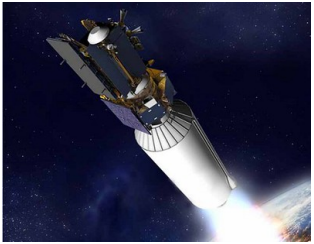
Stefan Borgwardt, Jörg Hoffmann, Alisa Kovtunova, Markus Krötzsch, Bernhard Nebel, Marcel Steinmetz
Technische Universität Dresden, Saarland University & University of Freiburg

Expressivity of Planning with Horn Description Logic Ontologies

AAAI'22, virtual, 22th February – 1st March, 2022

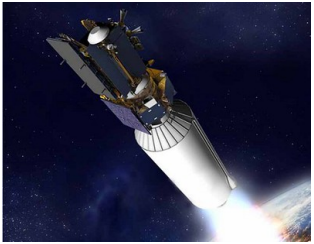
Planning and Reasoning



AI planning

- actions that modify abstract states
- fixed domain
- closed-world assumption

Planning and Reasoning



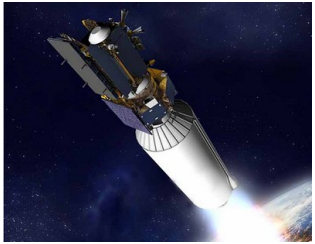
AI planning

- actions that modify abstract states
- fixed domain
- closed-world assumption

+ Static reasoning

- global state constraints
- high-level background knowledge
- open-world assumption

Planning and Reasoning



AI planning

- actions that modify abstract states
- fixed domain
- closed-world assumption

+ Static reasoning

- global state constraints
- high-level background knowledge
- open-world assumption

explicit-input knowledge and action bases (eKABs)
(Calvanese, Montali, Patrizi, and Stawowy 2016)

Planning and Reasoning

Making eKABs more expressive and practical

AI planning

- actions that modify [illegible]tes
- fixed domain
- closed-world assumption

+ Static reasoning

- global state constraints
- high-level background knowledge
- open-world assumption

explicit-input knowledge and action bases (eKABs)
(Calvanese, Montali, Patrizi, and Stawowy 2016)

Outline

Preliminaries

- Description logics (DLs)
- Planning domain definition language (PDDL)
- Explicit-input knowledge and action bases (eKABs)

Results

- Compilability
- Non-compilability
- Experiments

Description logics (DLs) – A simple example

Description logics: decidable fragments of FOL

$$\text{TBox } \mathcal{T} = \{\ \text{Drone} \sqsubseteq \text{MovingObject},$$
$$\text{Drone} \sqcap \exists \text{near}.\text{MovingObject} \sqsubseteq \text{RiskOfPhysicalDamage},$$
$$\text{Drone} \sqsubseteq \exists \text{controlledBy}.\text{Operator} \sqcup \text{AI}\ \}$$
$$\text{state } s = \{\ \text{Drone}(a),\ \text{Drone}(b),\ \text{near}(a, b)\ \}$$

Description logics (DLs) – A simple example

Description logics: decidable fragments of FOL

$$\text{TBox } \mathcal{T} = \{\ \text{Drone} \sqsubseteq \text{MovingObject},$$
$$\text{Drone} \sqcap \exists \text{near}.\text{MovingObject} \sqsubseteq \text{RiskOfPhysicalDamage},$$
$$\text{Drone} \sqsubseteq \exists \text{controlledBy}.\text{Operator} \sqcup \text{AI}\ \}$$
$$\text{state } s = \{\ \text{Drone}(a),\ \text{Drone}(b),\ \text{near}(a, b)\ \}$$

Some DLs: DL-Lite, $\mathcal{ELH}_\bot$, Horn-$\mathcal{ALCHOIQ}$, Horn-$\mathcal{SHIQ}$, Horn-$\mathcal{SROIQ}$, $\mathcal{ALCI}$, $\mathcal{SH}$, …

Description logics (DLs) – A simple example

Description logics: decidable fragments of FOL

$$\text{TBox } \mathcal{T} = \{\ \text{Drone} \sqsubseteq \text{MovingObject},$$
$$\text{Drone} \sqcap \exists \text{near}.\text{MovingObject} \sqsubseteq \text{RiskOfPhysicalDamage},$$
$$\text{Drone} \sqsubseteq \exists \text{controlledBy}.\text{Operator} \sqcup \text{AI}\ \}$$
$$\text{state } s = \{\ \text{Drone}(a),\ \text{Drone}(b),\ \text{near}(a, b)\ \}$$

Some DLs: DL-Lite, $\mathcal{ELH}_\bot$, Horn-$\mathcal{ALCHOIQ}$, Horn-$\mathcal{SHIQ}$, Horn-$\mathcal{SROIQ}$, $\mathcal{ALCI}$, $\mathcal{SH}$, …

Conjunctive queries (CQs):

$$s, \mathcal{T} \models \exists x, y.\ \text{RiskOfPhysicalDamage}(x) \wedge \text{RiskOfPhysicalDamage}(y) \wedge \text{near}(x, y)$$

Datalog rewritings for CQs over DLs

Datalog$^{(\neg)}$ rewriting $\mathcal{R}_{\mathcal{T},\mathcal{Q}}$:

$$\mathrm{MovingObject}(x) \leftarrow \mathrm{Drone}(x)$$

$$\mathrm{RiskOfPhysicalDamage}(x) \leftarrow \mathrm{Drone}(x) \land \mathrm{near}(x, y) \land \mathrm{MovingObject}(y)$$

$$\mathrm{Q} \leftarrow \mathrm{RiskOfPhysicalDamage}(x) \land \mathrm{RiskOfPhysicalDamage}(y) \land \mathrm{near}(x, y)$$

Datalog rewritings for CQs over DLs

Datalog$^{(\neg)}$ rewriting $\mathcal{R}_{\mathcal{T},Q}$:

$$\mathrm{MovingObject}(x) \leftarrow \mathrm{Drone}(x)$$
$$\mathrm{RiskOfPhysicalDamage}(x) \leftarrow \mathrm{Drone}(x) \wedge \mathrm{near}(x, y) \wedge \mathrm{MovingObject}(y)$$
$$\mathrm{Q} \leftarrow \mathrm{RiskOfPhysicalDamage}(x) \wedge \mathrm{RiskOfPhysicalDamage}(y) \wedge \mathrm{near}(x, y)$$

- exist for $\mathcal{ELH}_\bot$ (polynomial), Horn-$\mathcal{SHIQ}$ (exponential), …
- reduce open-world to closed-world reasoning

Planning domain definition language (PDDL)

(:init (Drone a) (Drone b) (near a b) . . .)

Planning domain definition language (PDDL)

```
(:init (Drone a) (Drone b) (near a b) ...)

(:action Move
  :parameters (?x ?y)
  :precondition (and (Drone ?x) (near ?x ?y) (not (Object ?y)))
  :effect (and (not (Drone ?x)) (Drone ?y))
)
```

Planning domain definition language (PDDL)

```
(:init (Drone a) (Drone b) (near a b) ...)
(:action Move
  :parameters (?x ?y)
  :precondition (and (Drone ?x) (near ?x ?y) (not (Object ?y)))
  :effect (and (not (Drone ?x)) (Drone ?y))
)
(:goal
  (not (exists (?x ?y) (and (Drone ?x) (Drone ?y) (near ?x ?y))))
)
```

Planning domain definition language (PDDL)

```
(:init (Drone a) (Drone b) (near a b) ...)
(:action Move
  :parameters (?x ?y)
  :precondition (and (Drone ?x) (near ?x ?y) (not (Object ?y)))
  :effect (and (not (Drone ?x)) (Drone ?y))
)
(:goal
  (not (exists (?x ?y) (and (Drone ?x) (Drone ?y) (near ?x ?y))))
)
```

- closed-world states
- first-order formulas in conditions and action effects
- model checking is decidable
- task: find a plan to reach the goal from the initial state

eKABs

Explicit-input knowledge and action bases (Calvanese, Montali, Patrizi, and Stawowy 2016)

"PDDL + TBox + epistemic conjunctive queries (ECQs)"

eKABs

Explicit-input knowledge and action bases (Calvanese, Montali, Patrizi, and Stawowy 2016)

"PDDL + TBox + epistemic conjunctive queries (ECQs)"

$$\mathrm{Drone} \sqcap \exists \mathrm{near}.\mathrm{MovingObject} \sqsubseteq \mathrm{RiskOfPhysicalDamage}$$

```
(:goal (not (mko (exists (?x ?y) (and
  (RiskOfPhysicalDamage ?x)
  (RiskOfPhysicalDamage ?y)
  (near ?x ?y)
)))))
```

- epistemic minimal-knowledge operator (mko)
- open-world states and conditions, but closed-world effects

eKABs

Explicit-input knowledge and action bases (Calvanese, Montali, Patrizi, and Stawowy 2016)

"PDDL + TBox + epistemic conjunctive queries (ECQs)"

$$\mathrm{Drone} \sqcap \exists \mathrm{near}.\mathrm{MovingObject} \sqsubseteq \mathrm{RiskOfPhysicalDamage}$$

```
(:goal (not (mko (exists (?x ?y) (and
  (RiskOfPhysicalDamage ?x)
  (RiskOfPhysicalDamage ?y)
  (near ?x ?y)
)))))
```

- epistemic minimal-knowledge operator (mko)
- open-world states and conditions, but closed-world effects
- split modeling into static and dynamic part
- complexity of checking entailment of ECQs depends on the TBox

Outline

Preliminaries

- Description logics (DLs)
- Planning domain definition language (PDDL)
- Explicit-input knowledge and action bases (eKABs)

Results

- Compilability
- Non-compilability
- Experiments

Compiling eKABs into PDDL

(Calvanese, Montali, Patrizi, and Stawowy 2016):

- compilation of DL-Lite eKABs into PDDL
- key: rewriting (E)CQs into first-order formulas
- however, resulting PDDL exponentially larger
- can be made practical (Borgwardt, Hoffmann, Kovtunova, and Steinmetz 2021)

Compiling eKABs into PDDL

(Calvanese, Montali, Patrizi, and Stawowy 2016):

- compilation of DL-Lite eKABs into PDDL
- key: rewriting (E)CQs into first-order formulas
- however, resulting PDDL exponentially larger
- can be made practical (Borgwardt, Hoffmann, Kovtunova, and Steinmetz 2021)

Here:

- more expressive (Horn-)DLs
- polynomial-size compilation schemes that preserve plan size polynomially (Thiébaux, Hoffmann, and Nebel 2005)
- compilation may take super-polynomial time
- use (polynomial-size) Datalog$^{\neg}$ rewritings of (E)CQs

Main Compilability Result

Theorem

For the class of eKAB instances for which all ECQs are Datalog$^\neg$ rewritable w.r.t. the TBox, there is a compilation scheme into PDDL that preserves plan size exactly.

Main Compilability Result

Theorem

For the class of eKAB instances for which all ECQs are Datalog$^\neg$ rewritable w.r.t. the TBox, there is a compilation scheme into PDDL that preserves plan size exactly.

- use derived predicates to include Datalog$^\neg$ rules in PDDL 2.2:

$$\mathrm{RiskOfPhysicalDamage}(x) \leftarrow \mathrm{Drone}(x) \wedge \mathrm{near}(x, y) \wedge \mathrm{MovingObject}(y)$$

```
(:derived (RiskOfPhysicalDamage ?x)
  (exists ?y (and (Drone ?x) (near ?x ?y) (MovingObject ?y)))
)
```

Main Compilability Result

Theorem

For the class of eKAB instances for which all ECQs are Datalog¬ rewritable w.r.t. the TBox, there is a compilation scheme into PDDL that preserves plan size exactly.

- use derived predicates to include Datalog¬ rules in PDDL 2.2:

$$\mathrm{RiskOfPhysicalDamage}(x) \leftarrow \mathrm{Drone}(x) \wedge \mathrm{near}(x, y) \wedge \mathrm{MovingObject}(y)$$

```
(:derived (RiskOfPhysicalDamage ?x)
  (exists ?y (and (Drone ?x) (near ?x ?y) (MovingObject ?y)))
)
```

- (plus some technicalities, optimizations)
- size of PDDL depends on size of Datalog¬ rewritings

Main Compilability Result

Theorem

For the class of eKAB instances for which all ECQs are Datalog¬ rewritable w.r.t. the TBox, there is a compilation scheme into PDDL that preserves plan size exactly.

- use derived predicates to include Datalog¬ rules in PDDL 2.2:

$$\mathrm{RiskOfPhysicalDamage}(x) \leftarrow \mathrm{Drone}(x) \wedge \mathrm{near}(x, y) \wedge \mathrm{MovingObject}(y)$$

```
(:derived (RiskOfPhysicalDamage ?x)
  (exists ?y (and (Drone ?x) (near ?x ?y) (MovingObject ?y)))
)
```

- (plus some technicalities, optimizations)
- size of PDDL depends on size of Datalog¬ rewritings
- immediately applicable to existing Datalog¬ rewritings for DLs
- [new polynomial-size Datalog¬ rewriting for Horn-$\mathcal{ALCHOIQ}$]

Polynomial-size Datalog$^{\neg}$ rewriting for Horn-$\mathcal{ALCHOIQ}$

Combined rewriting for Horn-$\mathcal{ALCHOIQ}$: (Carral, Dragoste, and Krötzsch 2018)

- exponential Datalog program using types $C(x) \to R(x, t_D) \wedge D(t_D)$
- complex filtration phase "check whether $F_{q,\sigma}$ is a rooted directed forest"

Polynomial-size Datalog$^{\neg}$ rewriting for Horn-$\mathcal{ALCHOIQ}$

Combined rewriting for Horn-$\mathcal{ALCHOIQ}$: (Carral, Dragoste, and Krötzsch 2018)

- exponential Datalog program using types $C(x) \rightarrow R(x, t_D) \wedge D(t_D)$
- complex filtration phase "check whether $F_{q,\sigma}$ is a rooted directed forest"

Transform into polynomial rule set:

- represent types using Datalog^S set terms (Ortiz, Rudolph, and Šimkus 2010)

$$\mathrm{concept}(C, X) \rightarrow \mathrm{role}(r, X, \{D\}) \wedge \mathrm{concept}(D, \{D\})$$

Polynomial-size Datalog$^{\neg}$ rewriting for Horn-$\mathcal{ALCHOIQ}$

Combined rewriting for Horn-$\mathcal{ALCHOIQ}$: (Carral, Dragoste, and Krötzsch 2018)

- exponential Datalog program using types $C(x) \to R(x, t_D) \wedge D(t_D)$
- complex filtration phase "check whether $F_{q,\sigma}$ is a rooted directed forest"

Transform into polynomial rule set:

- represent types using Datalog^S set terms (Ortiz, Rudolph, and Šimkus 2010)

$$\mathrm{concept}(C, X) \to \mathrm{role}(r, X, \{D\}) \wedge \mathrm{concept}(D, \{D\})$$

- encode filtration into stratified Datalog$^{S,\neg}$

$$\mathrm{edge}(i, j, V_1, \ldots, V_k) \to \mathrm{reach}(i, j, V_1, \ldots, V_k)$$
$$\ldots$$
$$\mathrm{reach}(i, i, V_1, \ldots, V_k) \to \mathrm{bad}(V_1, \ldots, V_k)$$

Polynomial-size Datalog$^\neg$ rewriting for Horn-$\mathcal{ALCHOIQ}$

Combined rewriting for Horn-$\mathcal{ALCHOIQ}$: (Carral, Dragoste, and Krötzsch 2018)

- exponential Datalog program using types $C(x) \to R(x, t_D) \wedge D(t_D)$
- complex filtration phase "check whether $F_{q,\sigma}$ is a rooted directed forest"

Transform into polynomial rule set:

- represent types using Datalog^S set terms (Ortiz, Rudolph, and Šimkus 2010)

$$\mathrm{concept}(C, X) \to \mathrm{role}(r, X, \{D\}) \wedge \mathrm{concept}(D, \{D\})$$

- encode filtration into stratified Datalog$^{S,\neg}$

$$\mathrm{edge}(i, j, V_1, \ldots, V_k) \to \mathrm{reach}(i, j, V_1, \ldots, V_k)$$
$$\ldots$$
$$\mathrm{reach}(i, i, V_1, \ldots, V_k) \to \mathrm{bad}(V_1, \ldots, V_k)$$

- translate set terms back into Datalog$^\neg$ using bit vectors

Non-compilability

(Thiébaux, Hoffmann, and Nebel 2005)

What about more expressive DLs?

Horn-$\mathcal{SROIQ}$ = extension of Horn-$\mathcal{ALCHOIQ}$ with one additional statement type

Non-compilability

(Thiébaux, Hoffmann, and Nebel 2005)

What about more expressive DLs?

Horn-$\mathcal{SROIQ}$ = extension of Horn-$\mathcal{ALCHOIQ}$ with one additional statement type

Theorem

Unless ExpTime$^{\text{NP}}$ = ExpTime*, there is no polynomial-size compilation scheme from Horn-$\mathcal{SROIQ}$ eKABs to PDDL preserving plan size polynomially.*

- based on 2ExpTime-hardness proof for CQ entailment in Horn-$\mathcal{SROIQ}$
- Horn-$\mathcal{SROIQ}$ eKABs (non-uniformly) simulate universal 2ExpTime Turing machine
- polynomial compilation implies 2ExpTime $\subseteq$ ExpTime/poly

Non-compilability

(Thiébaux, Hoffmann, and Nebel 2005)

What about more expressive DLs?

Horn-$\mathcal{SROIQ}$ = extension of Horn-$\mathcal{ALCHOIQ}$ with one additional statement type

Theorem

Unless ExpTime$^{\text{NP}}$ = ExpTime*, there is no polynomial-size compilation scheme from Horn-$\mathcal{SROIQ}$ eKABs to PDDL preserving plan size polynomially.*

- based on 2ExpTime-hardness proof for CQ entailment in Horn-$\mathcal{SROIQ}$
- Horn-$\mathcal{SROIQ}$ eKABs (non-uniformly) simulate universal 2ExpTime Turing machine
- polynomial compilation implies 2ExpTime $\subseteq$ ExpTime/poly
- $\Rightarrow$ weak exponential hierarchy collapses (Buhrman and Homer 1992)
- $\Rightarrow$ any bounded quantifier prefix in second-order logic or Presburger arithmetic could be eliminated (Gottlob, Leone, and Veith 1995; Haase 2014)

Non-compilability

(Thiébaux, Hoffmann, and Nebel 2005)

What about more expressive DLs?

Horn-$\mathcal{SROIQ}$ = extension of Horn-$\mathcal{ALCHOIQ}$ with one additional statement type

Theorem

Unless ExpTime$^{\text{NP}}$ = ExpTime*, there is no polynomial-size compilation scheme from Horn-$\mathcal{SROIQ}$ eKABs to PDDL preserving plan size polynomially.*

- based on 2ExpTime-hardness proof for CQ entailment in Horn-$\mathcal{SROIQ}$
- Horn-$\mathcal{SROIQ}$ eKABs (non-uniformly) simulate universal 2ExpTime Turing machine
- polynomial compilation implies 2ExpTime $\subseteq$ ExpTime/poly
- $\Rightarrow$ weak exponential hierarchy collapses (Buhrman and Homer 1992)
- $\Rightarrow$ any bounded quantifier prefix in second-order logic or Presburger arithmetic could be eliminated (Gottlob, Leone, and Veith 1995; Haase 2014)
- [similar results for $\mathcal{SH}$, $\mathcal{ALCI}$ eKABs]

Experiments

- implementation of compilation scheme using practical but exponential Datalog$^\neg$ rewriting for Horn-$\mathcal{SHIQ}$ (Clipper) (Eiter, Ortiz, Šimkus, Tran, and Xiao 2012)

Experiments

- implementation of compilation scheme using practical but exponential Datalog$^\neg$ rewriting for Horn-$\mathcal{SHIQ}$ (Clipper) (Eiter, Ortiz, Šimkus, Tran, and Xiao 2012)
- input: .pddl file with `mko` operators, .ttl/.owl file for the TBox
- output: compiled, normalized standard .pddl file
- planning: FD 20.06 (Helmert 2006)

Experiments

- implementation of compilation scheme using practical but exponential Datalog$^\neg$ rewriting for Horn-$\mathcal{SHIQ}$ (Clipper) (Eiter, Ortiz, Šimkus, Tran, and Xiao 2012)
- input: .pddl file with `mko` operators, .ttl/.owl file for the TBox
- output: compiled, normalized standard .pddl file
- planning: FD 20.06 (Helmert 2006)
- benchmarks:
  - 125 existing DL-Lite eKAB instances
  - 110 new, more expressive instances

Experiments

- implementation of compilation scheme using practical but exponential Datalog$^\neg$ rewriting for Horn-$\mathcal{SHIQ}$ (Clipper) (Eiter, Ortiz, Šimkus, Tran, and Xiao 2012)
- input: .pddl file with `mko` operators, .ttl/.owl file for the TBox
- output: compiled, normalized standard .pddl file
- planning: FD 20.06 (Helmert 2006)
- benchmarks:
  - 125 existing DL-Lite eKAB instances
  - 110 new, more expressive instances
- comparison with (Calvanese, Montali, Patrizi, and Stawowy 2016; Borgwardt, Hoffmann, Kovtunova, and Steinmetz 2021)
- platform: Intel Core i5-4590 CPU @3.30GHz, 8GB memory, 600s timeout

Results

Domain	#	# solved Cal16	# solved Bor21	# solved Horn	planning time Cal16	planning time Bor21	planning time Horn
Cats	20	14	20	20	63.46	0.13	0.03
Elevator	20	20	20	20	0.36	0.30	0.03
Robot	20	4	12	20	15.05	10.10	0.11
TaskAssign	20	3	20	20	0.81	0.12	0.06
TPSA	15	14	5	15	2.01	2.42	0.30
VTA	15	15	13	15	23.06	371.56	16.91
VTA-Roles	15	15	5	15	2.25	11.61	1.36
$\sum$	125	85	95	125	19.99	77.33	3.59
Drones	24			20			101.42
Queens	30			15			21.66
RobotConj	56			56			8.14
$\sum$	110			91			30.87

- our method (“Horn”) solves all DL-Lite instances very fast
- feasible also for more expressive TBoxes
- compilation itself also much faster on average

Summary

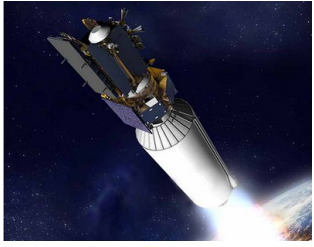
- new compilation for eKABs with expressive Horn-DL TBoxes
- outperforms previous eKAB compilations
- theoretical results on polynomial compilability (Horn-$\mathcal{ALCHOIQ}$ vs. Horn-$\mathcal{SROIQ}$)

Summary

- new compilation for eKABs with expressive Horn-DL TBoxes
- outperforms previous eKAB compilations
- theoretical results on polynomial compilability (Horn-$\mathcal{ALCHOIQ}$ vs. Horn-$\mathcal{SROIQ}$)

Future work:

- implement practical compilations beyond Horn-$\mathcal{SHIQ}$
- adapt to other semantics for action effects

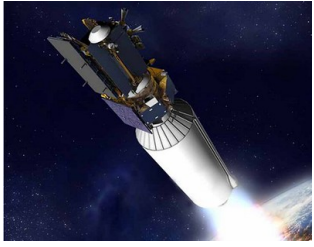


Summary

- new compilation for eKABs with expressive Horn-DL TBoxes
- outperforms previous eKAB compilations
- theoretical results on polynomial compilability (Horn-$\mathcal{ALCHOIQ}$ vs. Horn-$\mathcal{SROIQ}$)

Future work:

- implement practical compilations beyond Horn-$\mathcal{SHIQ}$
- adapt to other semantics for action effects



Thank you!

References I

Borgwardt, Stefan, Jörg Hoffmann, Alisa Kovtunova, and Marcel Steinmetz (2021). "Making DL-Lite Planning Practical". In: *Proc. of the 18th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'21)*, pages 641–645. doi: 10.24963/kr.2021/61.

Buhrman, Harry and Steven Homer (1992). "Superpolynomial Circuits, Almost Sparse Oracles and the Exponential Hierarchy". In: *Proc. of the 12th Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'92)*, pages 116–127. doi: 10.1007/3-540-56287-7__99.

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

Carral, David, Irina Dragoste, and Markus Krötzsch (2018). "The Combined Approach to Query Answering in Horn-$\mathcal{ALCHOIQ}$". In: *Proc. of the 16th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'18)*, pages 339–348. url: https://aaai.org/ocs/index.php/KR/KR18/paper/view/18076.

Eiter, Thomas, Magdalena Ortiz, Mantas Šimkus, Trung-Kien Tran, and Guohui Xiao (2012). "Query Rewriting for Horn-$\mathcal{SHIQ}$ Plus Rules". In: *Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI'12)*, pages 726–733. url: http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4931.

Gottlob, Georg, Nicola Leone, and Helmut Veith (1995). "Second Order Logic and the Weak Exponential Hierarchies". In: *Proc. of the 20th Int. Symp. on Mathematical Foundations of Computer Science (MFCS'95)*, pages 66–81. doi: 10.1007/3-540-60246-1__113.

Haase, Christoph (2014). "Subclasses of Presburger Arithmetic and the Weak EXP Hierarchy". In: *Proc. of the Joint Meeting of the 23rd EACSL Annual Conf. on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symp. on Logic in Computer Science (LICS)*, 14:1–14:10. doi: 10.1145/2603088.2603092.

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

References II

Ortiz, Magdalena, Sebastian Rudolph, and Mantas Šimkus (2010). "Worst-case Optimal Reasoning for the Horn-DL Fragments of OWL 1 and 2". In: *Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'10)*, pages 269–279. url: https://aaai.org/ocs/index.php/KR/KR2010/paper/view/1296.

Thiébaux, Sylvie, Jörg Hoffmann, and Bernhard Nebel (2005). "In Defense of PDDL Axioms". In: *Artificial Intelligence* 168, pages 38–69. doi: 10.1016/j.artint.2005.05.004.