Closed- and Open-world Reasoning in DL-Lite for Cloud Infrastructure Security (Extended Abstract)[⋆⋆⋆]

Claudia Cauli[1], Magdalena Ortiz[2], and Nir Piterman[1]

[1] University of Gothenburg
[2] TU Wien

Complex cloud infrastructure is managed through configuration files that are compiled into atomic deployment instructions as part of a process known as Infrastructure as Code (IaC). Configuration files contain declarations for the resources to be created, their settings, and their connectivity. Unfortunately, the same features that make IaC a convenient and powerful deployment tool—reusability, modularity, and shareability—also threaten the security of the cloud. The vulnerabilities arising from such a practice are subtle and widespread and need to be detected early, at the level of configuration files, *before* potentially-vulnerable infrastructure is deployed. To this end, we research the application of knowledge representation formalisms to the modeling and reasoning of IaC files. In particular, description logics allow for a succinct and natural description of these configuration files, and the open-world assumption captures the distributed nature of the cloud, where a newly deployed portion of infrastructure could connect to pre-existing resources not necessarily owned by the same user and whose configuration is only partially known. In previous work, we used the expressive $\mathcal{ALCOIQ}$ to model and reason about AWS CloudFormation, Amazon Web Services proprietary Infrastructure as Code framework [6,5]. Here, we suggest a lightweight DL that is specifically tailored for cloud infrastructure.

Core-closed Knowledge Bases We devise an extension of DL-Lite$^{\mathcal{F}}$ that allows for combining a core part that is completely defined (closed-world) and interacts with a partially known environment (open-world). We introduce the so-called "*core*-closed" knowledge bases, which are DL-Lite$^{\mathcal{F}}$ KBs defined as the tuple $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}, \mathcal{S}, \mathcal{M} \rangle$, built from a standard KB $\langle \mathcal{T}, \mathcal{A} \rangle$ and a *core* $\langle \mathcal{S}, \mathcal{M} \rangle$. The set $\mathcal{S}$ contains DL-Lite$^{\mathcal{F}}$ axioms representing the core structural cloud specifications for each type of resource that can be deployed, and the set $\mathcal{M}$ contains positive concept and role assertions representing the core user configuration. Syntactically, $\mathcal{M}$ is similar to an ABox $\mathcal{A}$ but, differently from $\mathcal{A}$, it is assumed to be complete with respect to the specifications $\mathcal{S}$. As usual, $\langle \mathcal{T}, \mathcal{A} \rangle$ encodes the incomplete terminological and assertional knowledge that, in our setting, may refer to both the (closed) core and the surrounding (open) world. We consider various reasoning problems over core-closed KBs and study their combined and

[⋆] Full paper to appear in KR 2021.

[⋆⋆]

data complexity [12]. As per standard DL-Lite$^{\mathcal{F}}$ results [4], we show that satisfiability of core-closed KBs *(i)* can be reduced to consistency of the functionality axioms and of the axioms in the negative closure of $\mathcal{T}$ and $\mathcal{S}$, and *(ii)* it is FOL-reducible. We also show that when dropping the unique name assumption on individuals not in the core satisfiability of DL-Lite$^{\mathcal{F}}$ core-closed KBs with inequalities is AC^0 in data complexity and P-complete in combined complexity.

Verification of Security Properties In security, we seek query languages to express that mitigations to security threats *must* be present (vs. may be absent) and vulnerabilities *may* be present (vs. must be absent). Such a requirement calls for efficient decision procedures for *query satisfiability*, in addition to query entailment. To reason about mitigations and vulnerabilities, we introduce Must and May conjunctive queries and devise a simple logical language for the specification of such properties. Technically, properties that *must* hold are resolved via query entailment and properties that *may* hold are resolved via query satisfiability. Regarding query entailment, as a result of the tight correspondence between the standard and the core-closed setting w.r.t. canonical model construction and query reformulation, we show that answering conjunctive queries in core-closed DL-Lite$^{\mathcal{F}}$ KBs is *FOL*-reducible. Regarding query satisfiability, we show that computing whether a tuple $\boldsymbol{t}$ is a *sat-answer* of a given query can be solved in logarithmic space in the core portion of the KB. We define a query language that allows for Boolean combinations of Must/May queries. Such a Boolean combination is a query that connects nested union of conjunctive queries in the scope of a Must or a May operator. Intuitively, the reasoning needed for answering the nested queries (either through entailment or satisfiability) can be decoupled from the reasoning needed to answer the higher-level Boolean combination.

Many authors have advocated for combining open- and closed-world reasoning in DLs in a variety of ways, e.g., [1,3,7,8]; for example, via *closed predicates* [7]. Our combination of open- and closed-world reasoning was tailored specifically for our application domain, and it is not obvious whether it can be easily expressed using the usual closed predicates, due to the presence of terms that are closed over part of the domain but open on the rest. One of the major challenges of extending DLs with closed predicates relates to complexity: they could be simulated in expressive DLs with nominals ($\mathcal{ALCO}$ and beyond), but for such logics satisfiability is at least ExpTime-hard [2] and conjunctive query entailment 2ExpTime-hard [11]. Unfortunately, query answering with closed predicates is also intractable in data complexity or FOL rewritable only under special safety restrictions that make the presence of the closed predicates irrelevant [10,9].

In our implementation of closed-world reasoning, core-closed KBs resemble safe KBs and are FOL rewritable, but the partial closed-world assumption plays an important role, particularly in the query satisfiability problem that arises from the May queries. For future work, we are interested in including more complex knowledge in the Tbox while still keeping (data) complexity tractable. Complex role inclusions would be required to reason about dataflow, which is a central aspect of security. Non-monotone extensions would be needed to be considered in order to reason about permissions and access policies.

References

1. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. J. of Automated Reasoning **14**(1), 149–180 (1995). https://doi.org/10.1007/BF00883932, https://doi.org/10.1007/BF00883932
2. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
3. Borgwardt, S., Forkel, W.: Closed-world semantics for conjunctive queries with negation over *ELH*_\bot ontologies. In: JELIA. Lecture Notes in Computer Science, vol. 11468, pp. 371–386. Springer (2019)
4. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reason. **39**(3), 385–429 (2007)
5. Cauli, C., Li, M., Piterman, N., Tkachuk, O.: Pre-deployment security assessment for cloud services through semantic reasoning. In: Computer Aided Verification - 33rd International Conference, CAV 2021, Proceedings. Springer (2021), to appear.
6. CloudFORMAL: Prototype Implementation (2020), http://github.com/claudiacauli/CloudFORMAL, Last accessed on 2020-10-15
7. Franconi, E., Ibáñez-García, Y.A., Seylan, I.: Query answering with dboxes is hard. Electr. Notes Theor. Comput. Sci. **278**, 71–84 (2011). https://doi.org/10.1016/j.entcs.2011.10.007, https://doi.org/10.1016/j.entcs.2011.10.007
8. Gaggl, S.A., Rudolph, S., Schweizer, L.: Fixed-domain reasoning for description logics. In: Kaminka, G.A., Fox, M., Bouquet, P., Hüllermeier, E., Dignum, V., Dignum, F., van Harmelen, F. (eds.) Proc. of the 22nd Eur. Conf. on Artificial Intelligence (ECAI 2016). Frontiers in Artificial Intelligence and Applications, vol. 285, pp. 819–827. IOS Press (2016). https://doi.org/10.3233/978-1-61499-672-9-819, https://doi.org/10.3233/978-1-61499-672-9-819
9. Lutz, C., Seylan, I., Wolter, F.: Ontology-based data access with closed predicates is inherently intractable(sometimes). In: Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI'2013). pp. 1024–1030. IJCAI/AAAI (2013)
10. Lutz, C., Seylan, I., Wolter, F.: The data complexity of ontology-mediated queries with closed predicates. Logical Methods in Computer Science **15**(3) (2019). https://doi.org/10.23638/LMCS-15(3:23)2019, https://doi.org/10.23638/LMCS-15(3:23)2019
11. Ngo, N., Ortiz, M., Šimkus, M.: Closed predicates in description logics: Results on combined complexity. In: Proc. Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2016). pp. 237–246. AAAI Press (2016)
12. Vardi, M.Y.: The complexity of relational query languages (extended abstract). In: STOC. pp. 137–146. ACM (1982)