Journées des Doctorants 2017

Practical resolution of the coherence of formulae in modal logic

Valentin Montmirail

CRIL-CNRS UMR 8188, F62300 Lens, France

Le Touquet - May 29th 2017

UNIVERSITÉ D'ARTOIS

Thesis Subject

Thesis goal

Create an efficient solver to answer modal logic satisfiability problems [BvBW06] (Called here Modal-SAT)

- Reduction from Modal-SAT to an existing problem for which there are already solvers (SAT [SV09], SMT [AFM15], ASP [ON04], . . .)
- Create an "ad-hoc" solver for Modal-SAT by adapting the techniques of the best solvers above

Modal Logic = Propositional Logic + $\Box$ and $\Diamond$

Modal Logic

- $\Box\phi$ means ϕ is necessarily true
- $\Diamond\phi$ means ϕ is possibly true

$$\Diamond\phi \leftrightarrow \neg\Box\neg\phi$$

$$\Box\phi \leftrightarrow \neg\Diamond\neg\phi$$

Preliminaries: Kripke Structure

- $\mathbb{P}$ finite non-empty set of propositional variables

Kripke Structure [Kri59]

$M = \langle W, R, V \rangle$ with:

- W, a non-empty set of possible worlds;
- R, a binary relation on W;
- V, a function that associate to each $p \in \mathbb{P}$, the set of possible worlds where p is true.

Pointed Kripke Structure: $\langle \mathcal{K}, w \rangle$

- $\mathcal{K}$: Kripke Structure
- w is a possible world in W

Preliminaries: Satisfaction Relation

Definition (Satisfaction Relation)

The relation $\models$ between Kripke Structures and formulae is recursively defined as follows:

$$
\begin{array}{lll}
\langle \mathcal{K}, w \rangle \models p & \text{iff} & w \in V(p) \\
\langle \mathcal{K}, w \rangle \models \neg\phi & \text{iff} & \langle \mathcal{K}, w \rangle \not\models \phi \\
\langle \mathcal{K}, w \rangle \models \phi_1 \wedge \phi_2 & \text{iff} & \langle \mathcal{K}, w \rangle \models \phi_1 \text{ and } \langle \mathcal{K}, w \rangle \models \phi_2 \\
\langle \mathcal{K}, w \rangle \models \phi_1 \vee \phi_2 & \text{iff} & \langle \mathcal{K}, w \rangle \models \phi_1 \text{ or } \langle \mathcal{K}, w \rangle \models \phi_2 \\
\langle \mathcal{K}, w \rangle \models \Box\phi & \text{iff} & (w, w') \in R \text{ implies } \langle \mathcal{K}, w' \rangle \models \phi \\
\langle \mathcal{K}, w \rangle \models \Diamond\phi & \text{iff} & (w, w') \in R \text{ and } \langle \mathcal{K}, w' \rangle \models \phi
\end{array}
$$

$\mathcal{K}$ that satisfied a formula ϕ will be called “Kripke model of ϕ”

Preliminaries: Example of a Kripke Structure

- ✔ $\phi_1 = \Box(\bullet)$
- ✖ $\phi_2 = \Box\Diamond(\bullet)$

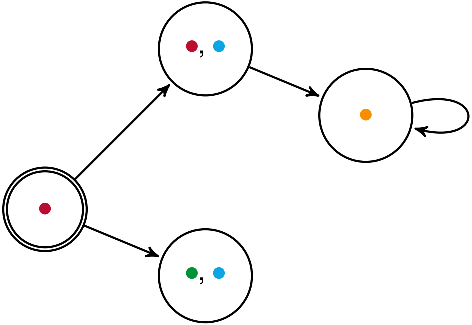


Figure: Example $\mathcal{K}$

Preliminaries: Different axioms ↔ Different logics

Axioms Schemata and corresponding properties

Name	Condition on $\mathcal{K}$	First Order constraint
Axiom (K)	None	
Axiom (T)	Reflexivity	$\forall w.R(w,w)$
Axiom (B)	Symmetry	$\forall w_1.\forall w_2.(R(w_1,w_2) \rightarrow R(w_2,w_1))$
Axiom (D)	Seriality	$\forall w_1.\exists w_2.R(w_1,w_2)$
Axiom (4)	Transitivity	$\forall w_1.\forall w_2.\forall w_3.((R(w_1,w_2) \wedge R(w_2,w_3)) \rightarrow R(w_1,w_3))$
Axiom (5)	Euclideanity	$\forall w_1.\forall w_2.\forall w_3.((R(w_1,w_2) \wedge R(w_1,w_3)) \rightarrow R(w_2,w_3))$

Structure Kind	Structural Properties	Structure Kind	Structural Properties
K		(S4) KT4 = KDT4	Reflexive and Transitive
KB	Symmetric	K45	Transitive and Euclidean
KT = KDT	Reflexive	KD	Serial
K4	Transitive	KDB	Serial and Symmetric
K5	Euclidean	KD4	Serial and Transitive
KBT = KBDT	Symmetric and Reflexive	KD5	Serial and Euclidean
KB4 = KB5 = KB45	Symmetric and Transitive	KD45	Serial, Transitive and Euclidean
(S5) KT5 = KBD4 = KBD5 = KBT4 = KBT5 = KDT5 = KT45 = KBD45 = KBT45 = KDT45 = KBDT4 = KBDT5 = KBDT45			Equivalence

Preliminaries: Different logics → Different complexities

Sum-up of complexities

NP
K5
K45
KB45
KD5
KD45
S5

PSPACE
K
KT
S4
KB
KD4
KD
K4
KDB
KBT

Proofs of complexities are in [Lad77, HR07]

Modal Logic S5 solver: S52SAT

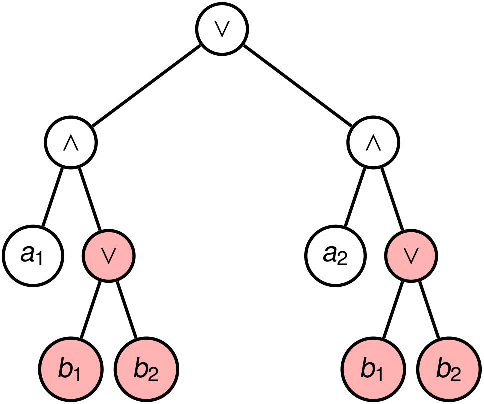
- Translation from S5-SAT to SAT.
- Polynomial reduction: S5-SAT is NP-complete [Lad77]
- Improvement upper-bound: Diamond-Degree (dd) [CLB+17]

$$\mathrm{tr}(\phi, n) = \mathrm{tr}(\phi, 1, n)$$
$$\mathrm{tr}(p, i, n) = p_i$$
$$\mathrm{tr}(\neg\psi, i, n) = \neg\,\mathrm{tr}(\psi, i, n)$$
$$\mathrm{tr}(\Box\phi, i, n) = \bigwedge_{j=1}^{n}((\mathrm{tr}(\phi, j, n))$$
$$\mathrm{tr}(\Diamond\phi, i, n) = \bigvee_{j=1}^{n}((\mathrm{tr}(\phi, j, n))$$

$$\mathrm{dd}(\phi) = \mathrm{dd}'(\mathrm{nnf}(\phi))$$
$$\mathrm{dd}'(\top) = \mathrm{dd}'(\neg\top) = 0$$
$$\mathrm{dd}'(p) = \mathrm{dd}'(\bot) = 0$$
$$\mathrm{dd}'(\phi \wedge \psi) = \mathrm{dd}'(\phi) + \mathrm{dd}'(\psi)$$
$$\mathrm{dd}'(\phi \vee \psi) = \max(\mathrm{dd}'(\phi), \mathrm{dd}'(\psi))$$
$$\mathrm{dd}'(\Box\phi) = \mathrm{dd}'(\phi)$$
$$\mathrm{dd}'(\Diamond\phi) = 1 + \mathrm{dd}'(\phi)$$

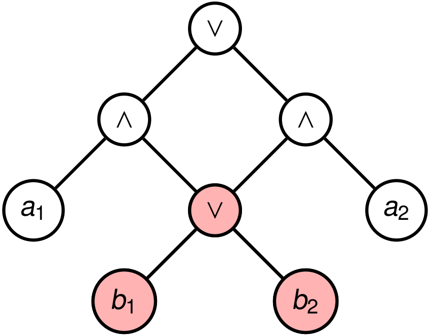
Modal Logic S5 solver: S52SAT

Let $\phi = \Diamond(a \wedge \Diamond b)$ as example (with n = 2).



Modal Logic S5 solver: S52SAT - structural caching

$\phi = \Diamond(a \wedge \Diamond b)$, with caching.



Modal Logic S5 solver: S52SAT - with/without caching

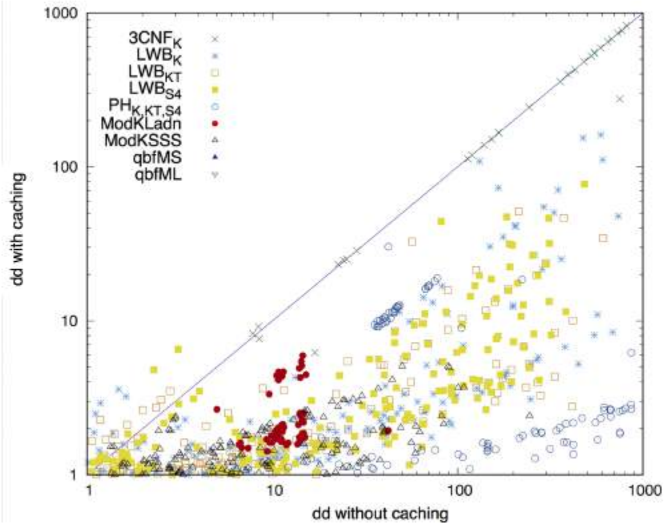


Figure: Scatter plot with/without caching

Modal Logic S5 solver: S52SAT - against SotA solvers

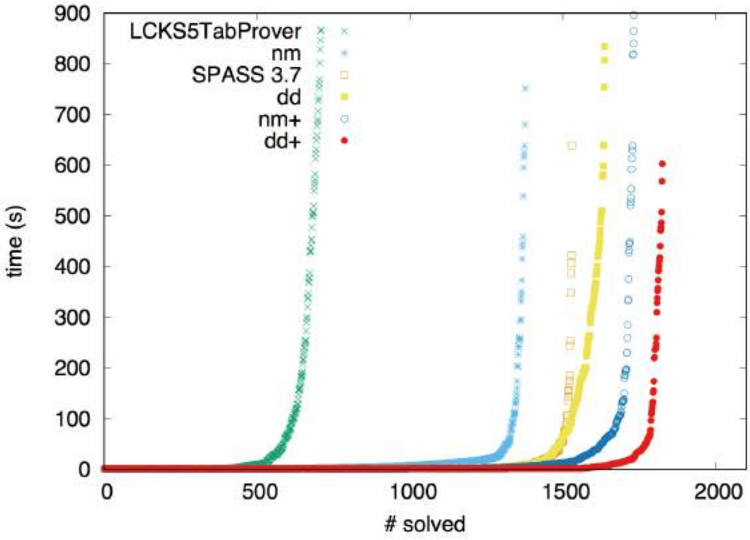


Figure: Cactus-Plot of the runtime distributions

Different logics → Different complexities

Sum-up of complexities

NP
K5
K45
KB45
KD5
KD45
S5

PSPACE
K
KT
S4
KB
KD4
KD
K4
KDB
KBT

Proofs of complexities are in [Lad77, HR07]

Modal Logic K : Translation to SAT

Translating in ‘one shot’ as in S5 is not efficient in K

- To translate into SAT will use too much memory (PSPACE)
- Diamond-Degree can not be used in modal logic K [LBdLM17]
- We want a framework to deal with PSPACE problems

Modal Logic K : Translation to SAT

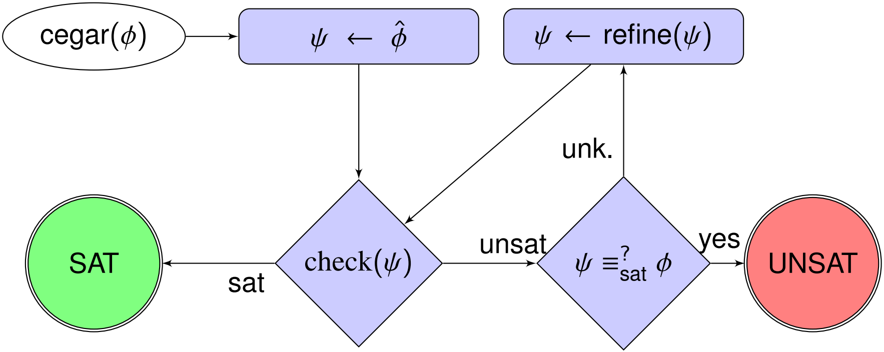
Translating in ‘one shot’ as in S5 is not efficient in K

- To translate into SAT will use too much memory (PSPACE)
- Diamond-Degree can not be used in modal logic K [LBdLM17]
- We want a framework to deal with PSPACE problems

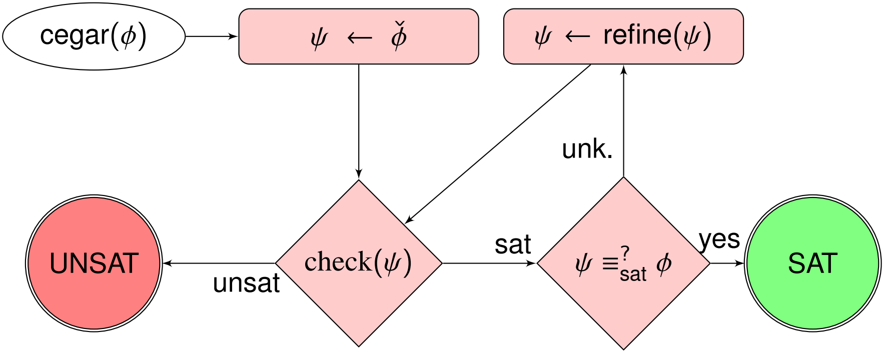
Recursive Explore and Check Abstraction Refinement

- Inspired by CEGAR [CGJ+03]
- Called *RECAR*
- Rely on 5 very important assumptions.

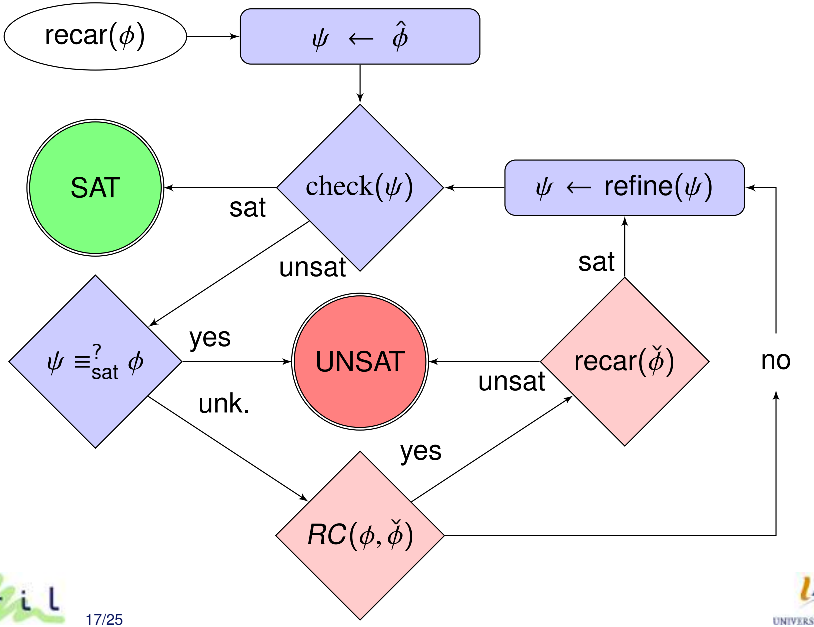
CounterExample Guided Abstraction Refinement: Over



CounterExample Guided Abstraction Refinement: Under



Recursive Explore and Check Abstraction Refinement



Recursive Explore and Check Abstraction Refinement

RECAR Assumptions

1. Function 'check' is sound, complete and terminates.

2. $\mathit{isSAT}(\hat{\phi})$ implies $\mathit{isSAT}(\text{refine}(\hat{\phi}))$
3. $\exists.n \in \mathbb{N}$ such that $\text{refine}^n(\hat{\phi}) \equiv^?_{\text{sat}} \phi$.

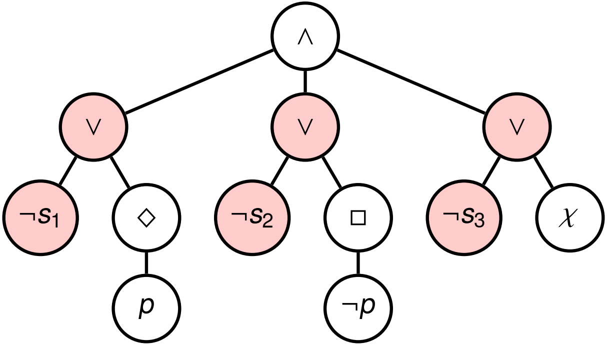
4. $\text{isUNSAT}(\check{\phi})$ implies $\text{isUNSAT}(\phi)$.
5. $\exists.n \in \mathbb{N}$ such that $\text{under}^n(\phi) \equiv^?_{\text{sat}} \text{under}^{n+1}(\phi)$.

MoSaiC : Over-Approximation

$$
\begin{aligned}
\text{over}(\phi, n) &= \text{over}(\phi, 0, n) \\
\text{over}(p_i, prev, n) &= p_{i,prev} \\
\text{over}(\neg\phi, prev, n) &= \neg\,\text{over}(\phi, prev, n) \\
\text{over}(\Box\phi, prev, n) &= \bigwedge_{i=0}^{n} (r_{prev,i} \rightarrow \text{over}(\phi, i, n)) \\
\text{over}(\Diamond\phi, prev, n) &= \bigvee_{i=0}^{n} (r_{prev,i} \wedge \text{over}(\phi, i, n))
\end{aligned}
$$

- $p_{i,j}$ means p_i is true in the world w_j.
- $r_{i,j}$ means that there is a relation between worlds w_i and w_j.

MoSaiC : Under-Approximation



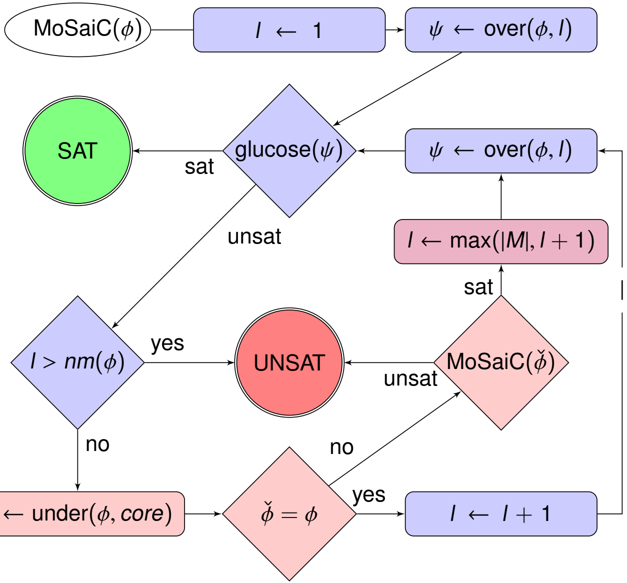
Modern SAT solvers returns ‘the reason’ why a formula with n worlds is unsatisfiable thanks to unsatisfiable cores.

MoSaiC : Under-Approximation

$$\text{under}(p, core) = p$$
$$\text{under}(\neg p, core) = \neg p$$
$$\text{under}(\Box\phi, core) = \Box(\text{under}(\phi, core))$$
$$\text{under}(\Diamond\phi, core) = \Diamond(\text{under}(\phi, core))$$
$$\text{under}((\phi \wedge \psi), core) = \text{under}(\phi, core) \wedge \text{under}(\psi, core)$$
$$\text{under}((\psi \vee \chi), core) = \begin{cases} \text{under}(\chi, core) & \text{if } \psi = \neg s_i, s_i \in core \\ \top & \text{if } \psi = \neg s_i, s_i \notin core \\ (\text{under}(\psi, core) & \\ \vee\, \text{under}(\chi, core)) & \text{otherwise} \end{cases}$$

- Unsatisfiable-cores: To create our under-approximations.

MoSaiC: RECAR Modal Logic K



MoSaiC : RECAR Modal Logic K

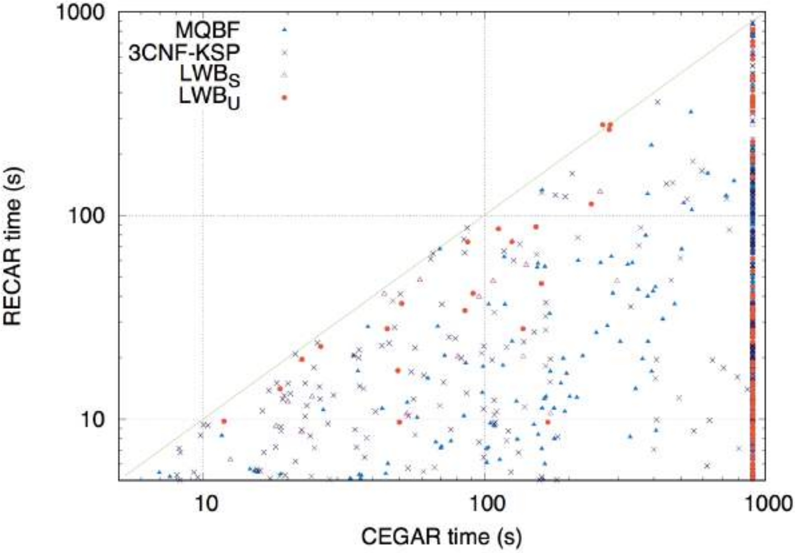


Figure: Scatter plot CEGAR vs RECAR

MoSaiC : RECAR Modal Logic K

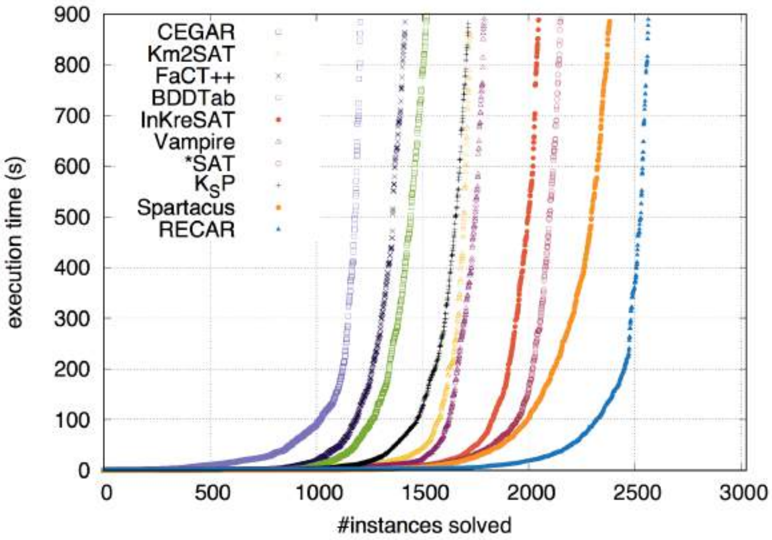


Figure: Cactus-Plot of the runtime distributions

Perspective: Other modal logics

Sum-up of complexities

NP
K5
K45
KB45
KD5
KD45
S5

PSPACE
K
KT
S4
KB
KD4
KD
K4
KDB
KBT

Perspective:

- How to deal with other modal logics?
- What about Diamond-Degree in other logics ?

Journées des Doctorants 2017

Practical resolution of the coherence of formulae in modal logic

Valentin Montmirail

CRIL-CNRS UMR 8188, F62300 Lens, France

Le Touquet - May 29th 2017

Bibliography I

- Carlos Areces, Pascal Fontaine, and Stephan Merz.
  Modal satisfiability via SMT solving.
  In *Software, Services, and Systems*, pages 30–45. Springer, 2015.
- Patrick Blackburn, Johan van Benthem, and Frank Wolter.
  Handbook of modal logic.
  Elsevier, 2006.
- Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith.
  Counterexample-guided abstraction refinement for symbolic model checking.
  Journal of the ACM, 50(5):752–794, 2003.

Bibliography II

Thomas Caridroit, Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail.
A SAT-based Approach For Solving The Modal Logic S5-Satisfiability Problem.
In *Proc. of AAAI'17*, 2017.

Joseph Y. Halpern and Leandro Chaves Rêgo.
Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic.
J. Log. Comput., 17(4):795–806, 2007.

Saul Kripke.
A completeness theorem in modal logic.
J. Symb. Log., 24(1):1–14, 1959.

Bibliography III

Richard E. Ladner.
The Computational Complexity of Provability in Systems of Modal Propositional Logic.
SIAM J. Comput., 6(3):467–480, 1977.

Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail.
A Recursive Shortcut for CEGAR: Application To The Modal Logic K Satisfiability Problem.
In *Proc. of IJCAI'17*, 2017.

Mauricio Osorio and Juan Antonio Navarro.
Answer set programming and S4.
In *Ibero-American Conference on Artificial Intelligence*, pages 353–363. Springer, 2004.

Bibliography IV

Roberto Sebastiani and Michele Vescovi.
Automated Reasoning in Modal and Description Logics via SAT Encoding: the Case Study of K(m)/ALC-Satisfiability.
Journal of Artificial Intelligence Research, 35:343–389, 2009.