Combining Proofs for Description Logic and Concrete Domain Reasoning (Extended Abstract)*

Christian Alrabbaa[1], Franz Baader[1], Stefan Borgwardt[1], Patrick Koopmann[2], and Alisa Kovtunova[1]

[1] Institute of Theoretical Computer Science, TU Dresden, Germany
`firstname.lastname@tu-dresden.de`
[2] Department of Computer Science, Vrije Universiteit Amsterdam, Netherlands
`p.k.koopmann@vu.nl`

Abstract. Reasoning in Description Logics (DLs) with numerical concrete domains combines abstract logical with concrete numerical reasoning. We show how consequences computed by such combined reasoning engines can be explained in a uniform way by proofs that integrate the numerical reasoning steps into the proofs on the DL side.

Proofs for explaining DL reasoning

For developers or users of DL-based ontologies, it is often hard to understand why a consequence computed by a DL reasoner[3] actually follows from the given, possibly very large ontology. In principle, such a consequence can be explained by producing a proof for it, i.e. by showing how the consequence can be derived from the axioms in the ontology by applying certain easy-to-understand inference rules. Until recently, work on explaining DL entailment was focused on computing so-called justifications, i.e. minimal subsets of the ontology from which the consequence in question follows [25,11,15]. With few exceptions [16,17], figuring out how the consequence can be derived from the justification was left to the user. In recent work, we have investigated how proofs for consequences derived by DL reasoners can be computed [2,3] and displayed [1] in a user-friendly way [6]. However, this work was restricted to DLs without concrete domains. In particular, we considered the DL $\mathcal{EL}_\bot$, which has concept constructors conjunction ($C \sqcap D$), existential restriction ($\exists r.C$) as well as the top ($\top$) and the bottom ($\bot$) concepts, and its extension by negation ($\neg C$), called $\mathcal{ALC}$ [10].

The need for concrete domains

Concrete domains [9,22] (CDs) have been introduced to enable reference to concrete objects (such as numbers) and predefined predicates on these objects (such

* This is an extended abstract of [4]
[3] see http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/

as numerical comparisons) when defining concepts. Using these predicates, we can then formulate constraints on the concrete values that can be associated with abstract individuals. In the presence of general concept inclusions (GCIs), integrating a CD into a DL may cause undecidability [23,12] even if solvability of the constraints that can be formulated in the CD is decidable. One way to overcome this problem is to disallow role paths [14,24,8] in concrete domain restrictions, which means that these restrictions can only constrain feature values of single individuals. Comparing feature values of different individuals, such as the age of a woman with that of her children, is then no longer possible. In the present paper, we adopt this approach. However, for tractable (i.e. polynomially decidable) DLs like $\mathcal{EL}_\bot$, preserving decidability is not sufficient: one wants to preserve tractability. As shown in [8], this is the case if one integrates a so-called p-admissible concrete domain into $\mathcal{EL}_\bot$. The only numerical p-admissible concrete domain exhibited in [8] is the CD $\mathcal{D}_{\mathbb{Q},\mathit{diff}}$, which supports constraints of the form $x = q$, $x > q$ and $x + q = y$ (for constants $q \in \mathbb{Q}$) over the rational numbers. Recently, additional p-admissible concrete domains have been introduced in [12], such as the CD $\mathcal{D}_{\mathbb{Q},\mathit{lin}}$, whose constraints are given by linear equations $\sum_{i=1}^{n} a_i x_i = b$ over $\mathbb{Q}$. In the present paper, we will concentrate on these two p-admissible CDs, though the developed ideas and techniques can also be used for other CDs. The following example illustrates how $\mathcal{D}_{\mathbb{Q},\mathit{lin}}$ can be used in a medical application.

Example 1. In an intensive care unit, values for the systolic and diastolic blood pressure, the heart rate, and the age of patients are available. We refer to them using the features sys, dia, hr, and age, respectively, which are interpreted as partial functions that assign rational numbers to patients. The pulse pressure, represented by pp, can then be defined via the GCI $\mathsf{ICUpatient} \sqsubseteq [\mathsf{sys} - \mathsf{dia} - \mathsf{pp} = 0]$, and the maximal heart rate by $\mathsf{ICUpatient} \sqsubseteq [\mathsf{maxHR} + \mathsf{age} = 220]$. If the heart rate almost reaches the maximal one or if the pulse pressure sinks to a very low value, we want to raise an alarm, which can be expressed using the GCIs $\mathsf{ICUpatient} \sqcap [\mathsf{maxHR} - \mathsf{hr} = 5] \sqsubseteq \mathsf{NeedAttention}$ and $\mathsf{ICUpatient} \sqcap [\mathsf{pp} = 25] \sqsubseteq \mathsf{NeedAttention}$.

Reasoning in DLs extended with concrete domains

We denote by $\mathcal{EL}_\bot[\mathcal{D}]$ the DL obtained by extending $\mathcal{EL}_\bot$ with any p-admissible concrete domain $\mathcal{D}$. The main reasoning task is *subsumption*, i.e. to decide whether a GCI $C \sqsubseteq D$ follows from an ontology, in which case we also say that C is subsumed by D. The paper [8] describes an inference procedure that can decide the subsumption problem in $\mathcal{EL}_\bot[\mathcal{D}]$ in polynomial time. Thus, this procedure can also be used for $\mathcal{EL}_\bot[\mathcal{D}_{\mathbb{Q},\mathit{lin}}]$ an $\mathcal{EL}_\bot[\mathcal{D}_{\mathbb{Q},\mathit{diff}}]$. Unfortunately, no implemented DL reasoner supports these two CDs. In particular, the highly efficient $\mathcal{EL}_\bot$ reasoner ELK [18] does not support any concrete domain. Instead of modifying ELK appropriately or implementing our own reasoner for $\mathcal{EL}_\bot[\mathcal{D}]$, we have developed an iterative algorithm that interleaves ELK reasoning with concrete domain reasoning. Basically, the concrete domain restrictions are abstracted away by new concepts and Elk is used to classify (i.e. compute the subsumption hierarchy) of the ontology obtained this way. If some concept C is subsumed by a collection $A_1, \dots, A_n$ of abstraction concepts, then CD reasoning is used to check whether the corresponding conjunction of constraints is unsatisfiable in the CD or implies the constraint corresponding to another abstraction concept B. In the first case, $A_1 \sqcap \dots \sqcap A_n \sqsubseteq \bot$ is added to the abstracted ontology, and in the second $A_1 \sqcap \dots \sqcap A_n \sqsubseteq B$. For example, if the $\mathcal{D}_{\mathbb{Q},lin}$ restrictions $[\mathsf{age} = 42]$, $[\mathsf{age} + \mathsf{maxHR} = 220]$, and $[\mathsf{maxHR} = 178]$ have been abstracted away by the concept names A_1, A_2, and B, respectively, and Elk has derived $C \sqsubseteq A_1$ and $C \sqsubseteq A_2$, then $A_1 \sqcap A_2 \sqsubseteq B$ is added. After this CD reasoning step, Elk is invoked again, and this interleaving of $\mathcal{EL}_\bot$ and CD reasoning is continued until no new subsumptions are computed. We can show that this approach yields a polynomial-time classification procedure for $\mathcal{EL}_\bot[\mathcal{D}]$ if $\mathcal{D}$ is p-admissible.

For the CD reasoning in $\mathcal{D}_{\mathbb{Q},lin}$ and $\mathcal{D}_{\mathbb{Q},\mathit{diff}}$, we could in principle have employed existing algorithms and implementations, like Gaussian elimination or the simplex method [26,13] for $\mathcal{D}_{\mathbb{Q},lin}$, and SMT systems that can deal with difference logic [20,7], such as Z3,[4] for $\mathcal{D}_{\mathbb{Q},\mathit{diff}}$. However, since our main purpose was to generate proofs, we developed our own reasoning procedures for $\mathcal{D}_{\mathbb{Q},\mathit{diff}}$ and $\mathcal{D}_{\mathbb{Q},lin}$, which were designed such that it is easy to extract proofs from them. For $\mathcal{D}_{\mathbb{Q},\mathit{diff}}$, this is a simple constraint propagation approach, and for $\mathcal{D}_{\mathbb{Q},lin}$ we use Gaussian elimination.

Generating proofs for DLs extended with concrete domains

Proofs for reasoning results in $\mathcal{EL}_\bot$ with a p-admissible CD can in principle be represented using the calculus introduced in [8] or an appropriate extension of the calculus employed by Elk. However, in these calculi, the result of CD reasoning (i.e. that a set of constraints is unsatisfiable or entails another constraint) is used as an applicability condition for certain rules, but the CD reasoning leading to the satisfaction of the conditions is not explained. Instead of augmenting such a proof with separate proofs on the CD side that show why the applicability conditions are satisfied, our goal was to produce a single proof that explains both the $\mathcal{EL}_\bot$ and the CD reasoning in a uniform proof format.

Our basic idea for generating proofs in $\mathcal{EL}_\bot[\mathcal{D}]$ was to look at the last run of Elk in our iterative procedure, and extract a pure $\mathcal{EL}_\bot$ proof from the abstracted ontology to which this last run was applied. This proof contains as asserted conclusions GCIs of the form $A_1 \sqcap \dots \sqcap A_n \sqsubseteq \bot$ and $A_1 \sqcap \dots \sqcap A_n \sqsubseteq B$, which have been added to the ontology due to the results of CD reasoning. In the above example, we have added the implication $A_1 \sqcap A_2 \sqsubseteq B$. A proof for the implication between the corresponding constraints based on Gaussian elimination looks as follows:

$$\frac{\mathsf{age} = 42 \qquad \mathsf{age} + \mathsf{maxHR} = 220}{\mathsf{maxHR} = 178}\,[-1, 1] \tag{1}$$

[4] https://theory.stanford.edu/~nikolaj/programmingz3.html

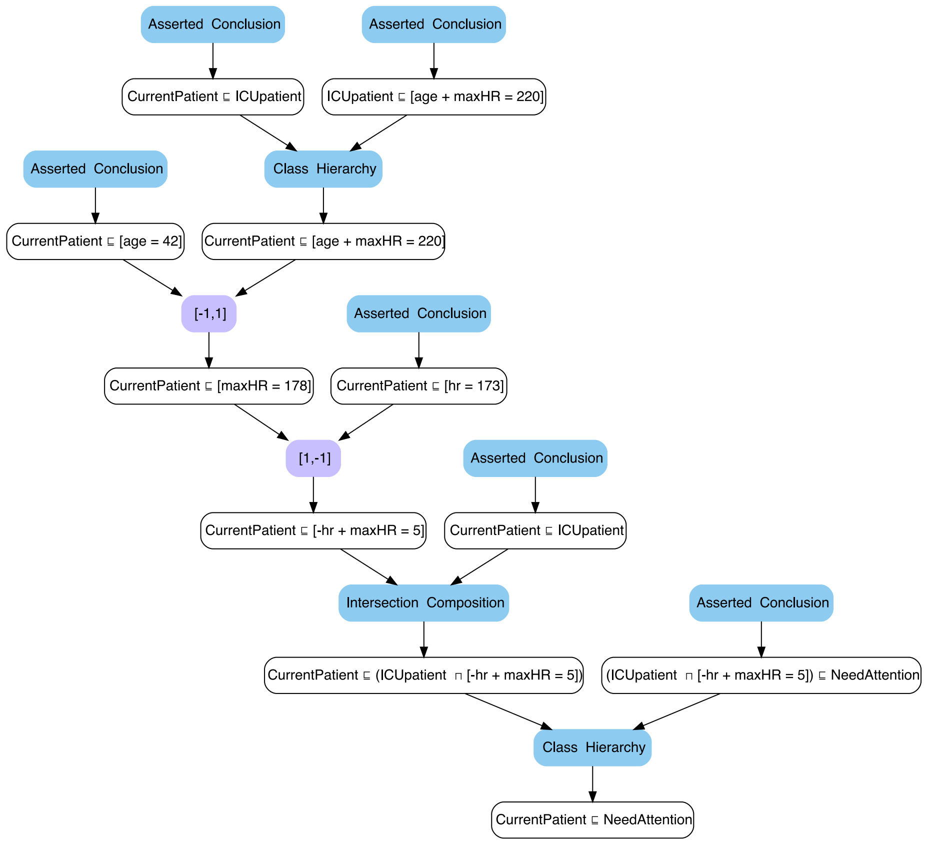

Fig. 1. Proof for $\mathsf{CurrentPatient} \sqsubseteq \mathsf{NeedAttention}$

where the features are now viewed as variables in the linear equations and the label $[-1, 1]$ for this proof step indicates that the constraint $\mathsf{age} + \mathsf{maxHR} = 220$ is multiplied by 1 and $\mathsf{age} = 42$ by -1 before they are added together, resulting in $\mathsf{maxHR} = 178$.

To integrate such a CD proof into the $\mathcal{EL}_\bot$ proof, we note that CD reasoning is always invoked in the context of a single concept C that is subsumed by the abstraction concepts of the involved constraints. We can now use this concept as the left-hand side of GCIs and thus transfer the CD proof (1) into a proof in $\mathcal{EL}_\bot[\mathcal{D}_{\mathbb{Q},lin}]$:

$$\frac{C \sqsubseteq [\mathsf{age} = 42] \qquad C \sqsubseteq [\mathsf{age} + \mathsf{maxHR} = 220]}{C \sqsubseteq [\mathsf{maxHR} = 178]}\ [-1, 1] \tag{2}$$

Figure 1 shows a proof of the subsumption $\mathsf{CurrentPatient} \sqsubseteq \mathsf{NeedAttention}$, where we have added information on the status of our current patient using the GCIs $\mathsf{CurrentPatient} \sqsubseteq [\mathsf{age} = 42]$, $\mathsf{CurrentPatient} \sqsubseteq [\mathsf{hr} = 173]$.

Further results

We have implemented the iterative reasoning approach for $\mathcal{EL}_\bot[\mathcal{D}]$ described above as well as inference procedures for the concrete domains $\mathcal{D}_{\mathbb{Q},lin}$ and $\mathcal{D}_{\mathbb{Q},diff}$. In addition, we have implemented proof extraction approaches for these concrete domains and have integrated them into proof extraction approaches for $\mathcal{EL}_\bot$, as sketched above. The proof shown in Figure 1 was automatically generated by our implementation.

We have also considered the integration of the CDs $\mathcal{D}_{\mathbb{Q},diff}$ and $\mathcal{D}_{\mathbb{Q},lin}$ into the more expressive DL $\mathcal{ALC}$. To this purpose, we developed a new calculus for subsumption w.r.t. $\mathcal{ALC}$ ontologies, which is inspired by the one in [19], but has a better worst-case complexity and can be used to produced proofs. This calculus can easily be extended to deal with concrete domain restrictions for p-admissible concrete domains, and our implementation can be used to generate proofs.

We have evaluated our implementations on several self-created benchmarks made specifically to challenge the CD reasoning and proof generation capabilities [5]. As results of the experiments, we could observe that the reasoning times were often dominated by the (incremental) calls to Elk, which shows that the CD reasoning did not add a big overhead, as long as the number of variables in the constraints did not become very large. In comparison to the incremental use of Elk as a black-box reasoner, the hypothetical "ideal" case of calling Elk only once on the final ontology (after all the new GCIs have been added) would not save a lot of time, which shows that the incremental nature of our approach also does not create a large overhead. The additional runtime for producing proofs was also mostly reasonable, staying within one order of magnitude of the pure reasoning time.

On the theoretical side, we have extended the approaches [2,3] for formally investigating the complexity of finding good (e.g. small) proofs to the case of DLs with CDs.

Conclusion and future work

We have demonstrated that it is feasible to support p-admissible concrete domains in DL reasoning algorithms, and even to produce integrated proofs to explain consequences in $\mathcal{EL}_\bot[\mathcal{D}]$ and $\mathcal{ALC}[\mathcal{D}]$. Although we have only considered subsumption reasoning here, the extension of our methods to data and reasoning about individuals, e.g. $\mathsf{fred} : \mathsf{ICUpatient} \sqcap [\mathsf{hr} = 90]$, encoded in so-called *ABoxes* [10], is straightforward. Our experiments show, however, that the size of the proofs can become quite large. Therefore, in future work we will study how to reduce the size of these proofs by identifying the most important parts and trying to condense the rest. On the theoretical side, we will also study the case of $\mathcal{ALC}[\mathcal{D}]$ with *admissible* CDs $\mathcal{D}$ [21], which would increase the expressivity on the CD side considerably.

Acknowledgments This work was supported by the DFG grant 389792660 as part of TRR 248 (`https://perspicuous-computing.science`).

References

1. Alrabbaa, C., Baader, F., Borgwardt, S., Dachselt, R., Koopmann, P., Méndez, J.: Evonne: Interactive proof visualization for description logics (system description). In: IJCAR (2022). https://doi.org/10.1007/978-3-031-10769-6_16
2. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding small proofs for description logic entailments: Theory and practice. In: LPAR (2020). https://doi.org/10.29007/nhpp
3. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding good proofs for description logic entailments using recursive quality measures. In: CADE (2021). https://doi.org/10.1007/978-3-030-79876-5_17
4. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Combining proofs for description logic and concrete domain reasoning. In: RuleML+RR (2023), to appear
5. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Combining Proofs for Description Logic and Concrete Domain Reasoning - RuleML+RR23 - Resources (2023). https://doi.org/10.5281/zenodo.8208780
6. Alrabbaa, C., Borgwardt, S., Hirsch, A., Knieriemen, N., Kovtunova, A., Rothermel, A.M., Wiehr, F.: In the head of the beholder: Comparing different proof representations. In: RuleML+RR (2022). https://doi.org/10.1007/978-3-031-21541-4_14
7. Armando, A., Castellini, C., Giunchiglia, E., Maratea, M.: A SAT-based decision procedure for the Boolean combination of difference constraints. In: SAT (2004). https://doi.org/10.1007/11527695_2
8. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: IJCAI (2005), http://ijcai.org/Proceedings/05/Papers/0372.pdf
9. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: IJCAI (1991), http://ijcai.org/Proceedings/91-1/Papers/070.pdf
10. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge Univ. Press (2017). https://doi.org/10.1017/9781139025355
11. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}^+$. In: KI (2007). https://doi.org/10.1007/978-3-540-74565-5_7
12. Baader, F., Rydval, J.: Using model theory to find decidable and tractable description logics with concrete domains. JAR **66**(3), 357–407 (2022). https://doi.org/10.1007/s10817-022-09626-2
13. Dutertre, B., de Moura, L.M.: A fast linear-arithmetic solver for DPLL(T). In: CAV (2006). https://doi.org/10.1007/11817963_11
14. Haarslev, V., Möller, R., Wessel, M.: The description logic $\mathcal{ALCNH}_{\mathcal{R}+}$ extended with concrete domains: A practically motivated approach. In: IJCAR (2001). https://doi.org/10.1007/3-540-45744-5_4
15. Horridge, M.: Justification Based Explanation in Ontologies. Ph.D. thesis (2011), https://www.research.manchester.ac.uk/portal/files/54511395/FULL_TEXT.PDF
16. Horridge, M., Parsia, B., Sattler, U.: Justification oriented proofs in OWL. In: ISWC (2010). https://doi.org/10.1007/978-3-642-17746-0_23
17. Kazakov, Y., Klinov, P., Stupnikov, A.: Towards reusable explanation services in Protege. In: DL (2017), https://ceur-ws.org/Vol-1879/paper31.pdf
18. Kazakov, Y., Krötzsch, M., Simancik, F.: The incredible ELK - From polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. JAR **53**(1), 1–61 (2014). https://doi.org/10.1007/s10817-013-9296-3

19. Koopmann, P., Schmidt, R.A.: Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In: FroCoS (2013). https://doi.org/10.1007/978-3-642-40885-4_7
20. Kroening, D., Strichman, O.: Decision Procedures - An Algorithmic Point of View, Second Edition. EATCS (2016). https://doi.org/10.1007/978-3-662-50497-0
21. Lutz, C.: The complexity of description logics with concrete domains. Ph.D. thesis (2002), https://nbn-resolving.org/urn:nbn:de:hbz:82-opus-3032
22. Lutz, C.: Description logics with concrete domains - A survey. In: Adv. in Modal Logic 4 (2002), http://www.aiml.net/volumes/volume4/Lutz.ps
23. Lutz, C.: NExpTime-complete description logics with concrete domains. ACM TOCL **5**(4), 669–705 (2004). https://doi.org/10.1145/1024922.1024925
24. Pan, J.Z., Horrocks, I.: Reasoning in the $\mathcal{SHOQ}(\mathrm{D_n})$ description logic. In: DL (2002), https://ceur-ws.org/Vol-53/Pan-Horrocks-shoqdn-2002.ps
25. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI (2003), http://ijcai.org/Proceedings/03/Papers/053.pdf
26. Turner, P.R.: Gauss elimination: Workhorse of linear algebra (1995), https://apps.dtic.mil/sti/pdfs/ADA313547.pdf, NAWCADPAX-96-194-TR