Efficient Axiomatization of OWL 2 EL Ontologies from Data by means of Formal Concept Analysis

Extended Abstract[1]

Francesco Kriegel

Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany

Building and maintaining ontologies is a laborious task, especially for large domains, where knowledge engineers and domain experts work together to transfer their knowledge into an explicit representation. In Description Logic, the ABox of an ontology is usually filled with observed symbolic data but constructing the TBox is a more complex endeavor. Assistance by automated or interactive methods is often valuable. To this end, we reconsider the Formal-Concept-Analysis-based approach to completely axiomatizing $\mathcal{EL}^{\bot}$ concept inclusions $C \sqsubseteq D$ from graph data [3, 4] and

1. thoroughly revise and simplify its technical description including proofs,
2. equip it with support for already known concept inclusions satisfied in the data (thus enabling it for ontology completion),
3. analyze its computational complexity,
4. explain how further types of TBox statements supported by $\mathcal{EL}^{++}$ that are not just syntactic sugar can be completely axiomatized, viz. role inclusions $r_1 \circ \cdots \circ r_n \sqsubseteq s$ and range restrictions $\top \sqsubseteq \forall r.C$,
5. describe how it can be implemented efficiently,
6. introduce variations that dispense with the computation of disjointness statements $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq \bot$ or extremely large concept inclusions without practical relevance, thereby rendering the approach applicable in practice, albeit some completeness is lost,
7. and evaluate the implementation on real-world datasets.

Formal Concept Analysis (FCA) [5] 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. The canonical implication base is a complete set of implications, i.e. it entails all implications satisfied in the data [6, 7], and no complete set with fewer implications exists [8, 9]. In other words, the implication base axiomatizes data in form of a formal context by means of implications.

[1] This is an extended abstract of an article [1] published in the proceedings of the 38th Annual AAAI Conference on Artificial Intelligence (AAAI-24) and of its extended version [2].

DL 2024: 37th International Workshop on Description Logics, June 18–21, 2024, Bergen, Norway

francesco.kriegel@tu-dresden.de (F. Kriegel)
0000-0003-0219-0330 (F. Kriegel)

CEUR Workshop Proceedings (CEUR-WS.org)

We employ the description logic $\mathcal{EL}^{++}$ [10], but we ignore nominals to avoid overfitting in the axiomatization method and concrete domains (datatypes for strings, numbers, etc.) as no $\mathcal{EL}^{++}$ reasoner currently supports them. The Web Ontology Language includes $\mathcal{EL}^{++}$ as the profile OWL 2 EL. By exploiting the similarity between concept inclusions and FCA implications, a complete TBox of concept inclusions can be axiomatized from observed data, i.e. which entails all concept inclusions satisfied in the data. More specifically, as input we expect graph data in form of an interpretation $\mathcal{I}$, which includes knowledge graphs, graph databases, and RDF data: the concept names are the node labels and the role names 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 [11]. Further given may be an already existing TBox $\mathcal{T}$ consisting of concept inclusions satisfied in $\mathcal{I}$ and relative to which $\mathcal{I}$ is to be axiomatized, or rather, of which we compute a completion w.r.t. $\mathcal{I}$. The axiomatization result is called a concept inclusion base or a completion in the following sense.

[1, Definition 4]. A TBox is *complete* for $\mathcal{I}$ if it entails all concept inclusions satisfied in $\mathcal{I}$. A *concept inclusion 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 $\mathcal{I}$. We may also call $\mathcal{B}$ a *completion* of $\mathcal{T}$ w.r.t. $\mathcal{I}$ as we obtain a complete TBox by adding all concept inclusions in $\mathcal{B}$ to $\mathcal{T}$.

A *canonical* completion of $\mathcal{T}$ w.r.t. $\mathcal{I}$ can be computed in exponential time [1, Theorem 10] (see below). This complexity result is tight since there are finite interpretations without any polynomial-size base.[1] Moreover, if all concept inclusions in $\mathcal{T}$ come in a particular normal form, then no base with fewer concept inclusions exists. In order to efficiently treat cycles in the input (both contained in $\mathcal{I}$ or induced by $\mathcal{T}$), the concept inclusions are formulated in an extension of $\mathcal{EL}^{\bot}$ that allows for non-tree-shaped concept descriptions. To obtain a usual $\mathcal{EL}^{++}$ TBox, the canonical concept inclusion base afterwards needs to be rewritten using variables.

In order to compute the canonical completion, the given interpretation $\mathcal{I}$ is transformed into a formal context $\mathbb{K}_{\mathcal{I}}$. Its finite attribute set $\mathbf{M}$ is specifically designed such that there is a concept inclusion base involving only conjunctions over $\mathbf{M}$, i.e. which consists of inclusions $\bigsqcap \mathbf{C} \sqsubseteq \bigsqcap \mathbf{D}$ for subsets $\mathbf{C}, \mathbf{D} \subseteq \mathbf{M}$ [1, Lemma 9]. This setting is perfectly suited for FCA since all necessary computations happen in the top-level conjunctions and there is no need to look inside the concept descriptions provided by $\mathbf{M}$ (which FCA could not do anyway). In particular, we first use an efficient FCA algorithm [13, 14] to compute the canonical base $\mathsf{Can}(\mathbb{K}_{\mathcal{I}})$, which consists of FCA implications $\mathbf{C} \to \mathbf{D}$ (i.e. $\bigwedge \mathbf{C} \to \bigwedge \mathbf{D}$ in logical notation), and we then rewrite these implications into inclusions $\bigsqcap \mathbf{C} \sqsubseteq \bigsqcap \mathbf{D}$ to obtain a minimal concept inclusion base of $\mathcal{I}$.

Moreover, the formal context $\mathbb{K}_{\mathcal{I}}$ is axiomatized relative to the implication set $\mathcal{L}_{\mathcal{I},\mathcal{T}}$. On the one hand, $\mathcal{L}_{\mathcal{I},\mathcal{T}}$ contains the implication $\{E\} \to \{F\}$ for each two concept descriptions $E, F \in \mathbf{M}$ with $E \sqsubseteq^{\emptyset} F$, viz. to avoid the axiomatization of tautological concept inclusions. On the other hand, the given TBox $\mathcal{T}$ is taken into account by transforming all its inclusions into implications over $\mathbf{M}$ and adding these to $\mathcal{L}_{\mathcal{I},\mathcal{T}}$.

[1, Theorem 10]. For each finite interpretation $\mathcal{I}$ and each $\mathcal{EL}^{\bot}_{\mathsf{si}}$ TBox $\mathcal{T}$ of which $\mathcal{I}$ is a model, the TBox $\mathsf{Can}(\mathcal{I}, \mathcal{T}) := \bigsqcap \mathsf{Can}(\mathbb{K}_{\mathcal{I}}, \mathcal{L}_{\mathcal{I},\mathcal{T}})$ is a concept inclusion base of $\mathcal{I}$ relative to $\mathcal{T}$. It

[1] Since each formal context can be seen as an interpretation without role names, this is an immediate corollary to a result in FCA: there is a sequence of formal contexts with $3{\cdot}n$ objects and $2{\cdot}n{+}1$ attributes for which the number of implications in their canonical implication bases is exponential in n [12].

is called *canonical concept inclusion base* and can be computed in time that is exponential in $\mathsf{Dom}(\mathcal{I})$ and polynomial in $\mathcal{T}$. If all concept inclusions in $\mathcal{T}$ have the form $C \sqsubseteq D^{[\mathcal{I}\mathcal{I}]}$, then it contains the fewest inclusions among all concept inclusion bases of $\mathcal{I}$ relative to $\mathcal{T}$. Furthermore, there are finite interpretations that have no polynomial-size concept inclusion base.

In addition to concept inclusions, $\mathcal{EL}^{++}$ further supports role inclusions and range restrictions. The latter can easily be read off from the input: for each role name r, we first compute the most specific concept description C that has all r-successors in $\mathcal{I}$ as instances and we then add the range restriction $\top \sqsubseteq \forall r.C$ to the TBox.

All role inclusions can be completely axiomatized by viewing $\mathcal{I}$ as a finite automaton. Specifically for objects x, y in $\mathcal{I}$ we denote by $\mathfrak{A}_{x,y}$ the automaton with initial state x and final state y. Now a role inclusion $r_1 \circ \cdots \circ r_n \sqsubseteq s$ is not satisfied in $\mathcal{I}$ iff. there are objects x, y in $\mathcal{I}$ with $(x, y) \in (r_1 \circ \cdots \circ r_n)^{\mathcal{I}}$ but $(x, y) \notin s^{\mathcal{I}}$. By definition, $(x, y) \in (r_1 \circ \cdots \circ r_n)^{\mathcal{I}}$ iff. the automaton $\mathfrak{A}_{x,y}$ accepts the word $r_1 \cdots r_n$. So, the complement automaton of the union automaton of all $\mathfrak{A}_{x,y}$ with $(x, y) \notin s^{\mathcal{I}}$ accepts the word $r_1 \cdots r_n$ iff. the role inclusion $r_1 \circ \cdots \circ r_n \sqsubseteq s$ is satisfied in $\mathcal{I}$. Like for the concept inclusions, we use variables to formulate the axiomatized role inclusions: $p \circ r \sqsubseteq q$ for each transition (p, r, q), and $\varepsilon \sqsubseteq i$ for each initial state, and $f \sqsubseteq s$ for each final state. It is decidable whether there are equivalent role inclusions without variables [15], but this seems unnecessary since most reasoners transform role inclusions into finite automata anyway. All in all, we obtain the following main result.

[1, Theorem 13]. For each finite interpretation $\mathcal{I}$, a complete TBox of $\mathcal{EL}^{\bot}$ concept inclusions, range restrictions, and role inclusions satisfied in $\mathcal{I}$ can be computed in exponential time. There are finite interpretations for which such a TBox cannot be of polynomial size.

A technical limitation is that completeness does not go together with the syntactic restriction on the interplay of role inclusions and range restrictions in an $\mathcal{EL}^{++}$ TBox $\mathcal{T}$ that ensures tractable reasoning [10]: for each role inclusion $r_1 \circ \cdots \circ r_n \sqsubseteq s$ in $\mathcal{T}$ where $n \geq 1$, if $\mathcal{T}$ does not entail the range restriction $\top \sqsubseteq \forall r_n.C$, then $\mathcal{T}$ neither entails $\top \sqsubseteq \forall s.C$ (i.e. new concept memberships for objects in the range of s are forbidden). This restriction might not be satisfied by a TBox that is complete for both role inclusions and range restrictions [2, Example XXIII].

To ensure that the TBox is within $\mathcal{EL}^{++}$, we could weaken the range restrictions: for each role name s, we obtain a suitable range restriction $\top \sqsubseteq \forall s.C$ by computing the most specific concept description C that has all s-successors in $\mathcal{I}$ as instances (as above) but also all r-successors for each role name r leading to a final state (since these represent role inclusions $\cdots \circ r \sqsubseteq s$).

Alternatively, we could remove all role inclusions that contribute to a violation of the syntactic restriction (by simply computing a language difference in the above automaton representation) and leave the range restrictions unchanged. To sum up, only two of the following goals can be achieved: the base satisfies the syntactic restriction, the base is complete for all range restrictions, the base is complete for all role inclusions.

Regarding efficient implementation, it is important to reduce the input interpretation by grouping together all objects that cannot be distinguished by any concept description. Thereby the interpretation can be made significantly smaller and all subsequent steps need less time. However, in first experiments several computations did not finish due to extremely large concept descriptions used in the concept inclusion base to ensure completeness. We conjecture that such huge parts in the concept inclusion base do not have practical relevance or suffer from overfitting, and thus we added parameters (conjunction size limit and role depth bound) to dispense with the computation of these irrelevant huge parts but also to control the loss of completeness. Experiments further revealed that often more than half of the computation time is required for generating disjointness statements $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq \bot$. We explain how intermediate computation steps can be stopped early in order to avoid computing them.

We implemented the method in the programming language Scala and we evaluated the prototype with the plethora of ontologies from real-world applications used in the ORE 2015 Reasoner Competition. This collection is split into OWL 2 EL and OWL 2 DL ontologies. The former are all expressible in $\mathcal{EL}^{++}$. For the latter, we syntactically transform as many axioms as possible into $\mathcal{EL}^{++}$ and remove the others. The goal then was to compute, for each such ontology, the completion of the TBox $\mathcal{T}$ w.r.t. the interpretation $\mathcal{I}$ obtained by viewing the respective ABox under closed-world assumption. To ensure that $\mathcal{I}$ is a model of $\mathcal{T}$, we saturate $\mathcal{I}$ by means of the inclusions in $\mathcal{T}$ if necessary. Altogether we obtained 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. Computation of concept inclusion bases 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 restrictions. 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 on a twelve-year-old computer server. However, we did not implement the rewriting of the concept inclusion base into $\mathcal{EL}^{++}$, nor the axiomatization of role inclusions and range restrictions.

That the theoretical approach itself can be extended to more expressive description logics has already been proven [16, 17], but it is unclear whether such an extended approach can still be efficiently implemented and used in practice. From the perspective of the underlying article [1], this seems possible for description logics characterized by simulations, e.g. $\mathcal{ELI}$ or Horn-$\mathcal{ALC}$.

An interesting question for future research would be whether one can give any kind of completeness guarantee if a conjunction size limit is used, as already done for the role depth bound [18]. A smaller task would be to investigate how role inclusions and range restrictions can be integrated into the background knowledge after they have been computed but prior to axiomatizing the concept inclusions, preferably yielding an overall minimal base.

Furthermore, the computation can be speed-up with even faster FCA algorithms for enumerating closures. The employed FCA algorithm [13, 14] 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 computer server, or optimize the prototype.

A concept inclusion $C \sqsubseteq D$ is *confident* if the ratio $|(C \sqcap D)^{\mathcal{I}}|/|C^{\mathcal{I}}|$ exceeds a pre-defined limit but need not be $100\,\%$. A confident inclusion base extends the canonical inclusion base [19], and the prototype could be easily upgraded as it already computes the necessary pieces.

We have not considered keys supported by the OWL 2 EL profile. Learning of keys from RDF data using FCA has been addressed [20–22]. To apply this approach to description logic and OWL it must be extended towards concept descriptions in place of concept names (RDF classes).

Acknowledgments

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) in Projects 430150274 (Repairing Description Logic Ontologies) and 389792660 (TRR 248: Foundations of Perspicuous Software Systems), and has further been supported by the Saxon State Ministry for Science, Culture, and Tourism (SMWK) by funding the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI).

References

[1] Francesco Kriegel. Efficient Axiomatization of OWL 2 EL Ontologies from Data by means of Formal Concept Analysis. In: *Proc. of AAAI*. 2024. DOI: 10.1609/aaai.v38i9.28930.

[2] Francesco Kriegel. *Efficient Axiomatization of OWL 2 EL Ontologies from Data by means of Formal Concept Analysis (Extended Version)*. LTCS-Report 23-01. Technische Universität Dresden, 2023. DOI: 10.25368/2023.214. See also the addendum: 10.5281/zenodo.10908141.

[3] Franz Baader, Felix Distel. A Finite Basis for the Set of $\mathcal{EL}$-Implications Holding in a Finite Model. In: *Proc. of ICFCA*. 2008, pp. 46–61. DOI: 10.1007/978-3-540-78137-0_4.

[4] Franz Baader, Felix Distel. Exploring Finite Models in the Description Logic $\mathcal{EL}_{\text{gfp}}$. In: *Proc. of ICFCA*. 2009, pp. 146–161. DOI: 10.1007/978-3-642-01815-2_12.

[5] Bernhard Ganter, Rudolf Wille. *Formal Concept Analysis – Mathematical Foundations*. 1999. DOI: 10.1007/978-3-642-59830-2.

[6] Jean-Luc Guigues, Vincent Duquenne. Famille minimale d'implications informatives résultant d'un tableau de données binaires. In: *Mathématiques et Sciences Humaines* 95 (1986), pp. 5–18. URL: http://www.numdam.org/item/MSH_1986__95__5_0.pdf.

[7] Gerd Stumme. Attribute Exploration with Background Implications and Exceptions. In: 1996, pp. 457–469. DOI: 10.1007/978-3-642-80098-6_39.

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

[9] Felix Distel. *Learning description logic knowledge bases from data using methods from formal concept analysis*. Doctoral thesis. Technische Universität Dresden, 2011. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-70199.

[10] Franz Baader, Sebastian Brandt, Carsten Lutz. Pushing the $\mathcal{EL}$ Envelope Further. In: *Proc. of OWLED*. 2008. URL: http://ceur-ws.org/Vol-496/owled2008dc_paper_3.pdf.

[11] Markus Krötzsch. Too Much Information: Can AI Cope with Modern Knowledge Graphs? In: *Proc. of ICFCA*. 2019, pp. 17–31. DOI: 10.1007/978-3-030-21462-3_2.

[12] Sergei O. Kuznetsov. On the Intractability of Computing the Duquenne-Guigues Base. In: *J. Univers. Comput. Sci.* 10.8 (2004), pp. 927–933. DOI: 10.3217/JUCS-010-08-0927.

[13] Radek Janoštík, Jan Konečný, Petr Krajča. LinCbO: Fast algorithm for computation of the Duquenne-Guigues basis. In: *Inf. Sci.* 572 (2021), pp. 223–240. DOI: 10.1016/j.ins.2021.04.104.

[14] Radek Janoštík, Jan Konečný, Petr Krajča. Pruning techniques in LinCbO for the computation of the Duquenne-Guigues basis. In: *Inf. Sci.* 616 (2022), pp. 182–203. DOI: 10.1016/j.ins.2022.10.057.

[15] Walter Bucher, Johann Hagauer. It is Decidable Whether a Regular Language is Pure Context-Free. In: *Theor. Comput. Sci.* 26 (1983), pp. 233–241. DOI: 10.1016/0304-3975(83)90088-9.

[16] Francesco Kriegel. Acquisition of Terminological Knowledge from Social Networks in Description Logic. In: *Formal Concept Analysis of Social Networks.* 2017, pp. 97–142. DOI: 10.1007/978-3-319-64167-6_5.

[17] Francesco Kriegel. Joining Implications in Formal Contexts and Inductive Learning in a Horn Description Logic. In: *Proc. of ICFCA.* 2019, pp. 110–129. DOI: 10.1007/978-3-030-21462-3_9.

[18] Daniel Borchmann, Felix Distel, Francesco Kriegel. Axiomatisation of general concept inclusions from finite interpretations. In: *J. Appl. Non Class. Logics* 26.1 (2016), pp. 1–46. DOI: 10.1080/11663081.2016.1168230.

[19] Daniel Borchmann. Towards an Error-Tolerant Construction of $\mathcal{EL}^{\perp}$-Ontologies from Data Using Formal Concept Analysis. In: *Proc. of ICFCA.* 2013, pp. 60–75. DOI: 10.1007/978-3-642-38317-5_4.

[20] Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, Jérémy Vizzini. Link key candidate extraction with relational concept analysis. In: *Discret. Appl. Math.* 273 (2020), pp. 2–20. DOI: 10.1016/j.dam.2019.02.012.

[21] Nacira Abbas, Alexandre Bazin, Jérôme David, Amedeo Napoli. Non-Redundant Link Keys in RDF Data: Preliminary Steps. In: *Proc. of FCA4AI.* 2021, pp. 125–130. URL: https://ceur-ws.org/Vol-2972/paper12.pdf.

[22] Nacira Abbas, Alexandre Bazin, Jérôme David, Amedeo Napoli. A Study of the Discovery and Redundancy of Link Keys Between Two RDF Datasets Based on Partition Pattern Structures. In: *Proc. of CLA.* 2022, pp. 175–189. URL: https://ceur-ws.org/Vol-3308/Paper14.pdf.