KSP: A Resolution-based Prover for Multimodal K Abridged Report

Cláudia Nalon
Department of Computer Science
University of Brasília
nalon@unb.br

Ullrich Hustadt and **Clare Dixon**
Department of Computer Science
University of Liverpool
{U.Hustadt,C.Dixon}@liverpool.ac.uk

Abstract

We briefly describe an implementation of a hyper-resolution-based calculus for the propositional basic multimodal logic, K_n. The prover, KSP, which allows for both local and global reasoning, is designed to support experimentation with different combinations of refinements for its basic calculus. We present an experimental evaluation that compares KSP with a range of existing reasoners for K_n.

1 Introduction

Modal logics have long been used in Computer Science for describing and reasoning about complex systems, including programming languages [Pratt, 1980], knowledge representation and reasoning [Rao and Georgeff, 1991; Halpern and Moses, 1992], verification of distributed systems [Hailpern, 1982; Halpern *et al.*, 1983; Halpern, 1987] and terminological reasoning [Schild, 1991]. The most basic of such logics is the multimodal K_n, which extends the classical language with new operators, $\Box_a$ and $\Diamond_a$, with $a \in \mathsf{A} = \{1, \ldots, n\}$, a fixed finite set of indexes (Section 2). The local satisfiability problem for the multimodal propositional case is PSPACE-complete [Halpern and Moses, 1992]. The global satisfiability and the local satisfiability under global constraint problems for K_n are EXPTIME-complete [Spaan, 1993]. Given the inherent intractability of the reasoning problem and also the wide range of applications to which those logics can be applied, the development of automatic, efficient tools for theorem proving is highly desirable.

Several proof methods and tools for modal reasoning exist, either in the form of methods applied direct to the modal language or obtained by translation into more expressive languages (First-Order Logic, for instance). Here we concentrate on a direct method for K_n, the modal resolution procedure described in [Nalon *et al.*, 2015]. Resolution [Robinson, 1965] is a saturation procedure, that is, formulae which are consequence of a given input set are *added* to this set until a contradiction is derived or the satisfiability of the set is evident. Following successful refinements and strategies used in theorem-provers for classical logic (see [Schulz and Möhrmann, 2016], for instance), our calculus is designed for restricting as much as possible the number of clauses produced during saturation. In particular, we take advantage of well-known properties for the satisfiability problem for K_n, namely: a formula φ is satisfiable if, and only if, it is satisfiable in a finite tree-like model; and also that checking the satisfiability of a subformula associated with level ml of the tree depends only on the subformulae occurring at levels greater or equal to ml [Areces *et al.*, 2000]. The procedure, which is presented in Section 3, requires the translation of a formula φ into a more expressive language, where labels are used to denote the level at which φ occurs. Labelled resolution is then applied in order to determine the satisfiability of φ.

In this paper, we briefly present KSP, a theorem prover for the basic multimodal logic K_n which implements the calculus described in [Nalon *et al.*, 2015] as a variation of the set of support strategy [Wos *et al.*, 1965]. The prover also implements several other refinements and simplification techniques in order to reduce the search space for a proof (Section 4). Besides the set of support strategy, all other refinements of the calculus are implemented as independent modules, allowing for a better evaluation of how effective they are. The experimental evaluation indicates that KSP works well on problems with high nesting of modal operators where the separation of modal layers can be exploited to improve the efficiency of reasoning (Section 5). An extended version of this paper can be found at [Nalon *et al.*, 2016a].

2 Language

Let $\mathsf{A} = \{1, \ldots, n\}$, $n \in \mathbb{N}$, be a finite fixed set of indexes and $\mathsf{P} = \{p, q, s, t, p', q', \ldots\}$ be a denumerable set of propositional symbols. The *set of modal formulae*, WFF, is the least set such that every $p \in \mathsf{P}$ is in WFF; if φ and ψ are in WFF, then so are $\neg\varphi$, $(\varphi \wedge \psi)$, and $\Box_a\varphi$ for each $a \in \mathsf{A}$. The formulae **false**, **true**, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, and $\Diamond_a\varphi$ are introduced as the abbreviations for $(\varphi \wedge \neg\varphi)$, $\neg$**false**, $\neg(\neg\varphi \wedge \neg\psi)$, $(\neg\varphi \vee \psi)$, and $\neg\Box_a\neg\varphi$, respectively (where $\varphi, \psi \in \mathsf{WFF}$). A *literal* is either a propositional symbol or its negation; the set of literals is denoted by L. A *modal literal* is either $\Box_a l$ or $\Diamond_a l$, where $l \in \mathsf{L}$ and $a \in \mathsf{A}$. The literals (resp. modal literals) l and $\neg l$ (resp. $\Box_a l$ and $\Diamond_a \neg l = \neg\Box_a l$) are said to be *complementary*. The *modal depth* of a formula is given by the maximal number of nested occurrences of modal operators in that formula. The *modal level* of a subformula is the maximal number of nested occurrences of modal operators in which scope the formula occurs. For instance, in $\Box_a \Diamond_a p$,

$$[\,\mathsf{LRES}]\quad \frac{ml_1:\ D \vee l \qquad ml_2:\ D' \vee \neg l}{ml:\ D \vee D'} \qquad [\mathsf{MRES}]\quad \frac{ml_1:\ l_1 \Rightarrow \boxed{a} l \qquad ml_2:\ l_2 \Rightarrow \diamondsuit_{\boxed{a}} \neg l}{ml:\ \neg l_1 \vee \neg l_2}$$

$$[\mathsf{GEN2}]\quad \frac{ml_1:\ l'_1 \Rightarrow \boxed{a} l_1 \qquad ml_2:\ l'_2 \Rightarrow \boxed{a} \neg l_1 \qquad ml_3:\ l'_3 \Rightarrow \diamondsuit_{\boxed{a}} l_2}{ml:\ \neg l'_1 \vee \neg l'_2 \vee \neg l'_3}$$

$$[\mathsf{GEN1}]\quad \frac{\begin{array}{r} ml_1:\ l'_1 \Rightarrow \boxed{a} \neg l_1 \\ \vdots \\ ml_m: l'_m \Rightarrow \boxed{a} \neg l_m \\ ml_{m+1}:\ l' \Rightarrow \diamondsuit_{\boxed{a}} \neg l \\ ml_{m+2}:\ l_1 \ \vee\ \ldots \vee l_m \vee l \end{array}}{ml:\ \neg l'_1 \vee \ldots \vee \neg l'_m \vee \neg l'} \qquad [\mathsf{GEN3}]\quad \frac{\begin{array}{r} ml_1:\ l'_1 \Rightarrow \boxed{a} \neg l_1 \\ \vdots \\ ml_m: l'_m \Rightarrow \boxed{a} \neg l_m \\ ml_{m+1}:\ l' \Rightarrow \diamondsuit_{\boxed{a}} l \\ ml_{m+2}:\ l_1 \ \vee\ \ldots \vee l_m \end{array}}{ml:\ \neg l'_1 \vee \ldots \vee \neg l'_m \vee \neg l'}$$

Figure 1: Inference rules, where $ml = \sigma(\{ml_1, \ldots, ml_{m+1}, ml_{m+2} - 1\})$ in GEN1, GEN3; $ml = \sigma(\{ml_1, ml_2\})$ in LRES, MRES; and $ml = \sigma(\{ml_1, ml_2, ml_3\})$ in GEN2.

the modal depth of p is 0 and its modal level is 2. Formal definitions can be found in [Nalon *et al.*, 2015].

As our calculus operates on a labelled clausal normal form that is closely linked to the tree model property of Kripke models for K_n, we briefly state the semantics of K_n. A *tree-like Kripke model M for n agents over* P is given by a tuple $(W, w_0, R_1, \ldots, R_n, \pi)$, where W is a set of possible *worlds* with a distinguished world w_0, each *accessibility relation* R_a is a binary relation on W such that their union is a tree with *root* w_0, and $\pi : W \to (\mathsf{P} \to \{\mathit{true}, \mathit{false}\})$ is a function which associates with each world $w \in W$ a truth-assignment to propositional symbols. Satisfaction of a formula at a world w of a model M is defined by:

- $\langle M, w\rangle \models p$ iff $\pi(w)(p) = \mathit{true}$, where $p \in \mathsf{P}$;
- $\langle M, w\rangle \models \neg\varphi$ iff $\langle M, w\rangle \not\models \varphi$;
- $\langle M, w\rangle \models (\varphi \wedge \psi)$ iff $\langle M, w\rangle \models \varphi$ and $\langle M, w\rangle \models \psi$;
- $\langle M, w\rangle \models \boxed{a}\varphi$ iff for all w', if wR_aw' then $\langle M, w'\rangle \models \varphi$.

Let $M = (W, w_0, R_1, \ldots, R_n, \pi)$ be a model. A formula φ is *locally satisfied in* M, denoted by $M \models_L \varphi$, if $\langle M, w_0\rangle \models \varphi$. The formula φ is *locally satisfiable* if there is a model M such that $\langle M, w_0\rangle \models \varphi$. A formula φ is *globally satisfied in* M, if for all $w \in W$, $\langle M, w\rangle \models \varphi$. We denote by $\mathsf{depth}(w)$ the length of the unique path from w_0 to w through the union of the accessibility relations in M. We call a *modal layer* the equivalence class of worlds at the same depth in a model.

We note that checking the local satisfiability of a formula φ can be reduced to the problem of checking the local satisfiability of its subformulae at the modal layer of a model which corresponds to the modal level where those subformulae occur (see [Areces *et al.*, 2000]). Also, checking the global satisfiability of φ can be reduced to checking the local satisfiability of φ at all modal layers (up to an exponential distance from the root) of a model [Goranko and Passy, 1992; Spaan, 1993]. Thus, an uniform approach based on modal levels can be used to deal with both problems, as given in the next section.

3 A Normal Form and Calculus for K_n

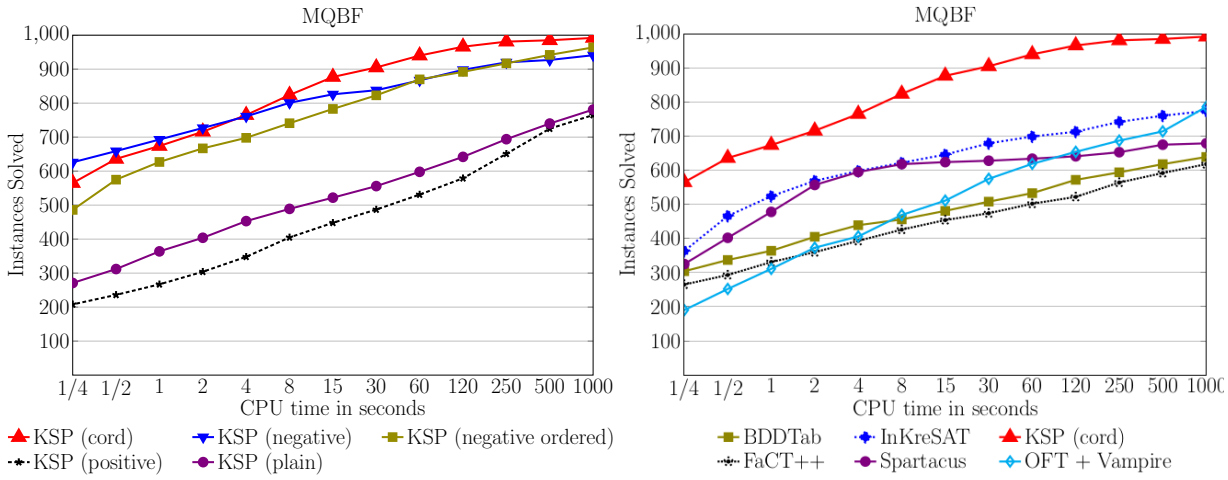
The calculus operates on labelled clauses of the form $ml : C$ (literal clause), $ml : l \Rightarrow \boxed{a} l'$ (positive a-clause), $ml : l \Rightarrow \diamondsuit_{\boxed{a}} l'$ (negative a-clause) where $ml \in \mathbb{N} \cup \{*\}$, C is a propositional clause, and l, l' are literals. The label ml indicates the depth of the Kripke model at which the clause is true. The special label $*$ is used if a clause is true at all depths/worlds in a model and it normally only occurs in the normal form if we want to check a formula for global satisfiability. Any formula of K_n can be transformed into an equi-satisfiable set of labelled clauses. The calculus uses a simple form of unification. The partial unification function σ on sets of labels is given by $\sigma(\{ml, *\}) = ml$; and $\sigma(\{ml\}) = ml$; otherwise, σ is undefined. The inference rules (Figure 1) can only be applied if the unification on labels is defined.

We show next an example of a refutation which uses both local and global reasoning (from [Nalon *et al.*, 2015], adapted from [Areces *et al.*, 1999]). Clauses 1 and 2 say that a person is either female or male. Clauses 3 and 4 say that tall people have children with blond hair. The particular situation of Tom, denoted here by t_0, is given in the following clauses. Clauses 5, 6, and 7 say that Tom's daughters are tall. Clauses 8 and 9 say that Tom has a grandchild who is not blond. We want to prove that Tom has a son, which appears negated in Clause 10. The refutation is given below.

1.	$*:$	$\mathit{female} \vee \mathit{male}$	
2.	$*:$	$\neg\mathit{female} \vee \neg\mathit{male}$	
3.	$*:$	$\neg\mathit{tall} \vee t_1$	
4.	$*:$	$t_1 \Rightarrow \boxed{c}\mathit{blond}$	
5.	$0:$	t_0	
6.	$0:$	$t_0 \Rightarrow \boxed{c} t_2$	
7.	$1:$	$\neg t_2 \vee \neg\mathit{female} \vee \mathit{tall}$	
8.	$0:$	$t_0 \Rightarrow \diamondsuit_{\boxed{c}} t_3$	
9.	$1:$	$t_3 \Rightarrow \diamondsuit_{\boxed{c}} \neg\mathit{blond}$	
10.	$0:$	$t_0 \Rightarrow \boxed{c} \neg\mathit{male}$	
11.	$1:$	$\neg t_1 \vee \neg t_3$	[MRES, 9, 4]
12.	$1:$	$\neg\mathit{tall} \vee \neg t_3$	[LRES, 11, 3]
13.	$1:$	$\neg t_3 \vee \neg t_2 \vee \neg\mathit{female}$	[LRES, 7, 12]
14.	$1:$	$\mathit{male} \vee \neg t_2 \vee \neg t_3$	[LRES, 13, 1]
15.	$0:$	$\neg t_0$	[GEN1, 10, 6, 8, 14]
16.	$0:$	**false**	[LRES, 15, 5]

4 KSP

KSP is an implementation, written in C, of the calculus in Figure 1. The main loop is based on the given-clause algorithm implemented in Otter [McCune, 2007], a variation of the set of support strategy [Wos *et al.*, 1965], a refinement



(a) KSP with different refinements

(b) All provers

Figure 2: Benchmarking results for MQBF

which restricts the set of choices of clauses participating in a derivation step. For the modal calculus, the set of clauses is partitioned according to the modal layer at which clauses are true. For each modal level ml, a literal clause is selected and the prover tries to resolve it with either a literal clause at the same modal level or with a set of modal clauses in the previous modal level, $ml - 1$.

The partitioning of the set of clauses avoids unnecessary applications of the resolution inference rules. For instance, the translation of the formula $\varphi = \Diamond_a \Diamond_a p \wedge \Box_a \neg p$ without considering the modal level at which its subformulae occur results in the set:

$$\begin{array}{ll} 1. & t_0 \\ 2. & t_0 \Rightarrow \Diamond_a t_1 \\ 3. & t_1 \Rightarrow \Diamond_a p \\ 4. & t_0 \Rightarrow \Box_a \neg p \end{array}$$

where t_0 and t_1 are new propositional symbols. Because $\Diamond_a p$ and $\Box_a \neg p$ are complementary, resolution applied to Clauses 3 and 4 results in $\neg t_0 \vee \neg t_1$, which cannot be used to find a contradiction. In contrast, by using labels, the translation of φ results in the set:

$$\begin{array}{lll} 1'. & 0: & t_0 \\ 2'. & 0: & t_0 \Rightarrow \Diamond_a t_1 \\ 3'. & 1: & t_1 \Rightarrow \Diamond_a p \\ 4'. & 0: & t_0 \Rightarrow \Box_a \neg p \end{array}$$

As the labels indicate, Clauses $3'$ and $4'$ occur at different modal levels and resolution is not applied to those clauses.

As another simple example, consider the formula $\Box_a \Box_a p \wedge \Diamond_a \Diamond_a \neg p$, whose layered normal form is given below.

$$\begin{array}{lll} 1. & 0: & t_0 \\ 2. & 0: & t_0 \Rightarrow \Box_a t_1 \\ 3. & 1: & t_1 \Rightarrow \Box_a p \\ 4. & 0: & t_0 \Rightarrow \Diamond_a t_2 \\ 5. & 1: & t_2 \Rightarrow \Diamond_a \neg p \end{array}$$

As clauses 3 and 4 are at the same modal level, MRES can be applied, resulting in $0 : \neg t_1 \vee \neg t_2$. By an application of GEN1 to this clause and Clauses 2 and 4, we obtain $0 : \neg t_0$. Applying LRES to $0 : \neg t_0$ and Clause 1, results in $0 :$ **false**, which shows that the original formula is unsatisfiable.

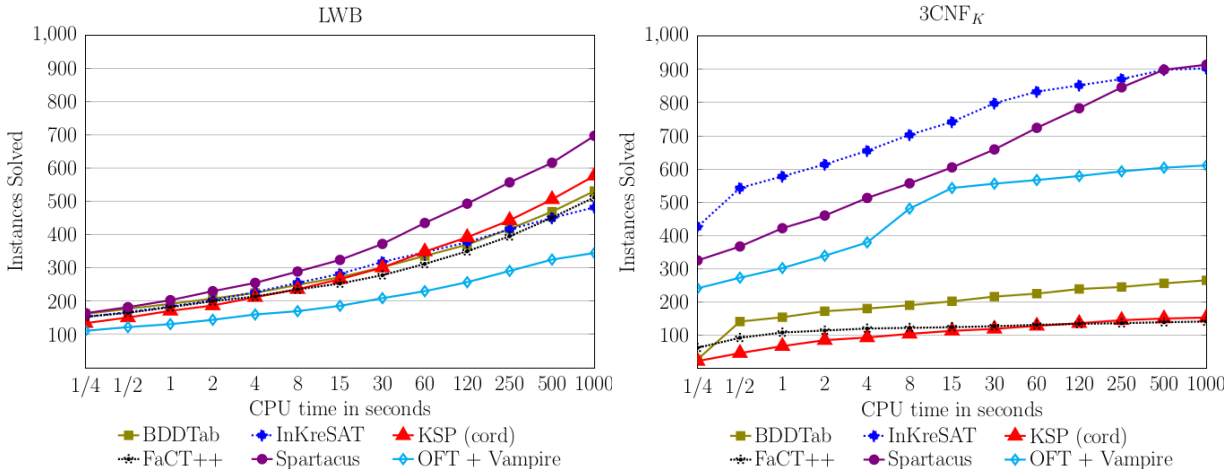
Besides the basic calculus with a set-of-support strategy, the user can further restrict LRES by choosing ordered (clauses can only be resolved on their maximal literals with respect to an ordering chosen by the prover in such a way to preserve completeness), negative (one of the premises is a negative clause, i.e. a clause where all literals are of the form $\neg p$ for some $p \in \mathsf{P}$), positive (one of the premises is a positive clause), or negative + ordered resolution (both negative and ordered resolution inferences are performed). The completeness of some of these refinements depends on the particular normal form chosen. The prover can either perform *local* or *global* reasoning. For a comprehensive description of KSP see [Nalon *et al.*, 2016a; Nalon *et al.*, 2016b].

5 Evaluation

We have compared KSP with BDDTab [Goré *et al.*, 2014], FaCT++ 1.6.3 [Tsarkov and Horrocks, 2006], InKreSAT 1.0 [Kaminski and Tebbi, 2013], Spartacus 1.0 [Götzmann *et al.*, 2010], and a combination of the optimised functional translation [Horrocks *et al.*, 2006] with Vampire 3.0 [Kovács and Voronkov, 2013]. Benchmarking was performed on PCs with an Intel i7-2600 CPU @3.40GHz and 16GB main memory. For each formula and each prover we have determined the median run time over five runs with a time limit of 1000 CPU seconds for each run. The experiments are fully described in [Nalon *et al.*, 2016a], from where the graphs shown in this section are taken.

Our benchmarks [Nalon *et al.*, 2016b] consist of three collections of modal formulae:

1. The complete set of TANCS-2000 modalised random QBF (MQBF) formulae [Massacci and Donini, 2000] complemented by the additional MQBF formulae provided by



(a) Results for LWB

(b) Results for 3CNF$_K$

Figure 3: Benchmarking results for LWB and 3CNF$_K$

Tebbi and Kaminski [2013]. This collection consists of five classes, with a total of 1016 formulae, of which 617 are known to be satisfiable and 399 are known to be unsatisfiable (due to the output of one of the provers).

2. Logics Workbench (LWB) basic modal logic benchmark formulae [Balsiger *et al.*, 2000], with 56 formulae chosen from each of the 18 parameterised classes. The median value of the maximal parameter value used for the 18 classes is 1880, far beyond what has ever been tested before. Of the resulting 1008 formulae, half are satisfiable and half are unsatisfiable by construction of the benchmark classes.
3. Randomly generated 3CNF$_K$ formulae [Patel-Schneider and Sebastiani, 2003] over 3 to 10 propositional symbols with modal depth 1 or 2. The formulae were chosen from each of the 11 parameter settings given in [Patel-Schneider and Sebastiani, 2003, page 372]. The resulting collection contains 1000 formulae, of which 457 are known to be satisfiable and 464 are known to be unsatisfiable.

Figure 2a compares the impact of different refinements on the performance of KSP on the MQBF collection. With *plain* KSP uses the rules shown in Figure 1, without additional refinement. With *cord* and *negative ordered*, KSP applies ordered resolution and negative + ordered resolution, respectively. The configuration *negative* (resp. *positive*) uses negative (resp. positive) resolution on a set of clauses. For all configurations, the same preprocessing techniques are used. The *cord* configuration offers the best performance. Ordered resolution restricts the applicability of the rules further than the other refinements. Not only is this an advantage on satisfiable formulae in that a saturation can be found more quickly, but also on unsatisfiable formulae where with this refinement KSP finds refutations much more quickly than with any of the other refinements.

Figure 2b compares the performance of all the provers on the MQBF collection. It shows that KSP performs better than any of the other provers. The classes of formulae on which KSP performs best are those where the number of propositional symbols is small and/or those symbols are uniformly distributed over a wide range of modal levels. This reduces the possibility of inference steps between modal clauses with complementary literals.

Figure 3 shows the benchmarking results on the LWB and 3CNF$_K$ collections. On the LWB collection KSP performs about as well as BDDTab, FaCT++ and InKreSAT, while Spartacus performs best and the combination of the optimised functional translation with Vampire (OFT + Vampire) performs worst. A characteristic of the classes on which KSP performs best is again that atomic subformulae are evenly spread over a wide range of modal levels.

Finally, on the 3CNF$_K$ collection, InKreSAT is the best performing prover and KSP the worst one. This should now not come as a surprise. For 3CNF$_K$ we specifically restricted ourselves to formulae with low modal depth which in turn means that the layered normal form has little positive effect.

6 Conclusions and Future Work

KSP implements a variation of the set-of-support strategy by restricting further the application of inference rules to clauses whose literals are at the same modal level. The evaluation indicates that KSP works well on problems with high modal depth where the separation of modal layers can be exploited to improve the efficiency of reasoning.

The prover implements both local and global reasoning, and the implementation of the local satisfiability under global constraint procedure is ongoing work. We are are also working on the extension of the calculus for dealing with different modal logics. The calculi for T_n and $\mathsf{4}_n$, which can be combined to provide a calculus for $\mathsf{S4}_n$, have already been implemented.

Several refinements and redundancy elimination techniques have been implemented. The choice of strategies and optimisations are, by now, all left to the user. The development of an "auto mode" in which the prover makes a choice of its own, based on an analysis of the given formula, is left for future work.

References

[Areces *et al.*, 1999] Carlos Areces, Hans de Nivelle, and Maarten de Rijke. Prefixed Resolution: A Resolution Method for Modal and Description Logics. In *Proc. of CADE-16*, volume 1632 of *LNAI*, pages 187–201. Springer, 1999.

[Areces *et al.*, 2000] Carlos Areces, Rosella Gennari, Juan Heguiabehere, and Maarten de Rijke. Tree-based heuristics in modal theorem proving. In *Proc. of ECAI 2000*, pages 199–203. IOS Press, 2000.

[Balsiger *et al.*, 2000] Peter Balsiger, Alain Heuerding, and Stefan Schwendimann. A benchmark method for the propositional modal logics K, KT, S4. *Journal of Automated Reasoning*, 24(3):297–317, 2000.

[Goranko and Passy, 1992] Valentin Goranko and Solomon Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation*, 2(1):5–30, 1992.

[Goré *et al.*, 2014] Rajeev Goré, Kerry Olesen, and Jimmy Thomson. Implementing tableau calculi using BDDs: BDDTab system description. In *Proc. of IJCAR 2014*, volume 8562 of *LNCS*, pages 337–343. Springer, 2014.

[Götzmann *et al.*, 2010] Daniel Götzmann, Mark Kaminski, and Gert Smolka. Spartacus: A tableau prover for hybrid logic. *ENTCS*, 262:127–139, 2010.

[Hailpern, 1982] Brent T. Hailpern. *Verifying Concurrent Processes Using Temporal Logic*, volume 129 of *LNCS*. Springer, 1982.

[Halpern and Moses, 1992] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, April 1992.

[Halpern *et al.*, 1983] Joseph Y. Halpern, Zohar Manna, and Ben Moszkowski. A Hardware Semantics Based on Temporal Intervals. In *Proc. of ICALP 1983*, volume 154, pages 278–291, 1983.

[Halpern, 1987] Joseph Y. Halpern. Using Reasoning about Knowledge to Analyze Distributed Systems. *Annual Review of Computer Science*, 2, 1987.

[Horrocks *et al.*, 2006] Ian R. Horrocks, Ullrich Hustadt, Ulrike Sattler, and Renate Schmidt. Computational modal logic. In *Handbook of Modal Logic*, pages 181–245. Elsevier, 2006.

[Kaminski and Tebbi, 2013] Mark Kaminski and Tobias Tebbi. InKreSAT: Modal reasoning via incremental reduction to SAT. In *Proc. of CADE-24*, volume 7898 of *LNAI*, pages 436–442. Springer, 2013.

[Kovács and Voronkov, 2013] Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In *Proc. of CAV 2013*, volume 8044 of *LNCS*, pages 1–35. Springer, 2013.

[Massacci and Donini, 2000] Fabio Massacci and Francesco M. Donini. Design and results of TANCS-2000 non-classical (modal) systems comparison. In *Proc. of TABLEAUX 2000*, volume 1847 of *LNCS*, pages 52–56. Springer, 2000.

[McCune, 2007] William W. McCune. OTTER 3.0 reference manual and guide, May 07 2007.

[Nalon *et al.*, 2015] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. A modal-layered resolution calculus for K. In *Proc. of TABLEAUX 2015*, volume 9323 of *LNCS*, pages 185–200. Springer, 2015.

[Nalon *et al.*, 2016a] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. KSP: A resolution-based prover for multimodal K. In *Proc. of IJCAR 2016*, volume 9706 of *LNCS*, pages 406–415. Springer, 2016.

[Nalon *et al.*, 2016b] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. KSP: sources and benchmarks. `http://www.cic.unb.br/~nalon/#software`, 2016.

[Patel-Schneider and Sebastiani, 2003] Peter F. Patel-Schneider and Roberto Sebastiani. A new general method to generate random modal formulae for testing decision procedures. *Journal of Artificial Intelligence Research (JAIR)*, 18:351–389, 2003.

[Pratt, 1980] Vaughan R. Pratt. Application of modal logic to programming. *Studia Logica*, 39(2/3):257–274, 1980.

[Rao and Georgeff, 1991] Anand S. Rao and Michael P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In *Proc. of KR&R-91*, pages 473–484, Cambridge, MA, USA, April 1991. Morgan-Kaufmann.

[Robinson, 1965] John A. Robinson. A Machine–Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[Schild, 1991] Klaus Schild. A Correspondence Theory for Terminological Logics. In *Proc. of IJCAI'91*, pages 466–471. Springer, 1991.

[Schulz and Möhrmann, 2016] Stephan Schulz and Martin Möhrmann. Performance of clause selection heuristics for saturation-based theorem proving. In *Proc. of IJCAR 2016*, volume 9706 of *LNCS*, pages 330–345. Springer, 2016.

[Spaan, 1993] Edith Spaan. *Complexity of Modal Logics*. PhD thesis, University of Amsterdam, 1993.

[Tsarkov and Horrocks, 2006] Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of IJCAR 2006*, volume 4130 of *LNCS*, pages 292–297. Springer, 2006.

[Wos *et al.*, 1965] Lawrence Wos, George A. Robinson, and Daniel F. Carson. Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. *Journal of the ACM*, 12:536–541, 1965.