

Practical Planning with Ontologies

Duy Nhu

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

Abstract

Integrating AI planning with ontology reasoning offers a promising approach to developing goal-driven planning systems that benefit from static domain knowledge. While AI planning guides a system via action sequences under closed-world assumptions, and ontology reasoning provides expressive static constraints under open-world semantics, combining the two introduces significant theoretical and practical challenges. This research explores novel frameworks for planning with ontologies, with a strong focus on description logics and classical planning. We aim to analyse the computational complexity and expressivity of such integrated systems, design compilations into existing planning formalisms, and develop practical algorithms where compilation is not possible. Building upon recent advancements in integrated formalism, known as eKABs planning, our latest work provides support for coherence update semantics, enabling planning with additional logical semantics requirements to facilitate the modelling process in practice. The project ultimately aims to bridge the gap between the theoretical richness of ontology reasoning and the practicality of planning algorithms, contributing foundational tools to both theory and real-world applications.

Keywords

AI planning, ontology reasoning, description logics, ontology planning

1. Introduction and Problem Statement

Artificial Intelligence (AI) planning and reasoning with ontology languages (ontology reasoning) are two extensively studied research fields in symbolic AI and *Knowledge Representation and Reasoning* (KR). While AI planning traverses a system's internal states via a sequence of predefined actions to reach a desired goal, ontologies are used to express *static constraints* during the system's operations. In this research, we consider the intersection of the two formalisms, which integrates current state-of-the-art planning algorithms with reasoning over static data models, yielding great potential for applications in real-world scenarios such as business process definitions, knowledge graphs, relational databases, and operational planning/scheduling models [1, 2]. Early research in KR revealed that the integration not only introduces more complexity to the system but also raises concerns about maintainability and interpretability [3, 4]. These results caught the interest of both the KR and planning communities and eventually encouraged research on studying the complexity of combining the two fields, as well as introducing new formalisms over the years [5, 6, 7, 8, 9].

Until now, theoretical aspects have been the focal point of research on integrating AI planning and ontology reasoning. For instance, [10, 9], and [11] investigated the computational complexity of verification and synthesis. A common problem is that studies in reasoning often lack practical algorithms while providing only scientific prototypes. Contrarily, the planning community centralises on developing real-world algorithms and only deploys background knowledge on demand [12, 13]. Additionally, AI planning also encompasses *knowledge engineering* (KE), which focuses on developing, validating and maintaining complex planning tasks (*domain models*) and their surrounding tools [14, 15]. In this project, we aim not only to analyse the relative expressivity and complexity of distinct combinations between planning and reasoning, but also to develop the compilation and practical algorithms for planning with ontologies.

It's necessary to clarify overlapping terms used in both fields to avoid confusion. We refer to *reasoning* as static ontology reasoning in KR, *domain* as the description of a single planning or reasoning problem,

Doctoral Consortium of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR 2025 DC), November 11-17, 2025, Melbourne, Australia

✉ hoang_duy.nhu@tu-dresden.de (D. Nhu)

ORCID: 0009-0003-2220-3263 (D. Nhu)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

and *state* as a finite set of grounded *first-order* (FO) atoms (*facts*). We differentiate between two semantics of FO-formulas, namely *open-world* and *closed-world* assumptions. Specifically, in AI planning, a state under the closed-world assumption fully describes a single FO interpretation over a fixed set of objects. Contrarily, ontology reasoning employs open-world semantics where infinitely many interpretations are considered over arbitrary sets of objects, as long as they are consistent with the observed state.

Ontology Reasoning. Generally, first-order logic (FOL) allows for modelling complex constraints about unknown objects/facts under open-world semantics. However, this leads to reasoning in FOL being undecidable and thus impractical for applications. To overcome this obstacle, we consider restricted subsets of FOL, which include *description logics* (DLs), *Datalog*, and *existential rules* [16]. This research focuses on DLs, which offer decidable open-world reasoning [17]. In particular, DLs describe a domain of interest through concepts and roles, which express sets of objects and binary relations between them. Starting from atomic concepts/roles, complex concepts/roles are represented by applying the appropriate constructs in the corresponding fragment of DLs (e.g. $\exists$, $\neg$, etc.). Then, a *knowledge base* (KB) or *ontology* contains extensional knowledge (ABox) (i.e. factual assertions of individuals/facts) and intentional knowledge (TBox) (i.e. axioms of classes of individuals) from the chosen fragment.

Example 1 demonstrates how an ontology expresses knowledge:

Example 1. *Axioms and facts in Blocks world ontology:*

$$\begin{aligned} \text{on_block} &\sqsubseteq \text{on}, \exists \text{on_block}^- \sqsubseteq \text{Block}, \text{funct on_block}, \\ \text{on_table} &\sqsubseteq \text{on}, \exists \text{on_table}^- \sqsubseteq \text{Table}, \text{Block} \sqsubseteq \neg \text{Table}, \\ \text{Block} &\equiv \exists \text{on}, \exists \text{on_block}^- \sqsubseteq \text{Blocked}, \\ \exists \text{on_block} &\sqsubseteq \neg \exists \text{on_table}, \text{on_block}(b_1, b_2), \text{on_table}(b_3, t) \end{aligned}$$

This ontology expresses that b_2 is blocked ($\text{Blocked}(b_2)$) since b_1 is on b_2 ($\text{on_block}(b_1, b_2)$) and every block that has another block on top is blocked ($\exists \text{on_block}^- \sqsubseteq \text{Blocked}$). Additionally, $\text{on_block}(b_1, b_3)$ cannot hold, since the on_block relation is functional (funct on_block).

In this project, we are interested in two main reasoning problems. The first one is to determine whether an ontology is *consistent*, meaning it contains no contradictions. The second problem is to decide if a query formula is entailed by the ontology, meaning it is satisfied in every interpretation of the ontology. The following approaches address these problems: *rewriting* into closed-world reasoning problems [18], exhaustively deriving all consequences of an ontology using *consequence-based methods* [19], extracting abstract representations of interpretations via the so-called *tableau algorithm* [20], and type-based reasoning for objects appearing in interpretations using *automata-based approaches* [17].

AI Planning. Classical planning employs closed-world semantics for procedural system models specified in the FOL-based modelling language *planning domain definition language* (PDDL) [21] while focusing on computational problems in the context of controlling systems' states [22]. Such states are governed by actions, each of which comprises a FOL precondition on the state and a set of effects represented by first-order literals that hold after the action's execution. The goal is to find a plan, a sequence of actions that transforms the initial state into a state satisfying a goal formula. However, one major drawback of this formalism is that action effects ignore both implicit knowledge and consistency of the subsequent state w.r.t. the observed intentional knowledge.

We will mainly focus on classical planning, in which one of the best candidates for computing plans is *heuristic search* [23, 24]. The technique involves heuristic functions that estimate the costs of reaching the goal to guide the search. Among those available, delete-relaxation heuristics [25, 23] and state-abstraction-based heuristics established themselves as the most successful techniques for finding plans [26].

Besides classical planning, there are various planning formalisms with more expressive frameworks [27, 28]. The diversity within the field enables *compilability*, i.e. the translation of tasks across different

formalisms, which allows for preceding theoretical results to be applied without extensive additional studies.

Combined Formalism. In planning, a state is interpretable as an ABox, whereas a TBox can describe its background knowledge. Exploiting these similarities, the most recent promising formalism for combining classical planning with ontology reasoning is *explicit-input Knowledge and Action Bases* (eKABs) [10]. The formalism allows for planning tasks to be compiled into PDDL using query rewriting techniques while integrating planning with the description logic *DL-Lite* [29], in which states are interpreted with the open-world semantics w.r.t. a *background ontology* specifying intensional knowledge using *DL-Lite* axioms.

Example 2. Consider the eKAB action $\text{move}(x, y, z)$ that moves Block x from position y to z . Its precondition is $[\text{on}(x, y)] \wedge \neg[\text{Blocked}(x)] \wedge \neg[\text{Blocked}(z)]$, where the atoms in brackets are evaluated w.r.t. the ontology axioms (epistemic semantics). Its effects consist of

$$\begin{aligned} & ((), [\text{Block}(y)], \emptyset, \{\neg\text{on_block}(x, y)\}), \\ & ((), [\text{Table}(y)], \emptyset, \{\neg\text{on_table}(x, y)\}), \\ & ((), [\text{Block}(z)], \{\text{on_block}(x, z)\}, \emptyset), \\ & ((), [\text{Table}(z)], \{\text{on_table}(x, z)\}, \emptyset), \end{aligned}$$

which remove $\text{on_block}(x, y)$ when y is entailed to be a Block, add $\text{on_table}(x, z)$ if z is a Table, and so on. Effectively, it removes $\text{on}(x, y)$ and adds $\text{on}(x, z)$.

For example, the action is applicable for the substitution $x \mapsto b_1, y \mapsto b_2, z \mapsto b_3$, since on_block is included in on and neither $\text{Blocked}(b_1)$ nor $\text{Blocked}(b_3)$ are entailed. Then, it would remove $\text{on_block}(b_1, b_2)$ and insert $\text{on_block}(b_1, b_3)$, as $\text{Block}(b_2)$ and $\text{Block}(b_3)$ are entailed.

2. Research Plan

A duration of 3 years is anticipated for the project's fruition.

Research Goals. As mentioned previously, our primary goal is the investigation of distinct possibilities for integrating AI planning and reasoning, with DLs and classical planning as focal points. Under this context, we will establish the computational complexity of the concerned reasoning problems, which we will reinforce with implemented experiments in planning. Starting with eKABs, we will examine existing proposals on implementing reasoning in planning and strive to support more expressive logical frameworks. The study will include the evaluation of their relative expressivity, efficiency, and usefulness for modelling real-world applications. Even though compilation approaches can easily adopt preceding results and exploit state-of-the-art planning systems, the technique is not universally applicable due to the differences in expressivity of the framework in observation. In such cases, we will instead design dedicated, integrated algorithms and study their properties. Summarily, the following are the main objectives of our research programme:

- (O_1) Complexity and expressivity analysis of combined planning and reasoning formalisms;
- (O_2) Where possible, designing efficient compilations of the formalisms in (O_1);
- (O_3) Development of integrated algorithms for more expressive planning over a background ontology (and vice versa).

3. Progress and Related Work

In our latest work [30], we focus on the low-complexity DL framework $DL-Lite_{core}^{(HF)}$ [29], which we denote as *DL-Lite* for simplicity, and extend upon eKAB planning formalism [10, 9] to provide support for the *coherence update semantics* [31] in PDDL. To achieve this, we allow for *multiple updates* (i.e. changes between state transitions) with the semantics, as its original definition supports only a single state update. The semantics helps tackle the problem demonstrated by the example below:

Example 3. *The effect of the action $move(b_1, b_2, b_3)$ in Example 2 is to add $on_block(b_1, b_3)$. However, this would make the state inconsistent, as argued in Example 1.*

We could remove $on(b_1, b_2)$ to obtain a consistent state. However, since this fact is not explicitly present in the state (ABox), this operation doesn't have any effect, and $[on(b_1, b_2)]$ would hold due to $on_block(b_1, b_2)$.

Even if we explicitly remove $on_block(b_1, b_2)$, we would lose that b_2 is a block, i.e. we should add $Block(b_2)$.

Example 3 illustrates that actions can cause three types of implicit effects: removing a fact requires (i) removing all stronger facts and (ii) adding previously implied facts to avoid losing information, whereas adding a fact requires (iii) removing any conflicting facts to ensure consistency.

Since the effects of the coherence update semantics coincide with the implicit effects listed in (i), (ii), and (iii), we define a new semantics for eKAB that satisfies these requirements by applying the coherence update semantics to all actions in a planning problem. The resulting planning task retains favourable behaviours of the epistemic eKAB semantics for action conditions and of the coherence update semantics for single-step updates between the states for *DL-Lite*. In particular, it is possible to rewrite all operations into Datalog⁺, and therefore into classical planning with PDDL derived predicates [21, 32], in polynomial time.

We believe that our work's main contribution to practical use lies in the ease of modelling for planning problems. In particular, explicitly dealing with the consequences of the background knowledge in every action can be avoided by simply encoding it in the ontology.

Admittedly, the coherence update semantics may not always be the most appropriate choice. For instance, actions that completely remove a fact and its consequences from a state might fail under the coherence update semantics due to (ii). Hence, we will also study combined semantics that allow switching, e.g. between eKAB and the new semantics in future works via a special action that ignores the implicit effects of the coherence update semantics. Nevertheless, such action might break the state's consistency, requiring additional repair mechanisms afterwards. Furthermore, we plan to provide support for conjunction in *DL-Lite*. However, it is unclear whether a suitable update semantics can be defined, since removing a conjunction involves a nondeterministic choice of which conjunct should be removed. Lastly, in the case of large ontologies, it would be beneficial to use DL reasoners directly instead of rewriting techniques, as they are optimised for dealing with such ontologies and can be integrated with the planning system in a black-box manner.

3.1. Closely Related Work

The eKAB formalism was optimised and extended to support all Datalog⁺-rewritable Horn DLs, via a compilation into PDDL with derived predicates [33, 9]. A similar approach uses a black-box, justification-based algorithm to compile an ontology-mediated planning problem into classical planning, even for non-Horn DLs [34].

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in grant 540204715 and by the Swiss National Science Foundation (SNSF) as part of the project "Practical Planning with Ontologies" (PPO).

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

- [1] M. Dumas, F. Fournier, L. Limonad, A. Marrella, M. Montali, J.-R. Rehse, R. Accorsi, D. Calvanese, G. De Giacomo, D. Fahland, et al., Ai-augmented business process management systems: a research manifesto, *ACM Transactions on Management Information Systems* 14 (2023) 1–19.
- [2] J. Hoffmann, I. Weber, F. M. Kraft, Sap speaks pddl: Exploiting a software-engineering model for planning in business process management, *Journal of Artificial Intelligence Research* 44 (2012) 587–632.
- [3] T. Eiter, G. Gottlob, The complexity of logic-based abduction, *Journal of the ACM (JACM)* 42 (1995) 3–42.
- [4] R. Reiter, A theory of diagnosis from first principles, *Artificial intelligence* 32 (1987) 57–95.
- [5] F. Baader, C. Lutz, M. Milicic, U. Sattler, F. Wolter, Integrating description logics and action formalisms: First results., in: *AAAI*, volume 5, 2005, pp. 572–577.
- [6] M. Cioffi, S. Thompson, Planning with the semantic web by fusing ontologies and planning domain definitions, in: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Springer, 2006, pp. 289–302.
- [7] B. B. Hariri, D. Calvanese, M. Montali, G. De Giacomo, R. De Masellis, P. Felli, Description logic knowledge and action bases, *Journal of Artificial Intelligence Research* 46 (2013) 651–686.
- [8] D. Calvanese, G. De Giacomo, M. Soutchanski, On the undecidability of the situation calculus extended with description logic ontologies, in: *Proceedings of the 24th International Joint Conference on Artificial Intelligence: Buenos Aires, Argentina, 25-31 July 2015*, AAAI Press, 2015, pp. 2840–2846.
- [9] S. Borgwardt, J. Hoffmann, A. Kovtunova, M. Krötzsch, B. Nebel, M. Steinmetz, Expressivity of planning with horn description logic ontologies, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022, pp. 5503–5511.
- [10] D. Calvanese, M. Montali, F. Patrizi, M. Stawowy, et al., Plan synthesis for knowledge and action bases, in: *IJCAI*, AAAI Press, 2016, pp. 1022–1029.
- [11] T. John, P. Koopmann, Towards ontology-mediated planning with owl dl ontologies (extended version), *arXiv preprint arXiv:2308.08200* (2023).
- [12] C. Dornhege, P. Eyerich, T. Keller, S. Trüg, M. Brenner, B. Nebel, Semantic attachments for domain-independent planning systems, in: *Towards Service Robots for Everyday Environments: Recent Advances in Designing Service Robots for Complex Tasks in Everyday Environments*, Springer, 2012, pp. 99–115.
- [13] P. Haslum, F. Ivankovic, M. Ramirez, D. Gordon, S. Thiébaux, V. Shivashankar, D. S. Nau, Extending classical planning with state constraints: Heuristics and search for optimal planning, *Journal of Artificial Intelligence Research* 62 (2018) 373–431.
- [14] J. Barreiro, M. Boyce, M. Do, J. Frank, M. Iatauro, T. Kichkaylo, P. Morris, J. Ong, E. Remolina, T. Smith, et al., Europa: A platform for ai planning, scheduling, constraint programming, and optimization, 4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS) (2012) 6–7.
- [15] T. Plch, M. Chomut, C. Brom, R. Barták, Inspect, edit and debug pddl documents: Simply and efficiently with pddl studio, *System Demonstrations and Exhibits at ICAPS* (2012) 15–18.
- [16] M. Calautti, G. Gottlob, A. Pieris, Chase termination for guarded existential rules, in: *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2015, pp. 91–103.
- [17] F. Baader, I. Horrocks, C. Lutz, U. Sattler, *An introduction to description logic*, Cambridge University Press, 2017.

- [18] 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.
- [19] F. Baader, Pushing the el envelope (2005).
- [20] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible sroiq., *Kr* 6 (2006) 57–67.
- [21] M. Fox, D. Long, Pddl2. 1: An extension to pddl for expressing temporal planning domains, *Journal of artificial intelligence research* 20 (2003) 61–124.
- [22] K. Erol, D. S. Nau, V. S. Subrahmanian, Complexity, decidability and undecidability results for domain-independent planning, *Artificial intelligence* 76 (1995) 75–88.
- [23] J. Hoffmann, B. Nebel, The ff planning system: Fast plan generation through heuristic search, *Journal of Artificial Intelligence Research* 14 (2001) 253–302.
- [24] M. Helmert, The fast downward planning system, *Journal of Artificial Intelligence Research* 26 (2006) 191–246.
- [25] M. Helmert, C. Domshlak, Landmarks, critical paths and abstractions: what’s the difference anyway?, in: *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 19, 2009, pp. 162–169.
- [26] J. Seipp, M. Helmert, Counterexample-guided cartesian abstraction refinement, in: *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 23, 2013, pp. 347–351.
- [27] L. P. Kaelbling, M. L. Littman, A. R. Cassandra, Planning and acting in partially observable stochastic domains, *Artificial intelligence* 101 (1998) 99–134.
- [28] M. Daniele, P. Traverso, M. Y. Vardi, Strong cyclic planning revisited, in: *European Conference on Planning*, Springer, 1999, pp. 35–48.
- [29] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyashev, The DL-Lite family and relations, *J. Artif. Intell. Res.* 36 (2009) 1–69. doi:10.1613/JAIR.2820.
- [30] S. Borgwardt, D. Nhu, G. Röger, Automated planning with ontologies under coherence update semantics, in: *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR 2025*, November 11–17, 2025, 2025. To appear.
- [31] G. De Giacomo, X. Oriol, R. Rosati, D. F. Savo, Instance-level update in DL-Lite ontologies through first-order rewriting, *J. Artif. Intell. Res.* 70 (2021) 1335–1371. URL: <https://doi.org/10.1613/jair.1.12414>. doi:10.1613/JAIR.1.12414.
- [32] J. Hoffmann, S. Edelkamp, The deterministic part of ipc-4: An overview, *Journal of Artificial Intelligence Research* 24 (2005) 519–579.
- [33] S. Borgwardt, J. Hoffmann, A. Kovtunova, M. Steinmetz, Making DL-Lite planning practical, in: M. Bienvenu, G. Lakemeyer, E. Erdem (Eds.), *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021*, Online event, November 3–12, 2021, 2021, pp. 641–645. URL: <https://doi.org/10.24963/kr.2021/61>. doi:10.24963/KR.2021/61.
- [34] T. John, P. Koopmann, Planning with OWL-DL ontologies, in: U. Endriss, F. S. Melo, K. Bach, A. J. B. Diz, J. M. Alonso-Moral, S. Barro, F. Heintz (Eds.), *ECAI 2024 - 27th European Conference on Artificial Intelligence*, 19–24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2024, pp. 4165–4172. URL: <https://doi.org/10.3233/FAIA240988>. doi:10.3233/FAIA240988.