Making DL-Lite Planning Practical: Supplementary Material

Benchmark Description

Our collection of benchmarks consists of a total of 205 instances from varied sources, including existing ones as well as ones that we created.

Original eKAB benchmarks: We considered the scalable eKAB benchmark *Robot* used in prior work [Calvanese et al., 2016, Stawowy, 2016]. In this domain, a robot is positioned on a grid without knowing its exact position. The goal of the robot is to reach a specific cell. The ontology describes relations between rows and columns, e. g. $\mathsf{BelowOf6} \sqsubseteq \mathsf{BelowOf7}$ and $\mathsf{AboveOf1} \sqsubseteq \neg\mathsf{BelowOf1}$. In the original work, the Fast Downward[1] planner (FD) reached a 30-minutes timeout on a 7×7-grid. However, for the experiments reported in this paper, FD using our pre-compilation managed to solve an instance of size 22×22 in less than 2 minutes.

TaskAssign was also presented in [Calvanese et al., 2016] as a running example. In our paper, we also used some of its axioms in examples. The goal of this benchmark is to hire two electronic engineers for a company, among other personnel. The ontology describes disjointness relations of different job positions. For our experiments, we scale up the original problem by increasing the number of already hired people as well as the number of objects in the world.

Own eKAB domains: We created a new eKAB domain *Cats*, inspired by the well-known classical planning benchmark *BT* (see [Petrick and Bacchus, 2002] for a description). In this domain, there is a set of packages connected by a functional binary predicate $\mathsf{contains}$ either to a Cat or to a Bomb. No Cat is a Bomb, both of them are Objects and only a $\mathsf{Package}$ can $\mathsf{contain}$ an Object. The actions $\mathsf{release}$ and dunk can be performed on a package to release a cat or disarm a bomb from the package. The task is to find a plan such that only disarmed objects remain in the packages.

In addition, a scalable eKAB domain called *Elevator* was developed inspired by *Miconic* [Koehler and Schuster, 2000], a classical planning benchmark which we also consider in our experiments. An elevator can move up and down between floors to serve passengers according to their origins and destinations. Here, the ontology axioms establish, inter alia, that $\mathsf{Boarded}$ and Served are subcategories of $\mathsf{Passenger}$ and

[1] `http://www.fast-downward.org/`

each Floor is connected to exactly one other Floor by the functional binary predicate next.

Web-service composition: We adapted already existing web-service composition benchmarks [Hoffmann et al., 2008] to the eKAB framework: *VTA*, *VTA-Roles*, and *TPSA*. In the original *VTA* task, a travel agency must book a travel itinerary including a Flight, a Hotel and a CarRental. The ontology distinguishes different types of requests, e. g. $\mathsf{FlightRequest} \sqsubseteq \mathsf{Flight}$ and $\mathsf{FlightRequest} \sqsubseteq \neg\mathsf{CarRentalRequest}$. The only modification we made is setting up an order for choosing new, unbound objects from the world. This little trick simplifies grounding not only for our pre-compilation, but also for internal grounding in FF and FD. Next, since the *VTA* axioms in the original planning task are relatively simple, for *VTA-Roles* we used the expressiveness of $DL\text{-}Lite$ in eKAB to create a more elaborate ontology with binary predicates that connect all requests like FlightRequest or AirportShuttle to the Trip they were made for.

In *TPSA*, when a Service is requested, a business process for preparing a contract, a hardware and a product activation must be developed by a planner. In this domain, we also incorporated binary predicates as well as an assignment order on the level of an ontology.

In all web-service domains, *TPSA*, *VTA*, and *VTA-Roles*, most of the actions assign predicates to new objects in the world. Therefore, by increasing the number of objects (even with introduced assignment orders), the grounding and pre-compilation sizes grow drastically. That fact together with an absence of complex conditions in the domain actions results in a decreased performance when using our pre-compilation.

All of the previously described benchmarks are created specifically in order to extend planning with background knowledge expressed by an ontology. Thus, the problem instances for *Robot*, *TaskAssign*, *Cats*, *Elevator*, *TPSA*, *VTA* and *VTA-Roles* have two representations: the eKAB syntax and its translation[2] into PDDL. The benchmarks in the following paragraph do not have any ontology as a part of the domain description.

PDDL: We included two benchmark domains from the planning literature with complex action pre- and effect conditions: *Assembly*, and *Miconic*. We made minor modifications, introducing additional actions to make some predicates non-static. This prevents the planners from simplifying the conditions in a simple pre-processing step. In *Assembly*, given available parts and tools, a planner must find a valid way to assemble a bracket by setting assemble- and remove-orders and then implement this plan with respect to committed resources. *Miconic* resembles *Elevator*, but it does not have any ontology and it considers also some categories of passengers like attendant or never alone; the domain assumes that passengers may have conflicts with each other and an access to the elevator can also be blocked.

Additionally, we created a showcase domain, *GridPlacement*, specifically designed to contain challenging DNF transformations. In contrast to *Robot*, here a robot starts from a known position of the grid, it has a fixed weight capacity and there are some

[2]see `https://github.com/stawo/ekabPlanner`

objects with given weights scattered on a grid. By pickup, drop and move, the robot must transport the objects to specific coordinates on the grid.

References

[Calvanese et al., 2016] Calvanese, D., Montali, M., Patrizi, F., and Stawowy, M. (2016). Plan synthesis for knowledge and action bases. In Kambhampati, S., editor, *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI'16)*, pages 1022–1029. AAAI Press.

[Hoffmann et al., 2008] Hoffmann, J., Weber, I., Scicluna, J., Kacmarek, T., and Ankolekar, A. (2008). Combining scalability and expressivity in the automatic composition of semantic web services. In *8th International Conference on Web Engineering (ICWE'08)*.

[Koehler and Schuster, 2000] Koehler, J. and Schuster, K. (2000). Elevator control as a planning problem. In Chien, S. A., Kambhampati, S., and Knoblock, C. A., editors, *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge, CO, USA, April 14-17, 2000*, pages 331–338. AAAI.

[Petrick and Bacchus, 2002] Petrick, R. P. A. and Bacchus, F. (2002). A knowledge-based approach to planning with incomplete information and sensing. In Ghallab, M., Hertzberg, J., and Traverso, P., editors, *Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems, April 23-27, 2002, Toulouse, France*, pages 212–222. AAAI.

[Stawowy, 2016] Stawowy, M. (2016). *Plan Synthesis in Explicit-input Knowledge and Action Bases*. PhD thesis, IMT School for Advanced Studies Lucca, Italy.