

Expressivity of Planning with Horn Description Logic Ontologies (Extended Abstract)*

Stefan Borgwardt¹, Jörg Hoffmann², Alisa Kovtunova¹, Markus Krötzsch¹,
Bernhard Nebel³ and Marcel Steinmetz²

¹*Institute of Theoretical Computer Science, Technische Universität Dresden, 01062 Dresden, Germany*

²*Saarland University, Saarland Informatics Campus, Germany*


³*Faculty of Engineering, University of Freiburg, Germany*


Keywords

planning, expressive DLs, compilation scheme, experiments, benchmarks

AI planning is a well-investigated framework for describing the evolution of system states through actions [2]. Usually, a planning task describes an initial state of the agent’s environment, a set of actions that can affect that environment, and a goal formula that is to be satisfied. In order to reach the goal, actions can be applied whenever their preconditions are satisfied in the current state expressed as a finite set of facts. A *plan* is a sequence of (ground) actions that transform the initial state into a state satisfying the goal. First-order (FO) formulas in the preconditions and the goal are evaluated using a closed-domain, closed-world semantics, i.e. the domain is fixed a priori and facts that are not contained in a state are assumed to be false.


In contrast to preconditions which need to hold locally, the system can require to have global *state constraints*, constraints that should hold at every state. Knowledge representation formalisms are a natural way to introduce global constraints on permissible states; however, they usually interpret FO formulas over arbitrary models, instead of just one model with a fixed domain. Moreover, early work on open-world, open-domain formalisms for constraining possible states quickly noticed the so-called ramification problem [3], i.e. an action that makes a fact true also has to take care of satisfying the state constraints, possibly requiring to add or remove other facts, which may also involve new objects. One approach to deal with this problem is to not view states as finite interpretations, but as sets of formulas interpreted in an open-world fashion. In this way, implicit knowledge about the world added by the state constraints does not have to be made explicit in the states themselves. This means that the semantics of actions must be changed in such a way that action preconditions are also evaluated under open-world semantics, while action effects do not directly modify a single interpretation, but instead operate on a set of formulas.

 DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel


 stefan.borgwardt@tu-dresden.de (S. Borgwardt); hoffmann@cs.uni-saarland.de (J. Hoffmann);


alisa.kovtunova@tu-dresden.de (A. Kovtunova); markus.kroetzsch@tu-dresden.de (M. Krötzsch);

nebel@informatik.uni-freiburg.de (B. Nebel); steinmetz@cs.uni-saarland.de (M. Steinmetz)

 0000-0003-0924-8478 (S. Borgwardt); 0000-0003-1590-5876 (J. Hoffmann); 0000-0001-9936-0943 (A. Kovtunova);

0000-0002-9172-2601 (M. Krötzsch); 0000-0002-6833-6323 (B. Nebel)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

* This is an abstract of a paper presented at AAAI-22; technical report is available online [1]

DL-Lite Explicit-Input Knowledge and Action Bases (eKABs) [4] is an example of such an extended planning formalism. It was proposed to combine classical planning with state constraints and high-level concepts formulated in a DL-Lite *background ontology* [5], while allowing an open domain and interpreting action preconditions (specified using unions of conjunctive queries (CQs)) under open-world semantics. Due to FO rewritability of CQs in DL-Lite, eKABs can be translated into the classical planning language PDDL. In theory, this allows the use of off-the-shelf planning systems operating under closed-world semantics. Later, this compilation was further optimized to enable practical planning with DL-Lite background ontologies [6] by using PDDL *derived predicates* [7, 8]. Included in the PDDL 2.1 standard, derived predicates are special predicates not affected by any actions. Instead, the predicate’s extension is derived by a set of rules with the predicate in the head, which are re-evaluated at every new system state.

This paper extends the expressivity of state constraints in eKABs from the light-weight DL-Lite to more powerful (Horn) description logics (DLs) [9]. Our contribution is manifold.

A generic compilation scheme for any Datalog⁻-rewritable DL and queries. Informally, a compilation scheme [10, 11] is a mapping from eKAB planning tasks (over a specific DL) to PDDL tasks such that there exists a plan for an eKAB task iff there exists a plan for its PDDL translation. If in the mapped PDDL task the signature, domain, actions, and rules for derived predicates are bounded polynomially (exponentially) in the size of the eKAB description (i.e. a DL ontology and a set of actions), then the compilation scheme is *polynomial (exponential)*. For example, the compilation scheme for DL-Lite eKABs [4] is exponential.

Such compilation schemes can be used to investigate the relative expressivity of eKABs and PDDL. To be considered of the same expressivity, there should be a polynomial¹ compilation scheme that the length of the resulting plans grows at most polynomially, but ideally grows only by a constant amount. On the one hand, such a translation shows that the expressivity of eKABs with a specific DL is not higher than that of PDDL. On the other hand, although such eKABs could be considered syntactic sugar, they represent powerful tools that allow domain engineers to add open-world constraints to planning tasks.

Our generic compilation scheme for any DL and queries that enjoy *Datalog⁻-rewritability* essentially compiles an ontology and CQs in action preconditions into PDDL derived predicates of the same size. Using the property of Datalog⁻-rewritability in the compilation, we can employ many existing results from the DL literature [12, 13, 14] for enhanced AI planning. For example, this immediately implies that eKABs with Horn-*SHIQ* TBoxes have exponential compilations into PDDL with derived predicates (without negation) and for *ELH_⊥* and DL-Lite we even obtain polynomial compilations [13, 14]. The latter also holds for Horn-*SHOIQ* if all conditions are restricted to EIQs [12], i.e. unary atoms possibly prefixed by epistemic negation [15]. In general, our construction applies to any ontology language where CQs are Datalog⁻-rewritable, and to any specific TBoxes and queries that happen to be Datalog⁻-rewritable.

Polynomial compilation scheme for DL Horn-*ALCHOIQ*. To extend the previous compilability result even further, we provide a novel polynomial Datalog⁻-rewriting for queries in

¹The polynomial space requirement is attributed to the original work on compiling planning formalisms [10]: the result of a compilation should be concisely represented, i.e., in polynomial space.

the very expressive DL Horn- $\mathcal{ALCHOIQ}$ [16]. It is based on a query answering approach developed by [17] and encodes the relevant definitions from that paper into Datalog^{S, \neg} rules, which extend Datalog \neg by *set terms* that denote sets of objects. We then adapt a known polynomial translation to obtain a set of Datalog \neg rules [12].

Non-compilability for more expressive DLs. However, there are signs that our generic scheme does not work for the slightly more expressive DL Horn- \mathcal{SROIQ} [16] that does not enjoy Datalog \neg -rewritability. Let us consider two related problems. The *polynomial-step planning problem* is to decide whether a given task has a plan of polynomial length (for some given polynomial), and the *1-step planning problem* is the case where the polynomial is 1.

Theorem. *The polynomial-step planning problem for PDDL is EXPTIME-complete but the 1-step planning problem for Horn- \mathcal{SROIQ} eKABs is 2EXPTIME-complete.*

Although suggestive, this does not yet show that Horn- \mathcal{SROIQ} eKABs cannot be translated into PDDL using a polynomial compilation scheme (recall that we are only interested in the polynomial size of the resulting PDDL task, not the time it took to translate it). Nevertheless, we can show that such a compilation cannot exist (under a reasonable complexity-theoretic assumption) for Horn- \mathcal{SROIQ} as well as the non-Horn DLs \mathcal{SH} and \mathcal{ALCI} . To prove this, we follow the idea of a previous non-compilability result for PDDL with derived predicates into PDDL without derived predicates [11]. This more or less draws a line between the standardized ontology languages OWL 1,² which results in planning tasks of equal expressivity as PDDL, and OWL 2,³ where queries are strictly more expressive than PDDL.

Experimental evaluation. We conclude with an experimental evaluation that combines an existing practical implementation of an (exponential) Datalog-rewriting for Horn- \mathcal{SHIQ} [13] with our generic compilation scheme. While polynomial rewritings as above are nice in theory, they are often not practical due to a polynomial increase in the arity of predicates.

We encode Horn- \mathcal{SHIQ} eKAB tasks as ontology files accompanied by PDDL files whose syntax has been extended to allow conjunctive queries (CQ-PDDL). Our compiler generates a classical planning task in standard PDDL format, where the Datalog rewriting is generated by CLIPPER [13]. Our benchmark collection consists of 125 instances adapted from existing DL-Lite eKAB benchmarks (translated into our format) [4, 6], and 110 newly created instances using more expressive ontologies. Since Horn- \mathcal{SHIQ} is more expressive than DL-Lite, we created 3 new planning domains in which we make use of conjunctions, qualified existential restrictions occurring with negative polarity, and symmetric and transitive relations, all of which are not supported by DL-Lite [9].

On this benchmark set, our compilation approach showed superior runtimes compared to [4, 6], both for the compilation itself as well as for finding plans for the resulting PDDL tasks. In fact, we could solve all DL-Lite eKAB instances, which was not possible using the previous approaches. Our method also shows promising performance on the new benchmark instances, but there are not yet any other systems to compare it against.

²<http://www.w3.org/TR/owl-features/>

³<http://www.w3.org/TR/owl2-overview/>

Acknowledgments

This work is supported by DFG grant 389792660 as part of TRR 248 – CPEC, see <https://perspicuous-computing.science>).

References

- [1] S. Borgwardt, J. Hoffmann, A. Kovtunova, M. Krötzsch, B. Nebel, M. Steinmetz, Expressivity of Planning with Horn Description Logic Ontologies (Technical Report), 2022. URL: <https://arxiv.org/abs/2203.09361>. doi:10.48550/ARXIV.2203.09361.
- [2] M. Ghallab, D. Nau, P. Traverso, Automated Planning: Theory and Practice, Morgan Kaufmann, 2004.
- [3] M. L. Ginsberg, D. E. Smith, Reasoning about action I: A possible worlds approach, Artificial Intelligence 35 (1988) 165–195. doi:10.1016/0004-3702(88)90011-2.
- [4] D. Calvanese, M. Montali, F. Patrizi, M. Stawowy, Plan synthesis for knowledge and action bases, in: S. Kambhampati (Ed.), 25th Int. Joint Conf. on Artificial Intelligence (IJCAI), AAAI Press, 2016, pp. 1022–1029. URL: <https://www.ijcai.org/Abstract/16/149>.
- [5] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *dl-lite* family, Journal of Automated Reasoning 39 (2007) 385–429. doi:10.1007/s10817-007-9078-x.
- [6] S. Borgwardt, J. Hoffmann, A. Kovtunova, M. Steinmetz, Making DL-Lite planning practical, in: M. Bienvenu, G. Lakemeyer (Eds.), Proc. of the 18th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR’21), IJCAI, 2021, pp. 641–645. doi:10.24963/kr.2021/61.
- [7] M. Fox, D. Long, PDDL2.1: an extension to PDDL for expressing temporal planning domains, Journal of Artificial Intelligence Research 20 (2003) 61–124. doi:10.1613/jair.1129.
- [8] J. Hoffmann, S. Edelkamp, The deterministic part of IPC-4: An overview, Journal of Artificial Intelligence Research 24 (2005) 519–579. doi:10.1613/jair.1677.
- [9] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017. URL: <http://dltextbook.org/>. doi:10.1017/9781139025355.
- [10] B. Nebel, On the compilability and expressive power of propositional planning formalisms, Journal of Artificial Intelligence Research 12 (2000) 271–315. doi:10.1613/jair.735.
- [11] S. Thiébaux, J. Hoffmann, B. Nebel, In defense of PDDL axioms, Artificial Intelligence 168 (2005) 38–69. doi:10.1016/j.artint.2005.05.004.
- [12] M. Ortiz, S. Rudolph, M. Šimkus, Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR’10), AAAI Press, 2010, pp. 269–279. URL: <https://aaai.org/ocs/index.php/KR/KR2010/paper/view/1296>.
- [13] T. Eiter, M. Ortiz, M. Šimkus, T.-K. Tran, G. Xiao, Query rewriting for Horn-*SHIQ* plus rules, in: J. Hoffmann, B. Selman (Eds.), Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI’12), AAAI Press, 2012, pp. 726–733. URL: <http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4931>.

- [14] M. Bienvenu, M. Ortiz, Ontology-mediated query answering with data-tractable description logics, in: W. Faber, A. Paschke (Eds.), Reasoning Web. 11th Int. Summer School, volume 9203 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 218–307. doi:10.1007/978-3-319-21768-0_9.
- [15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, EQL-Lite: Effective first-order query processing in description logics, in: M. M. Veloso (Ed.), 20th Int. Joint Conf. on Artificial Intelligence (IJCAI), 2007, pp. 274–279. URL: <https://www.ijcai.org/Abstract/07/042>.
- [16] M. Ortiz, S. Rudolph, M. Šimkus, Query answering in the horn fragments of the description logics *SHOIQ* and *SRQIQ*, in: T. Walsh (Ed.), Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11), AAAI Press, 2011, pp. 1039–1044. doi:10.5591/978-1-57735-516-8/IJCAI11-178.
- [17] D. Carral, I. Dragoste, M. Krötzsch, The combined approach to query answering in horn-*ALCHOIQ*, in: M. Thielscher, F. Toni, F. Wolter (Eds.), Proc. of the 16th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'18), AAAI Press, 2018, pp. 339–348. URL: <https://aaai.org/ocs/index.php/KR/KR18/paper/view/18076>.