On evaluating decision procedures for modal logic

Ullrich Hustadt and **Renate A. Schmidt**
Max-Planck-Institut fur Informatik, 66123 Saarbriicken, Germany
{hustadt, schmidt} topi-sb.mpg.de

Abstract

This paper investigates the evaluation method of decision procedures for multi-modal logic proposed by Giunchiglia and Sebastiani as an adaptation from the evaluation method of Mitchell *et al* of decision procedures for propositional logic. We compare three different theorem proving approaches, namely the Davis-Putnam-based procedure KSAT, the tableaux-based system *KTUS* and a translation approach combined with first-order resolution. Our results do not support the claims of Giunchiglia and Sebastiani concerning the computational superiority of KSAT over KRIS*,* and an easy-hard-easy pattern for randomly generated modal formulae.

1 Introduction

There are a variety of automated reasoning approaches for the basic propositional multi-modal logic *K(m)* and its syntactical variant, the knowledge representation formalism *ALC.* Some approaches utilize standard first-order theorem proving techniques in combination with translations from propositional modal logic to first-order logic [Ohlbach and Schmidt, 1995]. Others use Gentzen systems [Goble, 1974]. Still others use tableaux proof methods [Baader and Hollunder, 1991].

Usually, the literature on theorem provers for modal logic confines, itself to a description of the underlying calculus and methodology accompanied with a consideration of the worst-case complexity of the resulting algorithm. Sometimes a small collection of benchmarks is given as in [Catach, 1991]. However, there have not been any exhaustive empirical evaluations or comparisons of the computational behavior of theorem provers based on different methodologies.

Giunchiglia and Sebastiani [1996a; 1996b] changed that. They report on an exhaustive empirical analysis of a new theorem prover, called KSAT, and the tableaux system *KTUS.* KSAT is an adaptation for the multi-modal logic *K(m)* of a SAT-procedure for checking satisfiability in propositional logic. The evaluation of Giunchiglia and Sebastiani has some shortcomings which we address. The random generator used to set up a benchmark suite produces formulae containing a substantial amount of tautologous and contradictory subformulae. It favours the SAT-procedure KSAT which utilizes a preprocessing routine that eliminates trivial tautologies and contradictions from the formulae. This property of the random formulae mislead Giunchiglia and Sebastiani in their analysis and comparison of KSAT and *KTUS.* We show the random generator does not produce challenging unsatisfiable modal formulae.

The paper is structured as follows. In Sections 2, 3 and 4 we briefly describe the inference mechanisms of KSAT, *KTUS* and the translation approach. Section 5 describes the evaluation method of Giunchiglia and Sebastiani. The main part is Section 6 which evaluates the test method.

2 The SAT-based procedure KSAT

By definition, a *formula* of the multi-modal logic $K(m)$, where m is a natural number, is a boolean combination of propositional and modal atoms. A *modal atom* is an expression of the form $\Box_i\psi$, where i is such that $1{\leq}i{\leq}m$ and ψ is a formula of $K(m)$. $\Diamond_i\psi$ is an abbreviation for $\neg\Box_i\neg\psi$. The semantics of $K(m)$ is given by the usual Kripke semantics.

KSAT tests the satisfiability of a given formula ϕ of $K(m)$. Its basic algorithm, called KSAT0, is based on the following two procedures:

KDP: Given a modal formula ϕ, this procedure generates a truth assignment μ for the propositional and modal atoms in ϕ which renders ϕ true propositionally. This is done using a decision procedure for propositional logic.

KM: For a given ϕ and μ computed by KDP, let $\Box_i\psi_{ij}$ denote any modal atom in ϕ with $\mu(\Box_i\psi_{ij}) = \bot$ and $\Box_i\phi_{ik}$ any modal atom with $\mu(\Box_i\phi_{ik}) = \top$. The procedure checks for each index i, $1{\leq}i{\leq}m$, and each j whether the formula $\varphi_{ij} = \bigwedge_k \phi_{ik} \wedge \neg\psi_{ij}$ is satisfiable. This is done with KDP. If at least one of the formulae φ_{ij} is not satisfiable, then KM *fails on* μ, otherwise it *succeeds*.

KSAT0 starts by generating a partial truth assignment μ for ϕ using KDP. If KM succeeds on μ, then ϕ is $K(m)$-satisfiable. Otherwise, we have to generate a new truth assignment for ϕ using KDP. If no further truth assignment is found, then ϕ is $K(m)$-unsatisfiable.

The decision procedure KDP for propositional logic can be described by a set of transition rules on ordered pairs $P \triangleright S$ where P is a sequence of pairs $\langle\phi, \mu\rangle$, and S is a set of satisfying truth assignments.

$$\texttt{dp\_sol:}\quad \frac{\langle\top, \mu\rangle \mid P \triangleright S}{P \triangleright S \cup \{\mu\}}$$

$$\texttt{dp\_clash:}\quad \frac{\langle\bot, \mu\rangle \mid P \triangleright S}{P \triangleright S}$$

$$\texttt{dp\_unit:}\quad \frac{\langle\phi[c], \mu\rangle \mid P \triangleright S}{\langle\phi', \mu \cup \{c = \top\}\rangle \mid P \triangleright S}$$

if c is a unit clause in ϕ and ϕ' is the result of replacing all occurrences of c and $\bar{c}$ by $\top$ and $\bot$, respectively, followed by boolean simplification.

$$\texttt{dp\_split:}\quad \frac{\langle\phi[m], \mu\rangle \mid P \triangleright S}{\langle\phi[m] \wedge p, \mu\rangle \mid \langle\phi[m] \wedge \neg p, \mu\rangle \mid P \triangleright S}$$

if dp_unit cannot be applied to $\langle\phi[m], \mu\rangle$, and m is a propositional or modal atom.

The symbol $\mid$ denotes concatenation. $\overline{\phi}$ and ϕ are complementary, e.g. $\overline{\neg p} = p$ and $\overline{\Box_i p} = \Diamond_i \neg p$.

Starting with $\langle\phi, \emptyset\rangle \triangleright \emptyset$, exhaustively applying the inference rules will result in $\emptyset \triangleright S$ where S is a complete set of partial truth assignments making ϕ true.

Note that the transition rules form a variant of the Davis-Putnam procedure for propositional formulae not in conjunctive normal form. The crucial nondeterminism of the procedure is the selection of the splitting 'variable' m in the transition rule dp_split. KSAT employs the heuristic that selects an atom with a maximal number of occurrences in .

3 The tableaux-based system *ICTUS*

While KSAT abstracts from the modal part of formulae to employ decision procedures for propositional logic, *KRIS* manipulates modal formulae (in variant notation) directly. More precisely, the inference rules of *KRIS* are relations on sequences of sets of labeled modal formulae of the form *w:ip** where *w* is a label chosen from a countably infinite set of labels Γ and ψ is modal formula. For improved readability we write $w{:}\psi, C$ instead of $\{w{:}\psi\} \cup C$.

$$\bot\texttt{-elim:}\quad \frac{w{:}\bot, C \mid S}{S}$$

$$\top\texttt{-elim:}\quad \frac{w{:}\top, C \mid S}{C \mid S}$$

$$\wedge\texttt{-clash:}\quad \frac{w{:}\phi, w{:}\overline{\phi}, C \mid S}{S}$$

$$\wedge\texttt{-elim:}\quad \frac{w{:}\phi \wedge \psi, C \mid S}{w{:}\phi, w{:}\psi, C \mid S}$$

$$\vee\texttt{-elim:}\quad \frac{w{:}\phi \vee \psi, C \mid S}{w{:}\phi, C \mid w{:}\psi, C \mid S}$$

if $w{:}\phi \vee \psi, C$ has been simplified by

$\vee\texttt{-simp}_0$: $w{:}\phi \vee \psi, w{:}\phi, D \rightarrow w{:}\phi, D$

$\vee\texttt{-simp}_1$: $w{:}\phi \vee \psi, w{:}\overline{\phi}, D \rightarrow w{:}\psi, w{:}\overline{\phi}, D.$

$$\Diamond_i\texttt{-elim:}\quad \frac{w{:}\Diamond_i\phi, D, C \mid S}{v{:}\phi \wedge \psi_1 \wedge \ldots \wedge \psi_n, D, C \mid S}$$

if $D = w{:}\Box_i\psi_1, \ldots, w{:}\Box_i\psi_n$, C does not contain any $w{:}\Box_i\psi$, none of the other rules can be applied to C, and v is a new label from Γ.

The application of the $\vee$-elim rule to any labeled formula $w{:}\phi \vee \psi$ is preceeded by the application of the simplification rules to that formula. In no other situation the simplification rules are invoked. Given a formula ϕ, the input sequence for $\mathcal{KRIS}$ is $w_0{:}\phi'$, where w_0 is a new label chosen from Γ and ϕ' is the modal negation normal form of ϕ. If $\mathcal{KRIS}$ arrives at a sequence $C \mid S$ such that no transformation rule can be applied to C, then the original formula ϕ is satisfiable. Otherwise the transformation rules will eventually reduce $w_0{:}\phi'$ to the empty sequence and ϕ is unsatisfiable.

4 The translation approach

The *translation approach* (TA) is based on the idea that modal inference can be done by translating modal formulae into first-order logic and conventional theorem proving. We use the *optimised functional translation* approach of Ohlbach and Schmidt [1995]. It has the property that ordinary resolution without any refinement strategies is a decision procedure for *K(m)* [Schmidt, 1997]. The translation maps modal formulae into a logic, called *basic path logic,* which is a monadic fragment of sorted first-order logic with one binary function symbol o that defines accessibility. A formula of path logic is further restricted in that its clausal form may only contain Skolem terms that are constants.

The optimised functional translation does a sequence of transformations. The first transformation Π_f maps a modal formula ϕ to its so-called functional translation defined by $\Pi_f(\phi) = \forall x\, \pi_f(\phi, x)$. For $K(m)$, π_f is defined by

$$\begin{aligned} \pi_f(p, s) &= P(s) \\ \pi_f(\Box_i\phi, s) &= \mathit{def}_i(s) \rightarrow \forall\alpha_i\, \pi_f(\phi, s \circ \alpha_i). \end{aligned}$$

p is a propositional variable and P is a unary predicate uniquely associated with p, def_i is a special unary predicate of sort i, and α_i denotes a variable of sort i. For the propositional connectives π_f is a homomorphism. The second transformation applies the so-called quantifier exchange operator Υ which moves existential quantifiers inwards over universal quantifiers using the rule '$\exists\alpha\forall\beta\,\psi$ becomes $\forall\beta\exists\alpha\,\psi$'. Ohlbach and Schmidt prove $\Upsilon\Pi_f$ preserves satisfiability.

Our aim is to test the satisfiability of a given modal formula ϕ. This can be achieved by testing the satisfiability of the clausal form of $\neg\Upsilon\Pi_f(\neg\phi)$. The theorem prover we use is SPASS Version 0.55 developed by Weidenbach *et al.* [1996] which is an ordered resolution-based theorem prover for sorted first-order logic.

5 The test method

The evaluation method adopted by Giunchiglia and Sebastiani follows the approach of Mitchell *et al* [1992]. To set up a benchmark suite for Davis-Putnam-based theorem provers Mitchell *et al* generate propositional formulae using the fixed clause-length model. Giunchiglia and Sebastiani modify this approach for *K(m).*

There are five parameters: the number of propositional variables *N,* the number of modalities M, the number of modal subformulae per disjunction *K,* the number of modal subformulae per conjunction L, the modal degree 25, and the probability P. Based on a given choice of parameters random modal -RTCNF formulae are defined inductively as follows. A *random (modal) atom* of degree 0 is a variable randomly chosen from the set of *N* propositional variables. A *random modal atom* of degree D, *D>0,* is with probability *P* a random modal atom of degree 0 or an expression of the form □Φ, otherwise, where □» is a modality randomly chosen form the set of *M* modalities and *Φ* is a random modal if CNF clause of modal degree D-l (defined below). A *random modal literal* (of degree *D)* is with probability 0.5 a random modal atom (of degree *D)* or its negation, otherwise. A *random modal KCNF clause* (of degree *D)* is a disjunction of *K* random modal literals (of degree *D).* Now, a *random modal KCNF formula* (of degree *D)* is a conjunction of *L* random modal KCNF clauses (of degree *D).*

For the comparison of the performance of KSAT and KRI*S,* Giunchiglia and Sebastiani proceed as follows. They fix all parameters except L, the number of clauses. For example, they choose N=3, M=I, K=3, D=5, and P=0.5. The parameter *L* ranges from *N* to 40N. For each value of the ratio *L/N* a set of 100 random modal KCNF formulae of degree *D* is generated. We will see that for small *L* the generated formulae are most likely to be satisfiable and for larger *L* the generated formulae are most likely to be unsatisfiable. For each generated formula *<p* they measure the time needed by one of the decision procedures to determine the satisfiability of Φ. Since checking a single formula can take arbitrarily long in the worst case, there is an upper limit for the CPU time consumed. As soon as the upper limit is reached, the computation for *Φ* is stopped. If the computation for more than 50% of the formulae of a set associated with a value of *L* has been abandoned, then the computation for the set is discontinued. Now, the median CPU time over the ratio *L/N* is presented. For example, the graphs of Figure 1 show the performance of *KRIS* and KSAT on the parameter settings PSO (N=3, M=I, K=3, D=5) and PS1 (N=5, M=I, K=3, *D=2).* Our tests have been run on a Sun Ultra 1/170E with 196MB main memory using a time-limit of 1000 CPU seconds. Altogether Giunchiglia and Sebastiani [1996b] present graphs for ten different parameter settings. Based on their graphs including Figure 1 they come to the following conclusions:

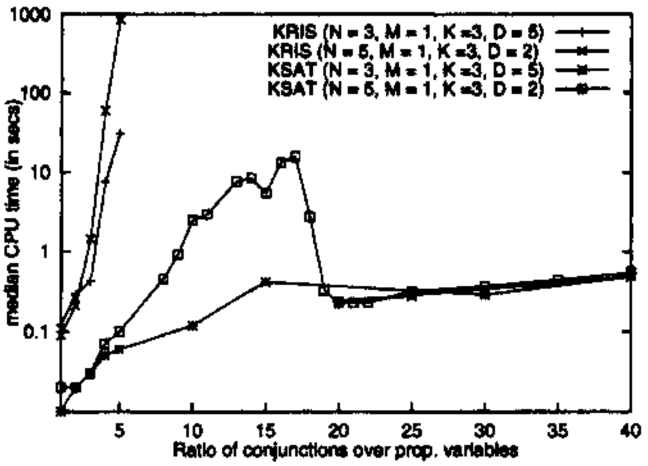

Figure 1: Performance comparison of $\mathcal{KRIS}$ and KSAT

(1) KSAT outperforms by orders of magnitude the previous state-of-the art decision procedures.

(2) All SAT-based modal decision procedures are intrinsically bound to be more efficient than tableaux-based decision procedures.

(3) There is partial evidence of an easy-hard-easy pattern on randomly generated modal logic formulae independent of all the parameters of evaluation considered.

We show that the situation is more complex and does not justify such strong claims. For our analysis it suffices to focus on the settings PSO and PS1 of Figure 1.

6 Analysis of the test method

Selecting good test instances is crucial when evaluating and comparing the performance of algorithms empirically. We address the question whether the random generator and the parameter settings chosen by Giunchiglia and Sebastiani [1996b; 1996a] are appropriate for this purpose and actually support claims (1) to (3).

It is important to note that the claim of Giunchiglia and Sebastiani [1996b, p. 307] that for *D=0* random modal 3CNF formulae coincide with random 3SAT

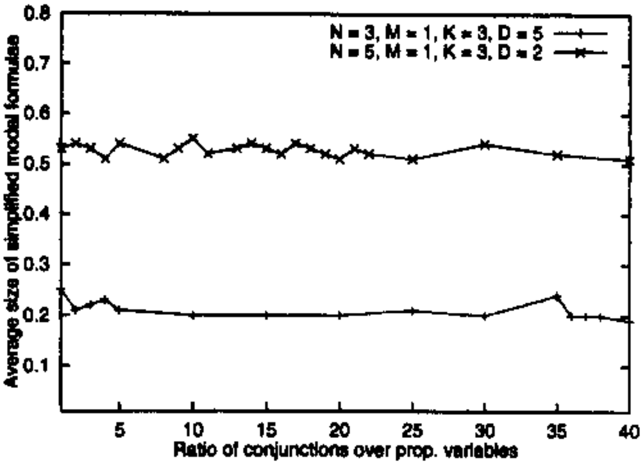

Figure 2: Effect of simplifying modal 3CNF formulae

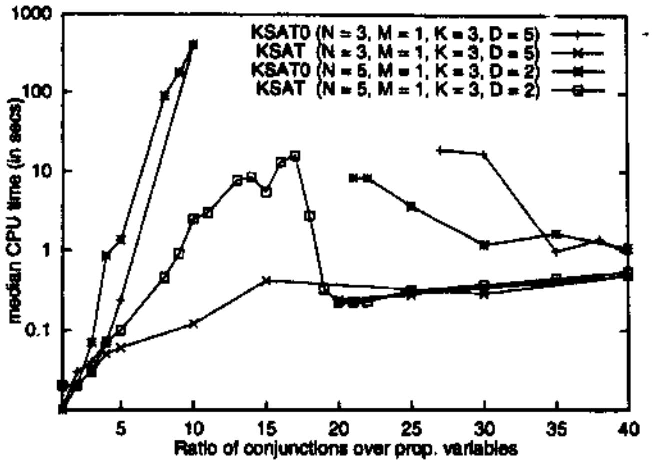

Figure 3: Performance comparison of KSAT and KSAT0

clauses as defined in Mitchell *et al.* is wrong: To generate a random 3SAT clause we have to randomly generate a *set* of three propositional variables and negate each member of the set with probability 0.5. In contrast, to generate a random modal 3CNF clause of degree 0, we have to randomly generate a *multiset* of three propositional variables and negate each member of the multiset with probability 0.5. For example, $p \vee q \vee \neg r$ is a 3SAT clause and also a modal 3CNF clause of degree 0. The clauses $p \vee \neg p \vee p$ and $p \vee p \vee q$ are not random 3SAT clauses, but both are random modal 3CNF clauses of degree 0. As we move to random modal 3CNF formulae of higher degree, such clauses may occur within the scope of a modal operator. For example, expressions like $\neg\Box_1(p \vee \neg p \vee p)$ may occur which are contradictory. Consequently, random modal 3CNF formulae contain tautological and contradictory subformulae. It is straightforward to remove these subformulae without affecting satisfiability. The extent to which the size of the random modal 3CNF formulae can be reduced by such simplifications is reflected by the graphs of Figure 2. They depict the average ratio of the size of the simplified random modal 3CNF formulae over the size of the original formulae. For the random modal 3CNF formulae generated using three propositional variables only, on average, the size of a simplified formula is only 1/4 of the size of the original formula. For the second parameter setting we see a reduction to 1/2 of the original size. In other words, one half to three quarters of the random modal 3CNF formulae is "logical garbage" that can be eliminated at little cost.

KSAT utilizes a form of preprocessing that removes duplicate and contradictory subformulae of an input formula. That is, KSAT performs exactly the simplification whose effect we have just described. *ICTUS* on the other hand does not perform a similar simplification. We consider how KSAT performs if we remove the preprocessing step from its code. In Figure 3 KSATO denotes this modified form of KSAT, since it is actually identical to the algorithm described in Section 2. We see that the behavior of KSATO differs from the behavior of KSAT by orders of magnitude. Since the preprocessing is not an intrinsic part of the decision procedures, for the comparison of the procedures, either both KSAT and *ICTUS* should utilize the preprocessing or none of them should. Simplification of the generated formulae is reasonable, so we have added the preprocessing function to KRI*S.* This modified version of *ICTUS* will be denoted by *KRIS**. The graphs in Figure 4 show the performance of KSAT and *ICTUS**. Although the performance of KSAT is still better than that of *KRIS*,* KSAT is no longer qualitatively better than *ICTUS* with preprocessing.

We now address claim (2) that, intrinsically, SAT-based modal decision procedures are bound to be more efficient than tableaux-based decision procedures. The explanation is based on the work by D'Agostino [1992], who shows that in the worst case algorithms using the

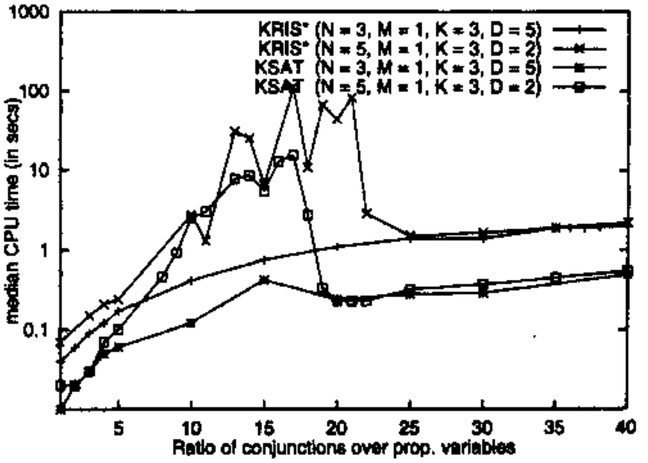

Figure 4: The performance of KSAT and *KRIS**

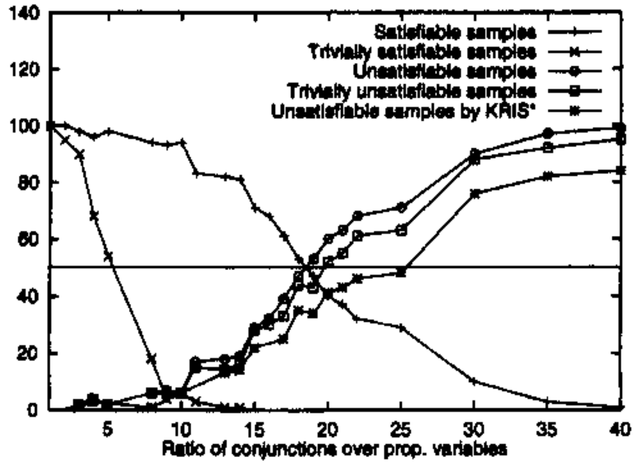

Figure 5: Quality of the test set **PS1**

$\vee$-elim rule cannot simulate truth tables in polynomial time. Instead one has to replace $\vee$-elim by the rule

$$\vee\text{-elim'}: \quad \frac{w{:}\phi \vee \psi, C \mid S}{w{:}\phi, C \mid w{:}\psi, w{:}\neg\phi, C \mid S}$$

This rule ensures that the two generated subproblems $w{:}\phi, C$ and $w{:}\psi, w{:}\neg\phi, C$ are mutually exclusive.

We have just seen that a major cause of the difference in computational behavior of the two algorithms is actually the absence of the preprocessing step in KRIS. To explain the remaining difference we study the quality of the random modal 3CNF formulae. Suppose that we want to test a random modal 3CNF formula Φ generated using N propositional variables for satisfiability in a Kripke model with only one world. We have to test at most 2^N truth assignments to the propositional variables. Since N<5 for the modal formulae under consideration, this is a trivial task, even if we use the truth table method. We say a random modal 3CNF formula Φ is *trivially satisfiable* if Φ is satisfiable in a Kripke model with only one world. We also say a random modal 3CNF formula Φ is *trivially unsatisfiable* if the conjunction of the purely propositional clauses of Φ is unsatisfiable. Again, testing whether Φ is trivially unsatisfiable requires the consideration of 2^N truth assignments only.

The graphs,of Figure 5 show the percentage of satisfiable, trivially satisfiable, unsatisfiable, trivially unsatisfiable, and unsatisfiable samples detected by *ICTUS** for the parameter setting **PS1.** We see that almost all unsatisfiable formulae are trivially unsatisfiable. We have verified that this also holds for all the other parameter settings used by Giunchiglia and Sebastiani. This indicates, none of these parameter settings is suited to generate challenging unsatisfiable modal formulae.

If we consider Figure 4 and 5 together, for ratios *L/N* between 19 and 21 we observe the graph of KRIS* deviates a lot (by a factor of more than 100) from the graph of KSAT. This is the area near the crossover point where the percentage of trivially unsatisfiable formulae rises above 50%, however, the percentage of unsatisfiable formulae detected by *KRIS** is still below 50% in this area. *ICTUS** does not detect all trivially unsatisfiable formulae within the time-limit which explains the deviation in performance from KSAT. The reason for *ICTUS** not detecting all trivially unsatisfiable formulae within the time limit, can be illustrated by the following example. Let Φ be a simplified modal 3CNF formula

$$p \wedge q \wedge (m_{11} \vee m_{12} \vee m_{13}) \\ \dots \\ \wedge (m_{n1} \vee m_{n2} \vee m_{n3}) \wedge (\neg p \vee \neg q)$$

where the m_{ij}, with $1 \leq i \leq n$, $1 \leq j \leq 3$, are modal literals different from p, q, $\neg p$, and $\neg q$. Evidently, ϕ is trivially unsatisfiable. KSAT does the following: Since p and q are unit clauses in ϕ, it applies the rule dp_unit twice to ϕ. The rule replaces the occurrences of p and q by $\top$, it replaces the occurrences of $\neg p$ and $\neg q$ by $\bot$, and it simplifies the formula. The resulting formula is $\bot$. At this point only the rule dp_clash is applicable and KSAT detects that ϕ is unsatisfiable. In contrast, $\mathcal{KRIS}^*$ proceeds as follows. First it applies the $\wedge$-elim rule $n+2$ times, eliminating all occurrences of the $\wedge$ operator. Then it applies the $\vee$-elim rule to all disjunctions, starting with $(m_{11} \vee m_{12} \vee m_{13})$ and ending with $(m_{n1} \vee m_{n2} \vee m_{n3})$. This generates 3^n subproblems. Each of these subproblems contains the literals p and q and the disjunction $\neg p \vee \neg q$. The rule $\vee$-simp$_1$ eliminates $\neg p \vee \neg q$ and a final application of the $\wedge$-clash rule exhibits the unsatisfiability of each subproblem. Obviously, for n large enough, $\mathcal{KRIS}^*$ will not be able to finish this computation within the time-limit.

Note, it makes no difference whether $\mathcal{KRIS}^*$ eliminates disjunctions by the $\vee$-elim rule or the $\vee$-elim' rule. The reason for $\mathcal{KRIS}^*$ not finishing within the time-limit is that it does not apply the simplification rules $\vee$-simp$_1$ and $\vee$-simp$_2$ and the $\wedge$-clash rule eagerly before any application of the $\vee$-elim rule.

Finally, we consider claim (3) conjecturing an easy-hard-easy pattern, independent of all the parameters of evaluation, in randomly generated modal logic formulae. We have seen in Figure 1 that the mean CPU time consumption of KSAT decreases drastically at the ratio *L/N*—17.5 for the second sample. This is almost the point, where 50% of the sample formulae are satisfiable. This decline resembles the behavior of propositional SAT decision procedures on randomly generated 3SAT problems. Figure 6 compares the performance of KSAT with the performance of the translation approach on two parameter settings, where the easy-hard-easy pattern is most visible for KSAT. The translation approach does not show the peaking behavior of KSAT. The median CPU time grows monotonically with the size of

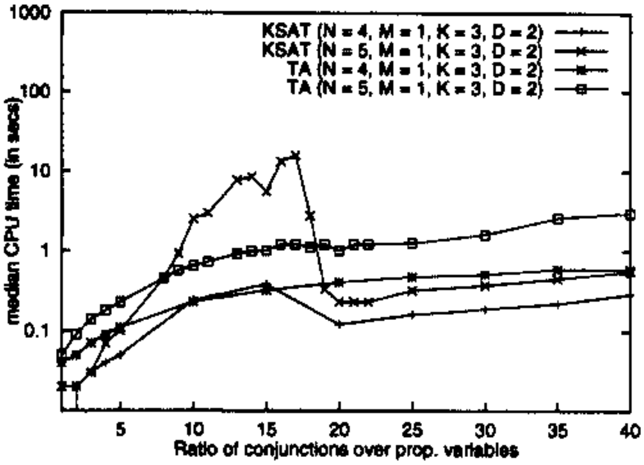

Figure 6: Performance comparison of KSAT and TA

modal formulae. Thus, the phase transition observed by Giunchiglia and Sebastiani is an artificial phenomenon of KSAT (and *KRIS),* and not an intrinsic property of the generated modal formulae.

Observe that the peaking behavior occurs in the area where the number of trivially satisfiable sample formulae approaches zero. The following example tries to explain this. Let be a simplified modal 3CNF formula of the form

$$\neg\Box_1 s \wedge \Box_1(p \vee r) \wedge (\Box_1\neg r \vee \Box_1 q) \wedge (\neg\Box_1 p \vee \Box_1 r)$$
$$\wedge\, (m_{11} \vee m_{12} \vee m_{13}) \wedge \ldots \wedge (m_{n1} \vee m_{n2} \vee m_{n3})$$

where the m_{ij}, with $1{\leq}i{\leq}n$, $1{\leq}j{\leq}3$, are modal literals different from the modal literals in the first three conjunctions of ψ. Assume, ψ is satisfiable. $\Box_1\neg r$ is false in any model of ψ. In the situation that $\Box_1\neg r$ is one of the first split literals chosen by KSAT, it generates a huge search tree without finding a satisfying truth assignment before it eventually turns to the case where $\Box_1\neg r$ is assigned $\perp$. This explains the bad behaviour of KSAT on those sample formulae where satisfiability tests in the non-propositional contexts are essential. KRIS* behaves similarly.

In contrast, the translation approach proceeds as follows. It generates a clause set for ψ containing

$$\begin{array}{l}
\quad def_1(\underline{\iota}) \\
\neg S(\underline{\iota} \circ \underline{a}) \\
\neg def_1(\underline{\iota}) \vee P(\underline{\iota} \circ x) \vee R(\underline{\iota} \circ x), \\
\neg def_1(\underline{\iota}) \vee \neg R(\underline{\iota} \circ x) \vee \neg def_1(\underline{\iota}) \vee Q(\underline{\iota} \circ y), \\
\neg P(\underline{\iota} \circ \underline{b}) \vee \neg def_1(\underline{\iota}) \vee R(\underline{\iota} \circ x)
\end{array}$$

where I, *a* and B denote Skolem constants and *x* and *y* are variables. SPASS applies unit propagation to the first clause followed by subsumption. Three resolvents can be derived: $P(\underline{\iota} \circ x) \vee Q(\underline{\iota} \circ y)$, $\neg P(\underline{\iota} \circ \underline{b}) \vee Q(\underline{\iota} \circ y)$, and $R(\underline{\iota} \circ \underline{b}) \vee R(\underline{\iota} \circ x)$. A factoring step on the last resolvent yields *R(t o b),* This means -r is false in any model. An additional inference step computes the unit clause *Q(t* o *y).* No further inference is possible on this subset.

7 Conclusion

We have pointed out a number of problems with evaluating the performance of different algorithms for modal reasoning. Our investigations show benchmarking needs to be done with great care. A crucial factor is the quality of the randomly generated problems, which we think are too easy. Further investigations are required concerning the parameter settings and fundamental properties of modal KCNF formulae before we can come to safe conclusions about different theorem proving approaches for modal logic.

A longer version of this paper is available as Research Report MPI-I-97-2-003.

References

[Baader and Hollunder, 1991] F. Baader and B. Hollunder. A terminological knowledge representation system with complete inference algorithms. In *Proc. PDK '91,* LNAI 567, pages 67-86. Springer, 1991.

[Catach, 1991] L. Catach. Tableaux: A general theorem prover for modal logics. *J. Automated Reasoning,* 7(4):489~510, 1991.

[D'Agostino, 1992] M. D'Agostino. Are tableaux an improvement on truth-tables? *J. Logic, Language, and Information,* 1:235-252, 1992.

[Giunchiglia and Sebastiani, 1996a] F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures: Case study of modal K. In *Proc. CADE-13,* LNAI 1104, pages 583-597, 1996.

[Giunchiglia and Sebastiani, 1996b] F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for *ACC.* In *Proc. KR'96,* pages 304-314, 1996.

[Goble, 1974] L. F. Goble. Gentzen systems for modal logic. *Notre Dame J. Formal Logic,* 15:455-461,1974.

[Mitchell *et al,* 1992] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proc. AAAI-10,* pages 459-465, 1992.

[Ohlbach and Schmidt, 1995] H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. Res. Report MPI-I-95-2-002, MPI fur Informatik, Saarbrucken, 1995.

[Schmidt, 1997] R. A. Schmidt. Resolution is a decision procedure for many propositional modal logics: Extended abstract. To appear in *Proc. AiML'96,* 1997.

[Weidenbach *et al,* 1996] C. Weidenbach, B. Gaede, and G. Rock. SPASS & FLOTTER version 0.42. In *Proc. CADE-13,* LNAI 1104, pages 141-145, 1996.