neq P^{II}$. Since P satisfies (PI2), we have $P = P^\mathcal{L}$ and thus $P^{\mathcal{L}^*} \cap M = P$. It follows that $P^{\mathcal{L}^*} \cap M \neq (P^{\mathcal{L}^*} \cap M)^{II}$, i.e., (PI*1) is satisfied by $P^{\mathcal{L}^*}$.

(PI*2) By definition $P^{\mathcal{L}^*}$ is closed under all implications in \mathcal{L}^* , and so (PI*2) is fulfilled by $P^{\mathcal{L}^*}$.

(PI*3) Last, assume that Q is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* such that $Q \cap M \subset P^{\mathcal{L}^*}$. Recall that $P^{\mathcal{L}^*} \cap M = P^\mathcal{L} = P = P \cap M$, and thus $Q \cap M \subset P \cap M$, i.e., $Q \subset_M P$. By induction hypothesis, $Q \cap M$ is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L} . Since $Q \cap M \subset P$, (PI3) yields $(Q \cap M)^{II} \subseteq P$. Since $P \subseteq P^{\mathcal{L}^*}$, we conclude that $(Q \cap M)^{II} \subseteq P^{\mathcal{L}^*}$, i.e., (PI*3) is satisfied by $P^{\mathcal{L}^*}$. \square

Proposition XLIV. Consider a formal context $\mathbb{K} := (G, M, I)$, an implication set \mathcal{L} over M , and an implication set \mathcal{L}^* over M^* , where $M \subseteq M^*$. Further assume that \mathcal{L}^* is a conservative extension of \mathcal{L} . The canonical implication bases $\text{Can}(\mathbb{K}, \mathcal{L})$ and $\text{Can}(\mathbb{K}, \mathcal{L}^*)$ are equal.

Proof. Consider an implication $P \rightarrow P^{II}$ in $\text{Can}(\mathbb{K}, \mathcal{L})$, i.e., P is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L} . According to Lemma XLIII, $P^{\mathcal{L}^*}$ is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* such that $P^{\mathcal{L}^*} \cap M = P$. From Definition XLII we infer that $\text{Can}(\mathbb{K}, \mathcal{L}^*)$ contains the implication $P^{\mathcal{L}^*} \cap M \rightarrow (P^{\mathcal{L}^*} \cap M)^{II}$, which equals $P \rightarrow P^{II}$.

In the opposite direction, let $(P \cap M) \rightarrow (P \cap M)^{II}$ be an implication in $\text{Can}(\mathbb{K}, \mathcal{L}^*)$, i.e., P is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L}^* . By Lemma XLIII it follows that $P \cap M$ is a pseudo-intent of \mathbb{K} w.r.t. \mathcal{L} , and thus $\text{Can}(\mathbb{K}, \mathcal{L})$ contains the implication $(P \cap M) \rightarrow (P \cap M)^{II}$. \square

The following example shows that a background implication set can have an exponentially smaller representation in form of a conservative extension, just like $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ is exponentially smaller than $\mathcal{L}_{\mathcal{I}, \mathcal{T}}$.

Example XLV. For a number $\ell \geq 0$, consider the implication set $\mathcal{L} := \{U \rightarrow \{\#_n\} \mid U \subseteq \{x_1, \dots, x_\ell\} \text{ and } n = |U|\}$ over the attribute set $M := \{x_1, \dots, x_\ell\} \cup \{\#_0, \dots, \#_\ell\}$. The implications in \mathcal{L} count the number of elements in a subset of $\{x_1, \dots, x_\ell\}$. More specifically, the cardinality of a subset $U \subseteq \{x_1, \dots, x_\ell\}$ equals the largest n for which $\#_n$ is contained in the closure $U^\mathcal{L}$. Note that the number of implications in \mathcal{L} is exponential in M .

We define a conservative extension \mathcal{L}^* of \mathcal{L} by means of additional attributes $\#_n^{i,j}$ each of which expresses that n attributes with an index between i and j have been seen. This meaning is axiomatized as follows.

$$\begin{aligned} \mathcal{L}^* := & \{\emptyset \rightarrow \{\#_0\}\} \\ & \cup \{\{x_i\} \rightarrow \{\#_1^{i,i}\} \mid 1 \leq i \leq \ell\} \\ & \cup \left\{ \{\#_n^{i,j}, x_k\} \rightarrow \{\#_{n+1}^{i,k}\} \mid \begin{array}{l} 1 \leq i \leq j < k \leq \ell \\ \text{and } 1 \leq n < \ell \end{array} \right\} \\ & \cup \left\{ \{\#_n^{i,j}\} \rightarrow \{\#_n\} \mid \begin{array}{l} 1 \leq i \leq j \leq \ell \\ \text{and } 1 \leq n \leq \ell \end{array} \right\} \end{aligned}$$

By construction, $U^{\mathcal{L}^*} \cap M = U^\mathcal{L}$ for each subset $U \subseteq M$. Moreover, the number of implications in \mathcal{L}^* is only polynomial in M .

A.3 Implementation Details

Last, we explain further implementation details in addition to Section 4.

Computing the Background Implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$. The Instructions (BI1) and (BI3)–(BI5) can be executed in the obvious way. For Instruction (BI2) we should make a case distinction whether the involved ERs are in \mathbf{M} or in \mathbf{M}^* .

Lemma XLVI. The implications in $\mathcal{L}_{\mathcal{I}, \mathcal{T}}^*$ according to Instruction (BI2) are the following:

- $\{\exists r. X^\mathcal{I}\} \rightarrow \{\exists r. Y^\mathcal{I}\}$ for each $\exists r. X^\mathcal{I} \in \mathbf{M}$ and each $\exists r. Y^\mathcal{I} \in \mathbf{M}$ with $c(X) \subseteq c(Y)$
- $\{\exists r. X^\mathcal{I}\} \rightarrow \{\exists r. F\}$ for each $\exists r. X^\mathcal{I} \in \mathbf{M}$ and each $\exists r. F \in \mathbf{M}^*$ with $X \subseteq F^\mathcal{I}$
- $\{\exists r. E\} \rightarrow \{\exists r. F\}$ for each $\exists r. E \in \mathbf{M}^*$ and each $\exists r. F \in \mathbf{M}^*$ with $E \sqsubseteq^{\mathcal{T}_+} F$
- $\{\exists r. E\} \rightarrow \{\exists r. Y^\mathcal{I}\}$ for each $\exists r. E \in \mathbf{M}^*$ and each $\exists r. Y^\mathcal{I} \in \mathbf{M}$ with $(Y, e) \in \mathfrak{S}_{\emptyset(\mathcal{I}), \text{sat}(\mathcal{E}, \mathcal{T}_+)}$, where $E = \exists^{\text{sim}}(\mathcal{E}, e)$

Proof. Consider ERs $\exists r. X^\mathcal{I}$ and $\exists r. Y^\mathcal{I}$ in \mathbf{M} . According to Proposition 3, we have $X^\mathcal{I} \sqsubseteq^{\mathcal{T}_+} Y^\mathcal{I}$ iff. $(X^\mathcal{I})^{\mathcal{T}_+} \sqsubseteq^\emptyset Y^\mathcal{I}$. Since \mathcal{I} is a model of \mathcal{T}_+ , the \sqsubseteq_+ -Rule is not applicable to $X^\mathcal{I}$, and thus $(X^\mathcal{I})^{\mathcal{T}_+}$ equals $X^\mathcal{I}$. Recall that, since X and Y are generators, $c(X) = X^{\mathcal{I}\mathcal{I}}$ and $c(Y) = Y^{\mathcal{I}\mathcal{I}}$ by Lemma XXX. On the one hand, if $X^\mathcal{I} \sqsubseteq^\emptyset Y^\mathcal{I}$, then $X^{\mathcal{I}\mathcal{I}} \subseteq Y^{\mathcal{I}\mathcal{I}}$ by Property (G5), and thus $c(X) \subseteq c(Y)$. On the other hand, $c(X) \subseteq c(Y)$ implies $X^\mathcal{I} \sqsubseteq^\emptyset Y^\mathcal{I}$ by Properties (G2) and (G4). Altogether we have shown that $X^\mathcal{I} \sqsubseteq^{\mathcal{T}_+} Y^\mathcal{I}$ iff. $c(X) \subseteq c(Y)$.

Next, let $\exists r. X^\mathcal{I} \in \mathbf{M}$ and $\exists r. F \in \mathbf{M}^*$. We have $X^\mathcal{I} \sqsubseteq^{\mathcal{T}_+} F$ iff. $(X^\mathcal{I})^{\mathcal{T}_+} \sqsubseteq^\emptyset F$ (by Proposition 3) iff. $X^\mathcal{I} \sqsubseteq^\emptyset F$ (as above) iff. $X \subseteq F^\mathcal{I}$ (by Property (G1)).

The condition for two ERs $\exists r.E$ and $\exists r.F$ in \mathbf{M}^* is as in the definition of $\mathcal{L}_{\mathcal{I},\mathcal{T}}^*$ and so there is nothing to show.

Last, assume $\exists r.E \in \mathbf{M}^*$ and $\exists r.Y^{\mathcal{I}} \in \mathbf{M}$. Then $E \sqsubseteq^{\mathcal{T}_+} Y^{\mathcal{I}}$ iff. $E^{\mathcal{T}_+} \sqsubseteq^{\emptyset} Y^{\mathcal{I}}$. Recall that $E^{\mathcal{T}_+} = \exists^{\text{sim}}(\text{sat}(\mathcal{E}, \mathcal{T}_+), e)$ where $E \equiv^{\emptyset} \exists^{\text{sim}}(\mathcal{E}, e)$. Thus $E^{\mathcal{T}_+} \sqsubseteq^{\emptyset} Y^{\mathcal{I}}$ iff. $(\wp(\mathcal{I}), Y) \preceq (\text{sat}(\mathcal{E}, \mathcal{T}_+), e)$ iff. $(Y, e) \in \mathfrak{S}_{\wp(\mathcal{I}), \text{sat}(\mathcal{E}, \mathcal{T}_+)}$. \square

The proof of Proposition XL shows that, in the above implications, it suffices to consider ERs $\exists r.E \in \mathbf{M}^*$ that occur in conclusions of CIs in \mathcal{T}_+ and, similarly, ERs $\exists r.F \in \mathbf{M}^*$ that occur in premises of CIs in \mathcal{T}_+ . This allows to use a slightly smaller set of background implications in an implementation.

In the role-depth-bounded case, the second implication type is characterized by $X \subseteq F^{\mathcal{I}}$ if F satisfies the bound. Otherwise, one would really need to check whether $X^{\mathcal{I}} \sqsubseteq^{\mathcal{T}_+} F$. All these subsumptions where F is an \mathcal{EL} CD can efficiently be precomputed with ELK. For the others one would need to saturate the left-hand side by means of the \sqsubseteq_+ -Rule and then decide the subsumption by checking existence of a simulation.

Computing the CI Base. Recall from Section 4.6 that we compute the canonical implication base with LinCbO. Since our background knowledge \mathcal{L} comes as a conservative extension \mathcal{L}^* , further modifications are necessary. In order to continue employing the improved LinClosure with reused counters, we must now operate with subsets of the extended attribute set $M \cup M^*$. In principle, we need to supplement a subset of M with a part that is only needed to store the additional attributes added by closing under \mathcal{L}^* . According to Definition XLII, such a subset $D \subseteq M \cup M^*$ represents an intent if the part $D \cap M$ is an intent, i.e. if $D \cap M = (D \cap M)^{II}$. Whenever a subset D has been found that represents no intent and is closed under all previously computed implications as well as under all background implications, then D represents a pseudo-intent and thus we store the implication $D \cap M \rightarrow (D \cap M)^{II}$.

Consequently, we modify Algorithm 5 in (Janošík, Konečný, Krajča, 2022b) as follows:

- Line 5: replace $D \neq D^{II}$ with $D \cap M \neq (D \cap M)^{II}$
- Lines 6, 8: replace $D \rightarrow D^{II}$ with $D \cap M \rightarrow (D \cap M)^{II}$
- Line 7: replace $i \in D$ with $i \in D \cap M$
- Lines 9, 10: replace D^{II} with $(D \cap M)^{II} \cup D$

Moreover, we still assume that $M = \{1, \dots, n\}$ is the set of attributes of the formal context and further that $M^* = \{n+1, \dots, m\}$ is the set of additional attributes used in \mathcal{L}^* . In Line 12 we still iterate over attributes i below n only.

Acknowledgements

The author thanks the anonymous reviewers for their helpful feedback. The author is funded by Deutsche Forschungsgemeinschaft (DFG) in Projects 430150274 (Repairing Description Logic Ontologies) and 389792660 (TRR 248: Foundations of Perspicuous Software Systems).

References

- Nacira Abbas, Alexandre Bazin, Jérôme David, Amedeo Napoli (2021). Non-Redundant Link Keys in RDF Data: Preliminary Steps. In: *Proceedings of the 9th International Workshop "What can FCA do for Artificial Intelligence?" co-located with the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), Montréal, Québec, Canada, August 21, 2021*. Ed. by Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph. Vol. 2972. CEUR Workshop Proceedings. CEUR-WS.org, pp. 125–130. URL: <https://ceur-ws.org/Vol-2972/paper12.pdf>.
- Nacira Abbas, Alexandre Bazin, Jérôme David, Amedeo Napoli (2022). A Study of the Discovery and Redundancy of Link Keys Between Two RDF Datasets Based on Partition Pattern Structures. In: *Proceedings of the Sixteenth International Conference on Concept Lattices and Their Applications (CLA 2022) Tallinn, Estonia, June 20-22, 2022., Tallinn, Estonia, June 20-22, 2022*. Ed. by Pablo Cordero, Ondrej Krídlo. Vol. 3308. CEUR Workshop Proceedings. CEUR-WS.org, pp. 175–189. URL: <https://ceur-ws.org/Vol-3308/Paper14.pdf>.
- Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, Tommaso Salvatori (2020). BoxE: A Box Embedding Model for Knowledge Base Completion. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, Hsuan-Tien Lin. URL: <https://proceedings.neurips.cc/paper/2020/file/6dbbe6abe5f14af882ff977fc3f35501-Supplemental.pdf>.
- Ghadah Alghamdi, Renate A. Schmidt, Warren Del-Pinto, Yongsheng Gao (2021). Upwardly Abstracted Definition-Based Subontologies. In: *K-CAP ’21: Knowledge Capture Conference, Virtual Event, USA, December 2-3, 2021*. Ed. by Anna Lisa Gentile, Rafael Gonçalves. ACM, pp. 209–216. DOI: 10.1145/3460210.3493564.
- Simon Andrews (2011). In-Close2, a High Performance Formal Concept Miner. In: *Conceptual Structures for Discovering Knowledge – 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK, July 25-29, 2011. Proceedings*. Ed. by Simon Andrews, Simon Polovina, Richard Hill, Babak Akhgar. Vol. 6828. Lecture Notes in Computer Science. Springer, pp. 50–62. DOI: 10.1007/978-3-642-22688-5_4.
- Simon Andrews (2014). A Partial-Closure Canonicity Test to Increase the Efficiency of CbO-Type Algorithms. In: *Graph-Based Representation and Reasoning – 21st International Conference on Conceptual Structures, ICCS 2014, Iași, Romania, July 27-30, 2014, Proceedings*. Ed. by Nathalie Hernandez, Robert Jäschke, Madalina Croitoru. Vol. 8577. Lecture Notes in Computer Science. Springer, pp. 37–50. DOI: 10.1007/978-3-319-08389-6_5.
- Simon Andrews (2017). Making Use of Empty Intersections to Improve the Performance of CbO-Type Algorithms. In: *Formal Concept Analysis – 14th International Conference,*

- ICFCA 2017, Rennes, France, June 13-16, 2017, *Proceedings*. Ed. by Karell Bertet, Daniel Borchmann, Peggy Cellier, Sébastien Ferré. Vol. 10308. Lecture Notes in Computer Science. Springer, pp. 56–71. DOI: 10.1007/978-3-319-59271-8_4.
- Simon Andrews (2018). A New Method for Inheriting Canonicity Test Failures in Close-by-One Type Algorithms. In: *Proceedings of the Fourteenth International Conference on Concept Lattices and Their Applications, CLA 2018, Olomouc, Czech Republic, June 12-14, 2018*. Ed. by Dmitry I. Ignatov, Lhouari Nourine. Vol. 2123. CEUR Workshop Proceedings. CEUR-WS.org, pp. 255–266. URL: <http://ceur-ws.org/Vol-2123/paper21.pdf>.
- Shima Asaadi, Eugenie Giesbrecht, Sebastian Rudolph (2023). Compositional matrix-space models of language: Definitions, properties, and learning methods. In: *Nat. Lang. Eng.* 29.1, pp. 32–80. DOI: 10.1017/S1351324921000206.
- Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, Jérémy Vizzini (2020). Link key candidate extraction with relational concept analysis. In: *Discret. Appl. Math.* 273, pp. 2–20. DOI: 10.1016/j.dam.2019.02.012.
- Franz Baader (1995). Computing a Minimal Representation of the Subsumption Lattice of all Conjunctions of Concepts Defined in a Terminology. In: *Proceedings of the International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency, KRUSE 95*. Santa Cruz, USA, pp. 168–178.
- Franz Baader (2003). Terminological Cycles in a Description Logic with Existential Restrictions. In: *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*. Ed. by Georg Gottlob, Toby Walsh. Morgan Kaufmann, pp. 325–330. URL: <http://ijcai.org/Proceedings/03/Papers/048.pdf>.
- Franz Baader, Sebastian Brandt, Carsten Lutz (2005). Pushing the \mathcal{EL} Envelope. In: *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 – August 5, 2005*. Ed. by Leslie Pack Kaelbling, Alessandro Saffiotti. Professional Book Center, pp. 364–369. URL: <http://ijcai.org/Proceedings/05/Papers/0372.pdf>.
- Franz Baader, Sebastian Brandt, Carsten Lutz (2008). Pushing the \mathcal{EL} Envelope Further. In: *Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions, Washington, DC, USA, 1-2 April 2008*. Ed. by Kendall Clark, Peter F. Patel-Schneider. Vol. 496. CEUR Workshop Proceedings. CEUR-WS.org. URL: http://ceur-ws.org/Vol-496/owlled2008dc_paper_3.pdf.
- Franz Baader, Felix Distel (2008). A Finite Basis for the Set of \mathcal{EL} -Implications Holding in a Finite Model. In: *Formal Concept Analysis, 6th International Conference, ICFCA 2008, Montreal, Canada, February 25-28, 2008, Proceedings*. Ed. by Raoul Medina, Sergei A. Obiedkov. Vol. 4933. Lecture Notes in Computer Science. Springer, pp. 46–61. DOI: 10.1007/978-3-540-78137-0_4.
- Franz Baader, Felix Distel (2009). Exploring Finite Models in the Description Logic $\mathcal{EL}_{\text{gfp}}$. In: *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*. Ed. by Sébastien Ferré, Sebastian Rudolph. Vol. 5548. Lecture Notes in Computer Science. Springer, pp. 146–161. DOI: 10.1007/978-3-642-01815-2_12.
- Franz Baader, Bernhard Ganter, Barış Sertkaya, Ulrike Sattler (2007). Completing Description Logic Knowledge Bases Using Formal Concept Analysis. In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. Ed. by Manuela M. Veloso, pp. 230–235. URL: <http://ijcai.org/Proceedings/07/Papers/035.pdf>.
- Franz Baader, Ian Horrocks, Carsten Lutz, Ulrike Sattler (2017). *An Introduction to Description Logic*. Cambridge University Press. DOI: 10.1017/9781139025355.
- Franz Baader, Ralf Molitor (2000). Building and Structuring Description Logic Knowledge Bases Using Least Common Subsumers and Concept Analysis. In: *Conceptual Structures: Logical, Linguistic, and Computational Issues, 8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany, August 14-18, 2000, Proceedings*. Ed. by Bernhard Ganter, Guy W. Mineau. Vol. 1867. Lecture Notes in Computer Science. Springer, pp. 292–305. DOI: 10.1007/10722280_20.
- Franz Baader, Barış Sertkaya (2004). Applying Formal Concept Analysis to Description Logics. In: *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*. Ed. by Peter W. Eklund. Vol. 2961. Lecture Notes in Computer Science. Springer, pp. 261–286. DOI: 10.1007/978-3-540-24651-0_24.
- Mikhail A. Babin, Sergei O. Kuznetsov (2010). Recognizing Pseudo-intents is coNP-complete. In: *Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain, October 19-21, 2010*. Ed. by Marzena Kryszkiewicz, Sergei A. Obiedkov. Vol. 672. CEUR Workshop Proceedings. CEUR-WS.org, pp. 294–301. URL: <http://ceur-ws.org/Vol-672/paper26.pdf>.
- Mikhail A. Babin, Sergei O. Kuznetsov (2013). Computing premises of a minimal cover of functional dependencies is intractable. In: *Discret. Appl. Math.* 161.6, pp. 742–749. DOI: 10.1016/j.dam.2012.10.026.
- Catriel Beeri, Philip A. Bernstein (1979). Computational Problems Related to the Design of Normal Form Relational Schemas. In: *ACM Trans. Database Syst.* 4.1, pp. 30–59. DOI: 10.1145/320064.320066.
- Bart Bogaerts, Maxime Jakubowski, Jan Van den Bussche (2022). SHACL: A Description Logic in Disguise. In: *Logic Programming and Nonmonotonic Reasoning - 16th International Conference, LPNMR 2022, Genova, Italy, September 5-9, 2022, Proceedings*. Ed. by Georg Gottlob, Daniela Inglezan, Marco Maratea. Vol. 13416. Lecture Notes in Computer Science. Springer, pp. 75–88. DOI: 10.1007/978-3-031-15707-3_7.

- Daniel Borchmann (2013). Towards an Error-Tolerant Construction of \mathcal{EL}^+ -Ontologies from Data Using Formal Concept Analysis. In: *Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings*. Ed. by Peggy Cellier, Felix Distel, Bernhard Ganter. Vol. 7880. Lecture Notes in Computer Science. Springer, pp. 60–75. DOI: 10.1007/978-3-642-38317-5_4.
- Daniel Borchmann (2015). Exploring Faulty Data. In: *Formal Concept Analysis – 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings*. Ed. by Jaume Baixeries, Christian Sacarea, Manuel Ojeda-Aciego. Vol. 9113. Lecture Notes in Computer Science. Springer, pp. 219–235. DOI: 10.1007/978-3-319-19545-2_14.
- Daniel Borchmann, Felix Distel, Francesco Kriegel (2016). Axiomatisation of general concept inclusions from finite interpretations. In: *J. Appl. Non Class. Logics* 26.1, pp. 1–46. DOI: 10.1080/11663081.2016.1168230.
- Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, Xi Chen, Erik Bryhn Myklebust (2023). An assertion and alignment correction framework for large scale knowledge bases. In: *Semantic Web* 14.1, pp. 29–53. DOI: 10.3233/SW-210448.
- Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, Ulrike Sattler (2008). Modular Reuse of Ontologies: Theory and Practice. In: *J. Artif. Intell. Res.* 31, pp. 273–318. DOI: 10.1613/jair.2375.
- Felix Distel (2010). Hardness of Enumerating Pseudo-intents in the Llectic Order. In: *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*. Ed. by Léonard Kwuida, Barış Sertkaya. Vol. 5986. Lecture Notes in Computer Science. Springer, pp. 124–137. DOI: 10.1007/978-3-642-11928-6_9.
- Felix Distel (2011). *Learning description logic knowledge bases from data using methods from formal concept analysis*. Doctoral thesis. Technische Universität Dresden. URL: <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-70199>.
- Felix Distel, Barış Sertkaya (2011). On the complexity of enumerating pseudo-intents. In: *Discret. Appl. Math.* 159.6, pp. 450–466. DOI: 10.1016/j.dam.2010.12.004.
- Andreas Ecke, Rafael Peñaloza, Anni-Yasmin Turhan (2015). Similarity-based relaxed instance queries. In: *J. Appl. Log.* 13.4, pp. 480–508. DOI: 10.1016/j.jal.2015.01.002.
- Maurice Funk, Jean Christoph Jung, Carsten Lutz (2022). Frontiers and Exact Learning of ELI Queries under DL-Lite Ontologies. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. Ed. by Luc De Raedt. ijcai.org, pp. 2627–2633. DOI: 10.24963/ijcai.2022/364.
- Maurice Funk, Jean Christoph Jung, Carsten Lutz, Hadrien Pulcini, Frank Wolter (2019). Learning Description Logic Concepts: When can Positive and Negative Examples be Separated? In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. Ed. by Sarit Kraus. ijcai.org, pp. 1682–1688. DOI: 10.24963/ijcai.2019/233.
- Bernhard Ganter (1984). *Two Basic Algorithms in Concept Analysis*. FB4-Preprint 831. Darmstadt, Germany: Technische Hochschule Darmstadt. Republished as (Ganter, 2010).
- Bernhard Ganter (2010). Two Basic Algorithms in Concept Analysis. In: *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*. Ed. by Léonard Kwuida, Barış Sertkaya. Vol. 5986. Lecture Notes in Computer Science. Springer, pp. 312–340. DOI: 10.1007/978-3-642-11928-6_22.
- Bernhard Ganter, Rudolf Wille (1999). *Formal Concept Analysis – Mathematical Foundations*. Springer. DOI: 10.1007/978-3-642-59830-2.
- Russell Greiner, Barbara A. Smith, Ralph W. Wilkerson (1989). A Correction to the Algorithm in Reiter’s Theory of Diagnosis. In: *Artif. Intell.* 41.1, pp. 79–88. DOI: 10.1016/0004-3702(89)90079-9.
- Jean-Luc Guigues, Vincent Duquenne (1986). Famille minimale d’implications informatives résultant d’un tableau de données binaires. In: *Mathématiques et Sciences Humaines* 95, pp. 5–18. URL: http://www.numdam.org/item/MSH_1986__95__5_0.pdf.
- Ricardo Guimarães, Ana Ozaki, Cosimo Persia, Barış Sertkaya (2021). Mining \mathcal{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*. AAAI Press, pp. 6367–6374. DOI: 10.1609/aaai.v35i7.16790.
- Anneke Haga, Carsten Lutz, Johannes Marti, Frank Wolter (2020). A Journey into Ontology Approximation: From Non-Horn to Horn. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. Ed. by Christian Bessiere. ijcai.org, pp. 1827–1833. DOI: 10.24963/ijcai.2020/253.
- Monika R. Henzinger, Thomas A. Henzinger, Peter W. Kopke (1995). Computing Simulations on Finite and Infinite Graphs. In: *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*. IEEE Computer Society, pp. 453–462. DOI: 10.1109/SFCS.1995.492576.
- Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph (2010). *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC Press. URL: <http://www.semantic-web-book.org/>.
- Ian Horrocks, Jordi Olivares, Valerio Cocchi, Boris Motik, Dylan Roy (2022). The Dow Jones Knowledge Graph. In: *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*. Ed. by Paul Groth, Maria-Esther Vidal, Fabian

- M. Suchanek, Pedro A. Szekely, Pavan Kapanipathi, Catia Pesquita, Hala Skaf-Molli, Minna Tamper. Vol. 13261. Lecture Notes in Computer Science. Springer, pp. 427–443. DOI: 10.1007/978-3-031-06981-9_25.
- Mathias Jackermeier, Jiaoyan Chen, Ian Horrocks (2023). Box²EL: Concept and Role Box Embeddings for the Description Logic \mathcal{EL}^{++} . In: *CoRR* abs/2301.11118. DOI: 10.48550/arXiv.2301.11118. arXiv: 2301.11118.
- Nitisha Jain, Jan-Christoph Kalo, Wolf-Tilo Balke, Ralf Krestel (2021). Do Embeddings Actually Capture Knowledge Graph Semantics? In: *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*. Ed. by Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Óscar Corcho, Petar Ristoski, Mehwish Alam. Vol. 12731. Lecture Notes in Computer Science. Springer, pp. 143–159. DOI: 10.1007/978-3-030-77385-4_9.
- Radek Janošík, Jan Konečný, Petr Krajča (2021a). LinCbO: Fast algorithm for computation of the Duquenne-Guigues basis. In: *Inf. Sci.* 572, pp. 223–240. DOI: 10.1016/j.ins.2021.04.104.
- Radek Janošík, Jan Konečný, Petr Krajča (2021b). Pruning Techniques in LinCbO for Computation of the Duquenne-Guigues Basis. In: *Formal Concept Analysis – 16th International Conference, ICFCA 2021, Strasbourg, France, June 29 – July 2, 2021, Proceedings*. Ed. by Agnès Braud, Aleksey Buzmakov, Tom Hanika, Florence Le Ber. Vol. 12733. Lecture Notes in Computer Science. Springer, pp. 91–106. DOI: 10.1007/978-3-030-77867-5_6.
- Radek Janošík, Jan Konečný, Petr Krajča (2022a). LCM from FCA point of view: A CbO-style algorithm with speed-up features. In: *Int. J. Approx. Reason.* 142, pp. 64–80. DOI: 10.1016/j.ijar.2021.11.005.
- Radek Janošík, Jan Konečný, Petr Krajča (2022b). Pruning techniques in LinCbO for the computation of the Duquenne-Guigues basis. In: *Inf. Sci.* 616, pp. 182–203. DOI: 10.1016/j.ins.2022.10.057.
- Antonio Jimeno-Yepes, Ernesto Jiménez-Ruiz, Rafael Berlanga Llavori, Dietrich Rebbholz-Schuhmann (2009). Reuse of terminological resources for efficient ontological engineering in Life Sciences. In: *BMC Bioinform.* 10.S-10, p. 4. DOI: 10.1186/1471-2105-10-S10-S4.
- Yevgeny Kazakov, Pavel Klinov (2015). Advancing ELK: Not Only Performance Matters. In: *Proceedings of the 28th International Workshop on Description Logics, Athens, Greece, June 7-10, 2015*. Ed. by Diego Calvanese, Boris Konev. Vol. 1350. CEUR Workshop Proceedings. CEUR-WS.org. URL: <http://ceur-ws.org/Vol-1350/paper-27.pdf>.
- Yevgeny Kazakov, Markus Krötzsch, František Simančík (2014). The Incredible ELK – From Polynomial Procedures to Efficient Reasoning with \mathcal{EL} Ontologies. In: *J. Autom. Reason.* 53.1, pp. 1–61. DOI: 10.1007/s10817-013-9296-3.
- Szymon Klarman, Katarina Britz (2015). Towards Unsupervised Ontology Learning from Data. In: *Proceedings of the International Workshop on Defeasible and Ampliative Reasoning, DARE 2015, co-located with the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 27, 2015*. Ed. by Richard Booth, Giovanni Casini, Szymon Klarman, Gilles Richard, Ivan José Varzinczak. Vol. 1423. CEUR Workshop Proceedings. CEUR-WS.org. URL: https://ceur-ws.org/Vol-1423/DARE-15_5.pdf.
- Jan Konečný, Petr Krajča (2021). Systematic categorization and evaluation of CbO-based algorithms in FCA. In: *Inf. Sci.* 575, pp. 265–288. DOI: 10.1016/j.ins.2021.06.024.
- Boris Konev, Carsten Lutz, Ana Ozaki, Frank Wolter (2017). Exact Learning of Lightweight Description Logic Ontologies. In: *J. Mach. Learn. Res.* 18, 201:1–201:63. URL: <http://jmlr.org/papers/v18/16-256.html>.
- Petr Krajča, Jan Outrata, Vilém Vychodil (2010). Advances in Algorithms Based on CbO. In: *Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain, October 19-21, 2010*. Ed. by Marzena Kryszkiewicz, Sergei A. Obiedkov. Vol. 672. CEUR Workshop Proceedings. CEUR-WS.org, pp. 325–337. URL: <http://ceur-ws.org/Vol-672/paper29.pdf>.
- Francesco Kriegel (2017). Acquisition of Terminological Knowledge from Social Networks in Description Logic. In: *Formal Concept Analysis of Social Networks*. Ed. by Rokia Missaoui, Sergei O. Kuznetsov, Sergei A. Obiedkov. Lecture Notes in Social Networks. Springer, pp. 97–142. DOI: 10.1007/978-3-319-64167-6_5.
- Francesco Kriegel (2019a). *Constructing and extending description logic ontologies using methods of formal concept analysis*. Doctoral thesis. Technische Universität Dresden, Germany. URL: <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-360998>. See the summary (Kriegel, 2020).
- Francesco Kriegel (2019b). Joining Implications in Formal Contexts and Inductive Learning in a Horn Description Logic. In: *Formal Concept Analysis – 15th International Conference, ICFCA 2019, Frankfurt, Germany, June 25-28, 2019, Proceedings*. Ed. by Diana Cristea, Florence Le Ber, Barış Sertkaya. Vol. 11511. Lecture Notes in Computer Science. Springer, pp. 110–129. DOI: 10.1007/978-3-030-21462-3_9.
- Francesco Kriegel (2020). Constructing and Extending Description Logic Ontologies using Methods of Formal Concept Analysis: A Dissertation Summary. In: *Künstliche Intell.* 34.3, pp. 399–403. DOI: 10.1007/s13218-020-00673-8.
- Francesco Kriegel, Daniel Borchmann (2017). NextClosures: parallel computation of the canonical base with background knowledge. In: *Int. J. Gen. Syst.* 46.5, pp. 490–510. DOI: 10.1080/03081079.2017.1349570.
- Markus Krötzsch (2019). Too Much Information: Can AI Cope with Modern Knowledge Graphs? In: *Formal Concept Analysis - 15th International Conference, ICFCA 2019, Frankfurt, Germany, June 25-28, 2019, Proceedings*. Ed. by

- Diana Cristea, Florence Le Ber, Barış Sertkaya. Vol. 11511. Lecture Notes in Computer Science. Springer, pp. 17–31. DOI: 10.1007/978-3-030-21462-3_2.
- Sergei O. Kuznetsov (1993). A fast algorithm for computing all intersections of objects from an arbitrary semilattice. In: *Nauchno-Tekhnicheskaya Informatsiya Seriya 2 – Informatzionnye protsessy i sistemy 1*, pp. 17–20.
- Sergei O. Kuznetsov (2004). On the Intractability of Computing the Duquenne-Guigues Base. In: *J. Univers. Comput. Sci.* 10.8, pp. 927–933. DOI: 10.3217/jucs-010-08-0927.
- Sergei O. Kuznetsov, Sergei A. Obiedkov (2002). Comparing performance of algorithms for generating concept lattices. In: *J. Exp. Theor. Artif. Intell.* 14.2-3, pp. 189–216. DOI: 10.1080/09528130210164170.
- Sergei O. Kuznetsov, Sergei A. Obiedkov (2006). Counting Pseudo-intents and #P-completeness. In: *Formal Concept Analysis, 4th International Conference, ICFCA 2006, Dresden, Germany, February 13-17, 2006, Proceedings*. Ed. by Rokia Missaoui, Jürg Schmid. Vol. 3874. Lecture Notes in Computer Science. Springer, pp. 306–308. DOI: 10.1007/11671404_21.
- Sergei O. Kuznetsov, Sergei A. Obiedkov (2008). Some decision and counting problems of the Duquenne-Guigues basis of implications. In: *Discret. Appl. Math.* 156.11, pp. 1994–2003. DOI: 10.1016/j.dam.2007.04.014.
- Carsten Lutz, Robert Piro, Frank Wolter (2010). Enriching \mathcal{EL} -Concepts with Greatest Fixpoints. In: *ECAI 2010 – 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*. Ed. by Helder Coelho, Rudi Studer, Michael J. Wooldridge. Vol. 215. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 41–46. DOI: 10.3233/978-1-60750-606-5-41.
- Carsten Lutz, Frank Wolter (2010). Deciding inseparability and conservative extensions in the description logic \mathcal{EL} . In: *J. Symb. Comput.* 45.2, pp. 194–228. DOI: 10.1016/j.jsc.2008.10.007.
- Carsten Lutz, Frank Wolter (2011). Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In: *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. Ed. by Toby Walsh. IJCAI/AAAI, pp. 989–995. DOI: 10.5591/978-1-57735-516-8/IJCAI11-170.
- David Maier (1983). *The Theory of Relational Databases*. Computer Science Press. URL: <http://web.cecs.pdx.edu/~maier/TheoryBook/TRD.html>.
- Bijan Parsia, Nicolas Matentzoglou, Rafael S. Gonçalves, Birte Glimm, Andreas Steigmüller (2017). The OWL Reasoner Evaluation (ORE) 2015 Competition Report. In: *J. Autom. Reason.* 59.4, pp. 455–482. DOI: 10.1007/s10817-017-9406-8. The test ontologies are available from <https://doi.org/10.5281/zenodo.18578>.
- Raymond Reiter (1987). A Theory of Diagnosis from First Principles. In: *Artif. Intell.* 32.1, pp. 57–95. DOI: 10.1016/0004-3702(87)90062-2. See the erratum (Greiner, Smith, Wilkerson, 1989).
- Sebastian Rudolph (2004). Exploring Relational Structures Via $\mathcal{FL}\mathcal{E}$. In: *Conceptual Structures at Work: 12th International Conference on Conceptual Structures, ICCS 2004, Huntsville, AL, USA, July 19-23, 2004, Proceedings*. Ed. by Karl Erich Wolff, Heather D. Pfeiffer, Harry S. Delugach. Vol. 3127. Lecture Notes in Computer Science. Springer, pp. 196–212. DOI: 10.1007/978-3-540-27769-9_13.
- Sebastian Rudolph (2007). Some Notes on Pseudo-closed Sets. In: *Formal Concept Analysis, 5th International Conference, ICFCA 2007, Clermont-Ferrand, France, February 12-16, 2007, Proceedings*. Ed. by Sergei O. Kuznetsov, Stefan Schmidt. Vol. 4390. Lecture Notes in Computer Science. Springer, pp. 151–165. DOI: 10.1007/978-3-540-70901-5_10.
- Md. Kamruzzaman Sarker, Adila Alfa Krisnadhi, Pascal Hitzler (2016). OWLax: A Protege Plugin to Support Ontology Axiomatization through Diagramming. In: *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016*. Ed. by Takahiro Kawamura, Heiko Paulheim. Vol. 1690. CEUR Workshop Proceedings. CEUR-WS.org. URL: <https://ceur-ws.org/Vol-1690/paper83.pdf>.
- Barış Sertkaya (2009a). Some Computational Problems Related to Pseudo-intents. In: *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*. Ed. by Sébastien Ferré, Sebastian Rudolph. Vol. 5548. Lecture Notes in Computer Science. Springer, pp. 130–145. DOI: 10.1007/978-3-642-01815-2_11.
- Barış Sertkaya (2009b). Towards the Complexity of Recognizing Pseudo-intents. In: *Conceptual Structures: Leveraging Semantic Technologies, 17th International Conference on Conceptual Structures, ICCS 2009, Moscow, Russia, July 26-31, 2009, Proceedings*. Ed. by Sebastian Rudolph, Frithjof Dau, Sergei O. Kuznetsov. Vol. 5662. Lecture Notes in Computer Science. Springer, pp. 284–292. DOI: 10.1007/978-3-642-03079-6_22.
- Jingchuan Shi, Jiaoyan Chen, Hang Dong, Ishita Khan, Lizzie Liang, Qunzhi Zhou, Zhe Wu, Ian Horrocks (2023). Subsumption Prediction for E-Commerce Taxonomies. In: *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*. Ed. by Catia Pesquita, Ernesto Jiménez-Ruiz, Jamie P. McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphaël Troncy, Sven Hertling. Vol. 13870. Lecture Notes in Computer Science. Springer, pp. 244–261. DOI: 10.1007/978-3-031-33455-9_15.
- Gerd Stumme (1996). Attribute Exploration with Background Implications and Exceptions. English. In: *Data Analysis and Information Systems: Statistical and Conceptual Approaches, Proceedings of the 19th Annual Conference of the Gesellschaft für Klassifikation e.V., University of Basel, March 8–10, 1995*. Ed. by Hans-Hermann Bock, Wolfgang Polasek. Studies in Classification, Data Analysis,

and Knowledge Organization. Berlin, Heidelberg: Springer, pp. 457–469. DOI: 10.1007/978-3-642-80098-6_39.

Alfred Tarski (1955). A Lattice-Theoretical Fixpoint Theorem and its Applications. In: *Pacific Journal of Mathematics* 5.2, pp. 285–309. DOI: 10.2140/pjm.1955.5.285.

Marcel Wild (1994). A Theory of Finite Closure Spaces Based on Implications. In: *Advances in Mathematics* 108.1, pp. 118–139. DOI: 10.1006/aima.1994.1069.

Rudolf Wille (1982). Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In: *Ordered Sets*. Ed. by Ivan Rival. Dordrecht: Springer Netherlands, pp. 445–470. Republished as (Wille, 2009).

Rudolf Wille (2009). Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In: *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*. Ed. by Sébastien Ferré, Sebastian Rudolph. Vol. 5548. Lecture Notes in Computer Science. Springer, pp. 314–339. DOI: 10.1007/978-3-642-01815-2_23.

Benjamin Zarriß, Anni-Yasmin Turhan (2013). Most Specific Generalizations w.r.t. General \mathcal{EL} -TBoxes. In: *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. Ed. by Francesca Rossi. IJCAI/AAAI, pp. 1191–1197. URL: <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6709>.

Yizheng Zhao, Renate A. Schmidt, Yuejie Wang, Xuanming Zhang, Hao Feng (2020). A Practical Approach to Forgetting in Description Logics with Nominals. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, pp. 3073–3079. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/5702>.

Links

- [1] <https://www.w3.org/standards/semanticweb/>
- [2] <https://www.w3.org/TR/owl2-overview/>
- [3] <https://finregont.com>
- [4] <https://spec.edmcouncil.org/fibo/>
see also <https://github.com/edmcouncil/fibo>
- [5] <https://www.snomed.org>
- [6] <http://geneontology.org>
- [7] <http://www.w3.org/TR/owl2-profiles>
- [8] <http://phoenix.inf.upol.cz/~konecnja/fcalad/>
- [9] <https://github.com/francesco-kriegel/efficient-axiomatization-of-owl2el-ontologies-from-data>
- [10] <https://www.scala-lang.org>
- [11] <https://doi.org/10.5281/zenodo.18578>
- [12] <https://docs.oracle.com/javase/tutorial/essential/concurrency/forkjoin.html>
- [13] <https://github.com/exot/EL-exploration>