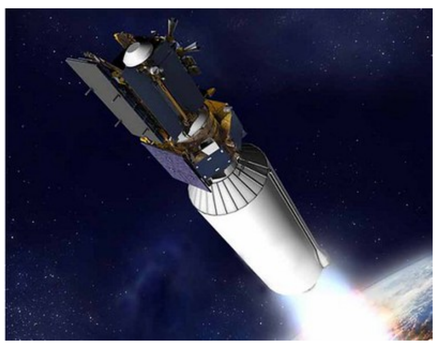
TECHNISCHE UNIVERSITÄT DRESDEN

CPEC CENTER FOR PERSPICUOUS COMPUTING

UNIVERSITÄT DES SAARLANDES

Making DL-Lite Planning Practical

Stefan Borgwardt Jörg Hoffmann Alisa Kovtunova Marcel Steinmetz



AI planning + static reasoning:

Making Explicit-input Knowledge and Action Bases (eKABs) More Practical

(Calvanese, Montali, Patrizi, and Stawowy 2016)

Example: Lift Control

Planning Domain Definition Language (PDDL)

```
(:objects pa pb ... fa fb ...)

(:init
  (passenger pa)
  (origin pa fe)
  (destin pa fa)
  ...
  (lift_at fa)
)

(:goal (not (exists (?x)
  (and
    (passenger ?x)
    (not (served ?x))
) ) ) )

(:action stop
  :parameters (?p ?f)
  :precondition (and (lift_at ?f) (passenger ?p))
  :effect (and
    (when
      (and (origin ?p ?f) (not (served ?p)))
      (boarded ?p)
    )
    (when
      (and (destin ?p ?f) (boarded ?p))
      (and (served ?p) (not (boarded ?p)))
) ) )

(:action move-up  ...
```

Fixed domain

Closed-world states

{passenger(pa), origin(pa,fe), ...}

First-order formulas (closed-world semantics)

$\neg\exists x.\, \mathtt{passenger}(x) \land \neg\mathtt{served}(x)$

$\mathtt{lift\_at}(f) \land \mathtt{passenger}(p)$

$\mathtt{origin}(p,f) \land \neg\mathtt{served}(p)$

Effect: add boarded(p) to the state

$\mathtt{destin}(p,f) \land \mathtt{boarded}(p)$

Effect: add served(p), remove boarded(p)

eKABs

(Calvanese, Montali, Patrizi, and Stawowy 2016)

Effects operate on open-world states (ABoxes)

Additional background ontology (*DL-Lite*$_{\mathcal{F}}$):

```
(:axioms
  (isA served passenger)
  (isA boarded passenger)
  (isA passenger (not floor))
  (isA (exists origin) passenger)
  (isA (exists destin) passenger)
  (isA (exists (inverse origin)) floor)
  ...
```

$\mathtt{served} \sqsubseteq \mathtt{passenger}$

$\mathtt{boarded} \sqsubseteq \mathtt{passenger}$

$\mathtt{passenger} \sqsubseteq \neg\mathtt{floor}$

$\exists\mathtt{origin} \sqsubseteq \mathtt{passenger}$

$\exists\mathtt{destin} \sqsubseteq \mathtt{passenger}$

$\exists\mathtt{origin}^- \sqsubseteq \mathtt{floor}$

Conditions are epistemic conjunctive queries (ECQs) (FO-formulas where the atoms are CQs with minimal-knowledge-operator):

```
(and (mko (lift_at ?f)) (mko (passenger ?p)))
```

(Calvanese, De Giacomo, Lembo, Lenzerini, and Rosati 2007)

Rewriting ECQs into PDDL

Use standard techniques to rewrite CQs into closed-world UCQs:

```
(mko (passenger ?p))  --ontology-->  (or
                                       (passenger ?p) (boarded ?p) (served ?p)
                                       (exists (?x) (origin ?p ?x))
                                       (exists (?x) (destin ?p ?x))
                                     )
```

Not suited for state-of-the-art planners:

- FO-formulas need to be pre-processed into ground DNF
- grounded UCQs are in DNF, but nested in other formulas

Our solution: Replace complex subformulas by new predicates

$$\Phi = (\mathtt{or}\ \Phi_1\ \Phi_2) \longmapsto P_\Phi$$

- (Nebel 2000): introduce new actions to re-evaluate P_Φ
- New approach: use derived predicates: $(\mathtt{:derived}\ P_\Phi\ (\mathtt{or}\ \Phi_1\ \Phi_2))$

Experiments

Existing eKAB benchmarks: Robot, TaskAssign

(Calvanese, Montali, Patrizi, and Stawowy 2016)

Adapted planning benchmarks: Cats, Elevator

Adapted web service composition benchmarks: TPSA, VTA(-Roles)

(Hoffmann, Weber, Scicluna, Kacmarek, and Ankolekar 2008)

Planning benchmarks with complex conditions: Assembly, Miconic

Artificial benchmark with huge CNFs: GridPlacement

		FF			FD		
Domain	#	O	Ne	DP	O	Ne	DP
Robot	20	20	1	20	4	1	20
TaskAssign	20	1	1	15	3	1	20
Cats	20	14	14	20	14	11	20
Elevator	20	12	0	20	20	0	20
TPSA	15	7	5	5	14	4	5
VTA	15	15	6	15	15	4	13
VTA-Roles	15	5	4	5	15	0	5
Assembly	30	0	0	24	30	0	30
GridPl.	20	5	1	17	6	2	20
Miconic	30	9	3	13	9	2	9
$\sum$	205	88	35	154	130	25	162

Planners

FF (Hoffmann and Nebel 2001)

FD 20.06 (Helmert 2006)

PDDL variants

Original eKAB PDDL

+ Nebel's encoding

+ Derived Predicates

Metric

instances solved

Intel Core i5-4590 @3.30GHz, 8GB RAM, 600s timeout

Summary

Simple pre-processing for eKAB PDDL encodings improves performance and can even help solve standard planning benchmarks.

References

Calvanese, Diego, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati (2007). "EQL-Lite: Effective First-Order Query Processing in Description Logics". In: *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07)*, pages 274–279. URL: https://www.ijcai.org/Abstract/07/042.

Calvanese, Diego, Marco Montali, Fabio Patrizi, and Michele Stawowy (2016). "Plan Synthesis for Knowledge and Action Bases". In: *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI'16)*, pages 1022–1029. URL: https://www.ijcai.org/Abstract/16/149.

Helmert, Malte (2006). "The Fast Downward Planning System". In: *Journal of Artificial Intelligence Research* 26, pages 191–246. DOI: 10.1613/jair.1705.

Hoffmann, Jörg and Bernhard Nebel (2001). "The FF Planning System: Fast Plan Generation Through Heuristic Search". In: *Journal of Artificial Intelligence Research* 14, pages 253–302. DOI: 10.1613/jair.855.

Hoffmann, Jörg, Ingo Weber, James Scicluna, Tomasz Kacmarek, and Anupriya Ankolekar (2008). "Combining Scalability and Expressivity in the Automatic Composition of Semantic Web Services". In: *8th International Conference on Web Engineering (ICWE'08)*. DOI: 10.1109/ICWE.2008.8.

Nebel, Bernhard (2000). "On the Compilability and Expressive Power of Propositional Planning Formalisms". In: *Journal of Artificial Intelligence Research* 12, pages 271–315. DOI: 10.1613/jair.735.