Deductive Module Extraction for Expressive Description Logics (Extended Abstract)

Patrick Koopmann[0000−0001−5999−2583]1 and Jieying Chen[0000−0002−2497−645X]2

[1] Inst. for Theoretical Computer Science, Technische Universität Dresden, Germany
[2] SIRIUS Centre, Department of Informatics, University of Oslo, Norway

In description logics (DLs), a *module* describes a subset of an ontology that preserves certain properties w.r.t. a given signature of concept and role names. Originally motivated by ontology reuse, ontology modularity has been widely used in different areas, such as in debugging or to improve reasoning. In this paper, we focus on applications in *ontology analysis* and *ontology reuse*, and consider a module notion based on *deductive inseparability* [6,12], also known as *concept inseparability* [9]. We call these modules *deductive modules*, but they have also been investigated under different names, such as basic modules [2] and subsumption modules [3].

Definition 1. *Let $\mathcal{L}$ be a DL, $\mathcal{O}$ an ontology and Σ a signature. Then, a subset $\mathcal{M} \subseteq \mathcal{O}$ is a* deductive $\langle \mathcal{L}, \Sigma \rangle$-module *of $\mathcal{O}$ w.r.t. Σ iff for every $\mathcal{L}$-axiom α with signature $sig(\alpha) \subseteq \Sigma$, we have that $\mathcal{O} \models \alpha$ iff $\mathcal{M} \models \alpha$.*

For deductive modules, not only the DL in which the ontology is formulated is relevant, but also the DL of the entailments we want to preserve. We focus on the DLs $\mathcal{ALC}$ and $\mathcal{ALCH}$ for the ontology and the entailments, as well as extensions with universal roles .

Different applications motivate different additional properties that a module should have. In *ontology analysis*, an ontology engineer might want to use modules to exhibit what an ontology states about some names of interest. For this, seeing as few axioms as necessary is usually desirable. *Subset-minimality* is thus a useful requirement for this use case. Rather than exhibiting what is stated, he might also want to exhibit *where* information is stated, and see all axioms that contribute to entailments in the selected signature. We cover this requirement under the notion of a *complete deductive module*. Usually, the module will use more names than are specified in the provided signature Σ. The ontology engineer might thus be interested to know *why* the axioms belong to a module, and how they contribute to entailments in the signature. For standard reasoning services, the necessity of explaining inferences has long been understood and implemented under the service of *justification* [1]. Our approach for computing deductive modules computes a so-called *annotated interpolant*, which shows the entailments in the selected signature, parts of which are annotated with axioms from the ontology that contribute to these entailments, and can thus be seen as an explanation of that module.

Another application of modules is for *ontology reuse*. Here, the ontology engineer wants to reuse a part of the ontology in another context in which only a subset of the signature is relevant, and thus a module is sufficient. Here, being complete and explainable is not as relevant, but *robustness* properties gain importance: specifically, it is not only important that all entailments in the signature are covered by the module, but also that entailments are preserved when further axioms are added: the module should be *replaceable* with the original ontology and still preserve the same entailments in the signature of interest.

Based on these motivations, we consider the following requirements: i) preserving entailments for a given DL, ii) being *subset-minimal*, iii) being *complete*, and iv) being *robust under replacements*. Since minimality is affected by whether we want to be complete or robust, we present different methods for different requirements. While most module notions investigated in the literature cover the above requirements, they often concern stronger notions of modules, which can make the problem of optimal module extraction hard. For instance, for the notion of *semantic modules*, already for the light-weight DL $\mathcal{EL}$ it is undecidable whether a given subset is a module for a given signature [8]. For this reason, most practical tools for module extraction either only compute approximations of minimal modules, or require restrictions on the input ontologies. In contrast, deductive modules in the expressive DL $\mathcal{ALC}$ are known to be decidable in 2ExpTime [5]. Furthermore, our experiments show that deductive modules are often substantially smaller than modules computed by existing methods.

Our methods make use of a method for *uniform interpolation* presented in [11] and adapted to our use case. Uniform interpolation computes a set of axioms that covers exactly the entailments the module has to preserve, the *uniform interpolant*. The basic idea underlying our approach is to track the inferences performed when computing the uniform interpolant from the original ontology. For this, the axioms in the ontology are first annotated using fresh concept names which we call *labels*. We then compute a uniform interpolant for the signature of interest extended by set of labels. The result is the annotated uniform interpolant, in which only a subset of the labels is still present. This way we can link each entailment presented in the uniform interpolant to the axioms in the input ontology from which they were derived, and thus construct a module for the signature of interest. To be able to construct modules that are robust under replacement, we extended the method from [11] to support universal roles, which also makes uniform interpolation faster in practice. A deeper modification of the uniform interpolation method is necessary to support role hierarchies. To compute subset-minimal modules, we use an algorithm that repeatedly computes and compares uniform interpolants of different subsets of the ontology. The annotated uniform interpolants makes it possible to compute and compare these uniform interpolants in short time.

We implemented a prototype of our approach which we evaluated and compared with existing methods ($\top\bot^*$-modules [7] computed using the OWL-API and modules computed by AMEX [4]), for which we focused on small signature sizes. For some signatures, it can become challenging to compute modules of minimal size. For those cases, our implementation allows for a flexible way to compute approximate modules: the more time given, the smaller the modules, and we know if the computed module is minimal. Our results indicate that in most cases, an approximation is not necessary, and that the modules computed by our method are significantly smaller than those computed with existing tools.

The full paper will be published in the proceedings of the International Joint Conference on Artificial Intelligence—Pacific Rim Conference on Artificial Intelligence (IJCAI-PRICAI-2020) [10].

Acknowledgement

Patrick Koopmann is partially supported by the DFG grant 389793660 as part of TRR 248 (`https://perspicuous-computing.science`). Jieying Chen is supported by the SIRIUS centre, which is funded by the Norwegian Research Council, project number 237898, and is co-funded by its partner companies.

References

1. Baader, F., Peñaloza, R.: Axiom pinpointing in general tableaux. In: Proceedings of TABLEAUX'07. pp. 11–27. Springer (2007)
2. Chen, J., Ludwig, M., Ma, Y., Walther, D.: Zooming in on ontologies: Minimal modules and best excerpts. In: Proceedings of ISWC'17. pp. 173–189. Springer (2017)
3. Chen, J., Ludwig, M., Walther, D.: On computing minimal $\mathcal{EL}$-subsumption modules. In: Proceedings of the Joint Ontology Workshops 2016 Episode 2: The French Summer of Ontology co-located with the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016), Annecy, France, July 6-9, 2016. CEUR Workshop Proceedings, vol. 1660. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1660/womocoe-paper6.pdf`
4. Gatens, W., Konev, B., Wolter, F.: Lower and upper approximations for depleting modules of description logic ontologies. In: Proceedings of ECAI'14. pp. 345–350. IOS Press (2014)
5. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proceedings of KR'06. pp. 187–197. AAAI Press (2006)
6. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: extracting modules from ontologies. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007. pp. 717–726. ACM (2007). https://doi.org/10.1145/1242572.1242669, `https://doi.org/10.1145/1242572.1242669`
7. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. **31**, 273–318 (2008)

8. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artificial Intelligence **203**, 66–103 (2013)
9. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artif. Intell. **174**(15), 1093–1141 (2010)
10. Koopmann, P., Chen, J.: Deductive module extraction for expressive description logics. In: Proceedings of IJCAI-PRICAI-2020 (2020), to appear.
11. Koopmann, P., Schmidt, R.A.: Forgetting concept and role symbols in $\mathcal{ALCH}$-ontologies. In: Proceedings of LPAR'13. pp. 552–567. Springer (2013)
12. Romero, A.A., Kaminski, M., Grau, B.C., Horrocks, I.: Module extraction in expressive ontology languages via datalog reasoning. J. Artif. Intell. Res. **55**, 499–564 (2016). https://doi.org/10.1613/jair.4898, `https://doi.org/10.1613/jair.4898`