Efficient SAT-Based Minimal Model Generation Methods for Modal Logic S5

Pei Huang[1,3], Rundong Li[1,3], Minghao Liu[1,3], Feifei Ma[1,2,3(✉)], and Jian Zhang[1,3(✉)]

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
{huangpei,lird,liumh,maff,zj}@ios.ac.cn
[2] Laboratory of Parallel Software and Computational Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. Modal logic S5 is useful in various applications of artificial intelligence. In recent years, the advance in solving the satisfiability problem of S5 has allowed many large S5 formulas to be solved within a few minutes. In this context, a new challenge arises: how to generate a minimal S5 Kripke model efficiently? The minimal model generation can be useful for tasks such as model checking and debugging of logical specifications. This paper presents several efficient SAT-based methods and provides a symmetry-breaking technique for the minimal model generation problem of S5. Extensive experiments demonstrate that our methods are good at tackling many large instances and achieve state-of-the-art performances. We find that a minimal model of a large S5 formula is usually very small, and we analyze this phenomenon via a graph model. Due to this characteristic, our incremental method performs best in most cases, and we believe that it is more suitable for minimal S5 Kripke model generation.

Keywords: SAT · MaxSAT · Modal logic S5 · Minimal model · Symmetry breaking

1 Introduction

Modal logics provide a theoretical framework for applications in various areas of artificial intelligence. In the past two decades, modal logics have been used in game theory [12], knowledge compilation [5], contingent planning [20] and formal verification [2,18]. Moreover, they also have potential in database theory [10] and distributed computing [13].

S5 is one of the five oldest systems of modal logic. It is suitable for representing and reasoning about the knowledge of a single agent [8,24] and has been used in knowledge compilation [5], contingent planning [20] and epistemic planner [25]. The huge application potential promotes us to improve the practical automated reasoning technique for S5.

C.-M. Li and F. Manyà (Eds.): SAT 2021, LNCS 12831, pp. 225–241, 2021.
https://doi.org/10.1007/978-3-030-80223-3_16

In this paper, we focus on how to efficiently find a minimal S5 Kripke model (MinS5-SAT). In Hardware Verification and Model Checking (e.g. safety property), the model is in fact an explanation of the bug found in the design. A smaller model is easier to understand; it can be more meaningful and helpful to the user for checking or locating the bug precisely. Since the 1990s, a few theoretical works (considering soundness and completeness) about minimal Herbrand model generation were given for modal logic S5 [21,22], but very little work focuses on MinS5-SAT. As for practical algorithms (or solvers), they were designed to decide the satisfiability of modal formulas but rarely considered generating a minimal S5 Kripke model.

The significant improvements of SAT-based S5 solvers in recent years [6,14] pave the way for well handling the MinS5-SAT problem. In 2018, Jean-Marie Lagniez et al. proposed an SAT-based method to solve the MinS5-SAT problem [17], but the translation method can cause the space explosion in many cases which greatly wears out the efficiency of the back-end SAT/MaxSAT solver. For some complex input S5 formulas, it consumes more than 60G of memory.

We provide several practical and efficient SAT-based methods for the MinS5-SAT problem. Compared with the previous works, our methods can make good use of the structural and semantic information to reduce memory usage and eliminate a lot of symmetric (isomorphic) search spaces. A more compact encoding can make the most of the performance of the SAT engine. Experimental results show that our SAT-based methods are efficient in tackling the MinS5-SAT problem, and the method which queries an SAT oracle incrementally is the most efficient one. We noticed that a minimal Kripke model is usually very small. So, we propose a graph model to analyze the reason and find out that "a small model is a high probability event". It also explains why the method based on the incremental framework performs better in both time and memory consumption compared with other methods.

2 Preliminaries

This section briefly reviews the syntax, the semantics and some concepts of modal logic S5.

2.1 Syntax and Semantics

The set of formulas ϕ of S5 is a language $\mathcal{L}$ which extends the propositional language with the modal connectives (or modal operators) $\Box$ and $\Diamond$. The language $\mathcal{L}$ is defined by the grammar:

$$\phi ::= \bot \mid \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \Box\phi \mid \Diamond\phi$$

where $p \in \mathbb{P}$ and $\mathbb{P}$ denotes a countably infinite non-empty set of propositional variables. Logical connectives '$\rightarrow$' and '$\leftrightarrow$' are omitted here.

Standard Kripke semantics for modal logic defines a frame, which consists of a non-empty set W of **possible worlds**, and a binary relation R. The relation

R, also known as the **accessibility relation**, is defined between the possible worlds in W. The axioms $\mathcal{K}$, $\mathcal{T}$, $\mathcal{B}$ and 4 restrict that the relation R in S5 is **reflexive**, **symmetric** and **transitive**. So the possible world semantics for S5 can be simplified as a simple version without accessibility relation [9]. The satisfiability relation $\vDash$ for formulas in $\mathcal{L}$ is recursively defined as follows:

$(W, I, w) \vDash \top$
$(W, I, w) \vDash p$ iff $I(w, p) = 1$
$(W, I, w) \vDash \neg\phi$ iff $(W, I, w) \nvDash \phi$
$(W, I, w) \vDash \phi \wedge \varphi$ iff $(W, I, w) \vDash \phi$ and $(W, I, w) \vDash \varphi$
$(W, I, w) \vDash \phi \vee \varphi$ iff $(W, I, w) \vDash \phi$ or $(W, I, w) \vDash \varphi$
$(W, I, w) \vDash \Box\phi$ iff $\forall w' \in W, (W, I, w') \vDash \phi$
$(W, I, w) \vDash \Diamond\phi$ iff $\exists w' \in W, (W, I, w') \vDash \phi$

2.2 Satisfiability

There are three types of satisfiability problems for S5.

- [**S5-Satisfiability (S5-SAT)**] Determining if there exists a model (W, I, w) that satisfies a given S5 formula θ.
- [**S5-K-Satisfiability (S5-K-SAT)**] Determining if there exists a model (W, I, w) where $|W| = K$ that satisfies a given S5 formula θ.
- [**Minimal S5-Satisfiability (MinS5-SAT)**] Finding a model (W, I, w) that satisfies a given S5 formula θ and it has no model (W', I', w') such that $|W'| < |W|$.

S5-SAT and S5-K-SAT are decision problems and both are NP-complete. MinS5-SAT is NP-hard and can be seen as an optimization problem.

2.3 The Number of Possible Worlds

In 1977, Ladner gave the upper bound of the number of possible worlds to decide the satisfiability of an S5 formula [16]. He proved that if an S5 formula with m modal operators is satisfiable, then there exists an S5-model satisfying the formula with at most $m + 1$ worlds. In 2017, a new upper-bound $dd(\theta) + 1$ was found where $dd(\theta)$ is called diamond degree [6] and it is recursively defined as:

$dd(\theta) = dd'(\text{nnf}(\theta))$
$dd'(\top) = dd'(\neg\top) = dd'(p) = dd'(\neg p) = 0$
$dd'(\phi \wedge \varphi) = dd'(\phi) + dd'(\varphi)$
$dd'(\phi \vee \varphi) = max(dd'(\phi), dd'(\varphi))$
$dd'(\Box\phi) = dd'(\phi)$
$dd'(\Diamond\phi) = 1 + dd'(\phi)$

In 2019, a more compact upper bound, $\chi + 1$ was given by Huang et al. [14]. For most cases, we have:

$$\chi \leq dd(\theta) \leq m \tag{1}$$

The upper-bound χ+1 is reasoned from the relationship among diamond subformulas via a polynomial-time approximation graph coloring algorithm.

2.4 S5-NF

Every S5 formula can be converted into an equivalent one that is in S5-NF. S5-NF is helpful for improving efficiency and saving memory when solving the S5 satisfiability problems. Identifying structural information and semantic information of an S5-NF formula is relatively easy. The definition of S5-NF is in [14]. We only briefly review it here. S5-NF is a kind of CNF-like first degree normal form but with some textural difference. The basic unit that makes it up is **S5-literal**:

Propositional literal: p
B-literal: $\Box(p \vee q \vee ... \vee r)$
D-literal: $\Diamond(p \wedge q \wedge ... \wedge r)$

where p, q, r are propositional literals.

The disjunction of S5-literals is called **S5-clause** and the conjunction of S5-clauses is called **S5-NF**.

Example 1. An S5 formula θ and its S5-NF ϕ with three S5-clauses.

$$\theta = \Diamond\Box((r \rightarrow p \wedge q) \wedge (\Diamond(\neg r \rightarrow \neg p \wedge q))) \wedge (\neg p \rightarrow \neg\Box(q \wedge r))$$

$$\phi = \underbrace{\Box(p \vee q \vee \neg r)}_{C_1} \wedge \underbrace{\{\Diamond(\neg p \wedge q) \vee \Diamond r\}}_{C_2} \wedge \underbrace{\{p \vee \Diamond(\neg q \wedge \neg r)\}}_{C_3}$$

2.5 Complexity Analysis

The MinS5-SAT is in $\mathcal{P}^{SAT}$ (or $\mathcal{FP}^{SAT}$).

Assuming that $\mathcal{M}^{S5\text{-}K\text{-}SAT}$ is an oracle Turing machine that has the capability to query an oracle for S5-K-SAT. For a given S5 formula, the upper bound μ of possible worlds can be computed in polynomial time, and it is less than the length (n) of input formula. As the Turing machine $\mathcal{M}^{S5\text{-}K\text{-}SAT}$ can find a minimal model via querying S5-K-SAT at most polynomial (μ) times, MinS5-SAT is in $\mathcal{P}^{S5\text{-}K\text{-}SAT}$. We know that S5-K-SAT is in $\mathcal{NPC}$, so MinS5-SAT $\in \mathcal{P}^{SAT}$.

It reveals that MinS5-SAT can be solved via iteratively calling an SAT engine at most $\mathcal{O}(n)$ times. On the other hand, the MinS5-SAT problem can be reduced to the MaxSAT problem.

In addition to SAT-based approaches, tableau, FOL-based and resolution-based methods can also tackle MinS5-SAT in theory. However, their practical execution efficiency is far less than SAT-based methods.

3 Methodology

In this section, we present the basic SAT-based methods. The encoding process is premised on S5-NF.

3.1 Querying SAT Iteratively

First, we will give the method via querying SAT iteratively. We use propositional variable p_j to denote the truth value of p in the possible world w_j. Then the S5-K-SAT can be encoded as SAT.

Definition 1. *Translation function $tr^{-}_{SAT}(\phi, K)$ can produce a propositional formula for an input S5-NF ϕ with K possible worlds:*

1. $\top \Rightarrow \top \quad \bot \Rightarrow \bot$
2. *For all propositional literals p in ϕ:* $p \Rightarrow p_0$
3. *For all B-literals in ϕ:*
   $\Box(p \vee q \vee \cdots \vee s) \Rightarrow \bigwedge_{j=0}^{K-1}(p_j \vee q_j \vee \cdots \vee s_j)$
4. *For all D-literals in ϕ:*
   $\Diamond(p \wedge q \wedge \cdots \wedge r) \Rightarrow \bigvee_{j=0}^{K-1}(p_j \wedge q_j \wedge \cdots \wedge r_j)$

If $tr^{-}_{SAT}(\phi, K)$ is satisfiable then we can conclude that the formula ϕ has a model with K possible worlds. So, the MinS5-SAT problem can be seen as an optimization problem:

$$\textbf{minimize} \quad K \quad \textbf{s.t.} \quad tr^{-}_{SAT}(\phi, K) \text{ is satisfiable.} \tag{2}$$

One can iterate the value of K between 0 and upper bound μ in an increasing (or decreasing, binary search, etc.) way to find a minimal K that can satisfy $tr^{-}_{SAT}(\phi, K)$. The advantage of this method is that it can make full use of the efficiency of the SAT engine.

3.2 Partial MaxSAT Model

If there is a D-literal $\Diamond\psi$ and $(W, I, w) \vDash \psi$, we say that the possible world w is assigned to $\Diamond\psi$ or w realizes $\Diamond\psi$. The key to find a minimal Kripke model for an S5-NF is how to assign the possible worlds to D-literals. For example, suppose that the upper bound for the formula in Example 1 is calculated to be 3 by some method. Then, we can construct a Kripke model M_1 in Fig. 1. The world w_1 is assigned to $\Diamond r$ in C_2 and w_2 is assigned to $\Diamond(\neg q \wedge \neg r)$. Now, if we want to save a possible world, we can try to assign w_0 to $\Diamond(\neg q \wedge \neg r)$ as shown in M_2. In this context, w_2 can be removed from the model because no diamond formula needs it. Based on this insight, some switch variables can be added for the D-literals. For each possible world w_j, we use a Boolean variable v_j to open or close it. When v_j is falsified, the possible world w_j will be closed. So, we can encode the MinS5-SAT as a partial MaxSAT (PMS) problem [7].

Definition 2. *Translation function $tr^{-}_{PMS}(\phi, \mu)$ can produce a partial MaxSAT formula for an input S5-NF ϕ with at most μ (upper bound) possible worlds:*

1. $\top \Rightarrow \top \quad \bot \Rightarrow \bot$
2. *For all propositional literals:* $p \Rightarrow p_0$

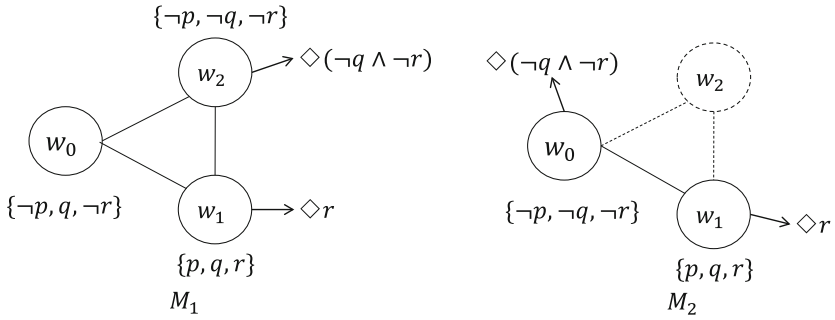

Fig. 1. Two Kripke models of the S5-NF in Example 1.

3. *For all B-literals:*
   $\Box(p \vee q \vee \cdots \vee s) \Rightarrow \bigwedge_{j=0}^{\mu-1}(p_j \vee q_j \vee \cdots \vee s_j)$
4. *For all D-literals:*
   $\Diamond(p \wedge q \wedge \cdots \wedge r) \Rightarrow \bigvee_{j=0}^{\mu-1}(v_j \wedge p_j \wedge q_j \wedge \cdots \wedge r_j)$
5. *Add a unit clause:* v_0
6. *Add unit soft clauses:* $\bigwedge_{j=1}^{\mu-1}(\neg v_j)$

Except for the clauses generated by rule 6, all other clauses are hard. The possible world w_0 must exist, so we add a hard unit clause v_0. The idea behind this partial MaxSAT formula is to close as many worlds as possible. Since the possible worlds do not need to be really removed, we just apply switch variables to diamond formulas. The number of satisfied switch variables is the size of a minimal model.

Example 2. If the upper bound μ for the S5-NF ϕ in Example 1 is 3, then the $tr^{-}_{PMS}(\phi, 3)$ is :

$$\bigwedge_{j=0}^{2}(p_j \vee q_j \vee \neg r_j) \wedge \{\bigvee_{j=0}^{2}(v_j \wedge \neg p_j \wedge q_j) \vee \bigvee_{j=0}^{2}(v_j \wedge r_j)\}$$

$$\wedge\{p_0 \vee \bigvee_{j=0}^{2}(v_j \wedge \neg q_j \wedge \neg r_j)\} \wedge v_0 \wedge \underbrace{\neg v_1}_{Soft} \wedge \underbrace{\neg v_2}_{Soft}$$

The advantage of the MaxSAT-based method is that UNSAT core and heuristics for optimization can be well used in finding a minimal model.

4 Improved Methods

Finding a minimal model for an S5-NF can be seen as searching for an optimal assignment of possible worlds for satisfied D-literals. In the basic model, the realization of D-literals will be tested in all possible worlds $\{w_0, w_1, ..., w_{\mu-1}\}$,

so there are a lot of symmetric (or isomorphic) situations. Suppose the process of finding kripke model is considered in order, clause by clause. Like Example 1, when we are considering how to realize the D-literals in C_2, and w_2, w_3 have not been considered for other D-literals now, actually, there is no difference trying to realize the D-literals in C_2 via w_2 or w_3.

4.1 SIF Strategy

We design a static symmetry breaking technique called **Smallest Index First (SIF)** strategy. Assuming that $tr^{-}_{SAT}(\phi, K)$ and $tr^{-}_{PMS}(\phi, \mu)$ translate the S5-NF ϕ from left to right, the index set of possible worlds $\{0, 1, ..., \mu - 1\}$ (or $\{0, 1, ..., K - 1\}$) can be divided into two parts: $\Omega_1 = \{l \in \mathbb{N} | 0 \leq l \leq L - 1\}$ and $\Omega_2 = \{l \in \mathbb{N} | L \leq l \leq \mu - 1\}$(or $\{l \in \mathbb{N} | L \leq l \leq K - 1\}$). The index in Ω_1 marks the possible worlds which have been used to test the realization of some D-literals. Ω_2 represents the possible worlds which have not been considered. If the S5-clause currently under consideration has D-literals, we only need to consider one more possible world at most. There is no difference for any one in Ω_2, because we can always find a permutation to rename the worlds in Ω_2. So we can add the w_L with the smallest index to Ω_1.

4.2 Improved S5-K-SAT Model

For S5-K-SAT encoding, initially $L = 1$, $\Omega_1 = \{0\}$, $\Omega_2 = \{1, ..., K-1\}$. Whenever the translation procedure encounters an S5-clause which has D-lierals, update L to $Min(L + 1, K - 1)$.

Definition 3. *Translation function $tr_{SAT}(\phi, K)$ is the improved version of $tr^{-}_{SAT}(\phi, K)$ with the SIF strategy:*

When the procedure is translating S5-clause C_i, update L, iff C_i has D-literals. For all D-literals in C_i:
$\Diamond(p \wedge q \wedge \cdots \wedge r) \Rightarrow \bigvee_{j \in \Omega_1}(p_j \wedge q_j \wedge \cdots \wedge r_j)$

Example 3. Assume that ϕ is the formula in Example 1 and $K = 3$, then the procedure of $tr_{SAT}(\phi, 3)$ can be like this:

- Step 1. Translating C_1. $L = 1$, $\Omega_1 = \{0\}$, $\Omega_2 = \{1, 2\}$.
  $\Box(p \vee q \vee \neg r) \Rightarrow \bigwedge_{j=0}^{2}(p_j \vee q_j \vee \neg r_j)$
- Step 2. Translating C_2. $L = L + 1$, $\Omega_1 = \{0, 1\}$, $\Omega_2 = \{2\}$.
  $\Diamond(\neg p \wedge q) \vee \Diamond r \Rightarrow \bigvee_{j=0}^{1}(\neg p_j \wedge q_j) \vee \bigvee_{j=0}^{1} r_j$
- Step 3. Translating C_3. $L = L + 1$, $\Omega_1 = \{0, 1, 2\}$, $\Omega_2 = \{\}$.
  $p \vee \Diamond(\neg q \wedge \neg r) \Rightarrow p_0 \vee \bigvee_{j=0}^{2}(\neg q_j \wedge \neg r_j)$

In the second step, we only consider that whether $\Diamond(\neg p \wedge q)$ and $\Diamond r$ can be realized via possible worlds w_0 and w_1. We do not need to consider the possible worlds in Ω_2 for it.

4.3 Improved PMS Model

For PMS encoding, initially $L = 1$, $\Omega_1 = \{0\}$ and $\Omega_2 = \{1, ..., \mu\}$. Whenever the translation procedure encounters an S5-clause which has diamond formula, update L to $Min(L + 1, \mu - 1)$.

Translation function $tr_{PMS}(\phi, \mu)$ is the improved version of $tr^{-}_{PMS}(\phi, \mu)$ with the SIF strategy:

(i) Add : $\bigwedge_{j=0}^{\mu-2}(v_{j+1} \rightarrow v_j)$
When the procedure is translating S5-clause C_i, update L, iff C_i has D-literals.

(ii) For all D-literals in C_i:
$\Diamond(p \wedge q \wedge \cdots \wedge r) \Rightarrow \bigvee_{j \in \Omega_1}(v_j \wedge p_j \wedge q_j \wedge \cdots \wedge r_j)$

The effect of (ii) is the same as the previous one. The added clauses (i) restrict that the indexes of the possible worlds in a minimal model are contiguous. This is a specific symmetry breaking for the MaxSAT encoding.

4.4 The Benefit of SIF

If we ignore the box subformulas and propositional literals, a minimal S5 model finding procedure for an S5-NF can be simply abstracted as a tree search process. It tests various assignments for D-literals and finds the one that uses the least number of possible worlds. Similarly, S5-K-SAT are searching for an assignment for these D-literals using just K possible worlds. The basic encoding of D-literals can be seen as testing possible assignments to realize it. In this perspective, the comparison of the basic methods with the improved methods is shown in Fig. 2. In the basic methods, all the possible assignments for D-literals will be tested. In the improved methods, fewer assignments will be tested for the D-literals at a high level of the tree. The D-literals in the first S5-clause only test $\{w_0, w_1\}$, and the D-literals in the second S5-clause only test $\{w_0, w_1, w_2\}$, and so on. So the search tree of the improved methods can be much "thinner" and a lot of symmetric situations are eliminated. When the upper bound μ is big, the search space reduced by the SIF strategy is considerable.

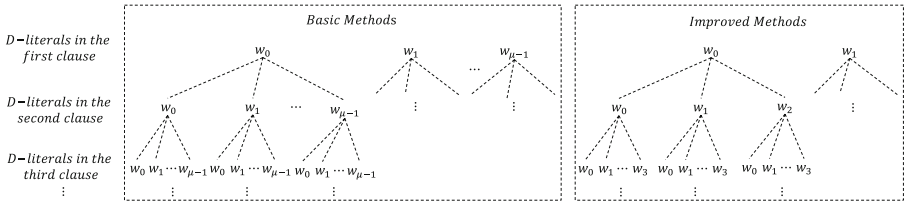

Fig. 2. The search spaces of the basic and improved methods.

5 Comparison

The state-of-the-art MinS5-SAT solver S52SAT 2.0 also uses a SAT based method [17]. The input formula θ will be translated to a MaxSAT formula. The main translation procedure $tr_s(\theta, \mu)$ is recursively defined as:

$$tr_s(\theta, \mu) = tr'_s(\theta, 0, \mu - 1)$$
$$tr'_s(p, i, \mu - 1) = p_i \qquad tr'_s(\neg p, i, \mu - 1) = \neg p_i$$
$$tr'_s((\varphi \wedge \delta), i, \mu - 1) = tr'_s(\varphi, i, \mu - 1) \wedge tr'_s(\delta, i, \mu - 1)$$
$$tr'_s((\varphi \vee \delta), i, \mu - 1) = tr'_s(\varphi, i, \mu - 1) \vee tr'_s(\delta, i, \mu - 1)$$
$$tr'_s(\Box\varphi, i, \mu - 1) = \bigwedge_{j=0}^{\mu-1}(s_j \rightarrow tr'_s(\varphi, j, \mu - 1))$$
$$tr'_s(\Diamond\varphi, i, \mu - 1) = \bigvee_{j=0}^{\mu-1}(s_j \wedge tr'_s(\varphi, j, \mu - 1))$$
$$\text{Add: } \bigwedge_{j=0}^{\mu-2}(s_{j+1} \rightarrow s_j)$$

Variables s_i are added to enable or disable worlds w_i.

In general, the input formula θ can be in a very complex form, which has nested modal operators and arbitrary combinations of logical connectives. This translation method can produce a large SAT formula with redundancies. Besides, there are a lot of symmetric situations in $tr_s(\theta, \mu)$. Compared with it, our translation method has three advantages: First, the input formula θ will be transformed to an equivalent S5-NF and the nested modal operators are eliminated. It makes our methods produce a relatively small formula in the translation phase. And our method will not add variables for box formulas. Second, the structural information of the formula can be used to eliminate a lot of isomorphic models and improve the efficiency in the search phase. Third, the reasoning can be relatively efficient in possible worlds except for w_0. The diamond formulas only have "$\wedge$". When the process tests whether some D-literals can be realized by a certain world, the unit propagation can be activated.

6 Experimental Evaluation

Based on the approaches presented in this article, we implemented the MinS5-SAT solver *S5cheetah 2.0*[1]. The upper bound used in *S5cheetah 2.0* is $\chi + 1$. The *Glucose 4.0* [3] is used as the back-end SAT solver and *RC2* [15] is used as the back-end MaxSAT solver which has the best performance in the MaxSAT competition 2019[2].

First, we clarify some notations in the experimental part:

[Inc.] It denotes the method which queries the satisfiability of $tr_{SAT}(\phi, K)$ with increasing values of K from 1 to the upper bound $\chi + 1$.

[Dec.] It denotes the method which queries the satisfiability of $tr_{SAT}(\phi, K)$ with decreasing values of K from the upper bound $\chi + 1$ to 1.

[Bs.] It denotes the method which queries the satisfiability of $tr_{SAT}(\phi, K)$ using binary search for the optimal K between 1 and the upper bound $\chi + 1$.

[1] S5cheetah and benchmarks: http://www.square16.org/tools/s5cheetah/.

[2] https://maxsat-evaluations.github.io/2019/rankings.html.

[**PMS.**] It denotes our MaxSAT method.

[**S52SAT 2.0**] It is a state-of-the-art MinS5-SAT solver[3].

[**S52SAT 1.0**] It is an S5-SAT solver and used as a reference in our experiment. In general, the MinS5-SAT problem is harder than the S5-SAT problem. So, the time and memory consumption of *S52SAT 1.0* can be regarded as the lower bound of the time and memory consumption of *S52SAT 2.0*. Based on the reports in paper [17], *S52SAT 1.0* uses a similar translation method like $tr_s()$ but without selector s_i and consumes less time than *S52SAT 2.0*.

[**Lck.**] It denotes the S5-SAT solver *LCKS5TabProver* [1]. It is the state-of-the-art tableau method solver for S5 and used as a reference. The advantage of SAT-based methods can be seen from the comparison of LCK with other SAT-based methods.

We cannot directly compare *S5cheetah 2.0* with other MinS5-SAT solvers, but we can infer the strength of our methods from the experimental results. The correctness of *S5cheetah 2.0* is cross-validated by our different methods.

6.1 Benchmarks

The well-established modal logic benchmarks include $QMLTP$[4], $3CNF$ [23], $3CNF_{mu}$ $QS5\_1$, $QS5\_2$, $MQBF_K$ [19], and $LWB_{K,KT,S4}$ [4]. $QMLTP$ is designed for testing automated theorem proving (ATP) systems for first-order modal logics. It contains 177 propositional benchmarks for S5 from different domains (e.g. planning, querying databases, natural language processing, general algebra). All the benchmarks in $QS5\_1$ and $QS5\_2$ are satisfiable and large, they are generated based on hard combinatorial designs about quasigroups. Most benchmarks in $3CNF$ are unsatisfiable, so we generate $3CNF_{mu}$ based on $3CNF$. All the instances in $3CNF_{mu}$, which are generated from small mutations of all the unsatisfiable instances in $3CNF$, are satisfiable. Note that $MQBF_K$ and $LWB_{K,KT,S4}$ are not designed for S5. However, the results on those benchmarks are still valuable. They share the same grammar and S5-SAT entails K, KT and S4-SAT. Furthermore, the size of a minimal S5 Kripke model can be seen as an upper bound for K, KT and S4. All the benchmarks can be downloaded from the link (See footnote 1).

6.2 Environment

The experiments are performed on a server with Intel(R) Xeon (R) CPU (2.40 GHz), Ubuntu 16.04 and 64G RAM. The time bound is set to 300 s for each instance. The experiments were performed on all SATISFIABLE instances.

[3] S52SAT 2.0. is not available at moment. So, we compare with S52SAT 1.0 as reference. http://www.cril.univ-artois.fr/%7emontmirail/s52SAT/v2/index.html.

[4] http://www.iltp.de/qmltp/.

6.3 Experimental Results

Table 1 shows the comparison of efficiency on each instance family. For each solver on each instance family, we report the number of instances where the solver uses the least time among all solvers in the table, denoted by "$\#Win$" and the average time ($T_{avg}(ms)$) for tackling each instance family. The unsolved instances are counted with the time bound. The sum of "$\#Win$" of all solvers is not necessarily equal to "$\#total$", since all solvers may fail. "$\#Mem$" records the maximum memory usage.

- [**Lck vs. Other solvers**] Lck failed in tackling many instances compared with SAT-based methods. Even on some small examples, it is much slower than other SAT-based methods.
- [**PMS vs. S52SAT 1.0**] Our PMS method consumes less time and memory than *S52SAT 1.0*. We infer that PMS performs better than *S52SAT 2.0*.
- [**PMS vs. Dec and Bs**] The time consumption of Dec and Bs is close to that of PMS, but PMS consumes more memory than the methods which are based on querying S5-K-SAT iteratively.
- [**Inc vs. Dec, Bs and PMS**] For most instances, the incremental method performs best in terms of memory usage and efficiency. Figure 3 shows the comparison of running time on some hard families (3CNF and QS5). We find that minimal Kripke models are usually small. So, the incremental method has the best performance in solving the MinS5-SAT problem.
- [**SIF vs. No SIF**] SIF is a symmetry-breaking strategy that hardly introduces extra computation cost. It really can help us solve more instances and improve the efficiency of our methods. For $3CNF_{mu}$, it can help Inc solve 5 more instances (943 vs. 938), help Dec solve 67 more instances (401 vs. 334), help Bs solve 53 more instances (401 vs. 334) and help PMS solve 180 more instances (409 vs. 229). For $QS5\_2$, it cannot help Inc solve more instances (154 vs 154) but can help Dec solve 43 more instances (154 vs. 111), help Bs solve 2 more instances (127 vs. 125) and help PMS solve 6 more instances (116 vs 110). For other benchmarks, all the instances can be solved by all methods (with or without SIF) but the methods with SIF consume less time. Figure 4 shows the comparison of running time of the methods with SIF and without SIF on $3CNF_{mu}$ and $QS5\_2$. Sometimes the method using SIF will take a little longer time because of the fluctuation of the CPU and randomized strategy of the SAT solver. SIF has more obvious advantages when the instance is more difficult and the minimum model is larger.

Table 2 shows the average size of minimal models ($MinW$) with the estimated upper bounds ($\chi + 1$) and ($dd(\theta) + 1$). We can see that a minimal model is very small and sometimes the theoretical upper bound is much larger than the size of the number of minimal possible worlds.

7 Why Is a Minimal Model Small?

Although basic modal logics have small model property, the concept is equivalent to the finite model property [11]. Sometimes this theoretical "small" model can

Table 1. The comparison of efficiency on all benchmarks. "-" means no instance can be solved within the time bound.

Ins (#Total)	Inc.			Dec.			Bs.			PMS.			S52SAT1.0		Lck.	
	$\#Win$	T_{avg}	Mem	$\#Win$	T_{avg}	Mem	$\#Win$	T_{avg}	Mem	$\#Win$	T_{avg}	Mem	T_{avg}	Mem	T_{avg}	Mem
$QMLTP$(41)	41	0.5	0.8M	0	1.3	1.0M	0	0.91	0.9M	0	1.1	20.1M	78.90	25.0M	43910.9	21G
$QS5\_1$(252)	248	1995.6	4.8G	0	29745.3	5.5G	0	6800.64	5.3G	4	5846.4	6.6G	107378.2	64G	–	–
$QS5\_2$(240)	149	129303.9	3.1G	0	189785.7	8.5G	0	164857.9	6.2G	5	181272.8	8.8G	290205.8	64G	–	–
$3CNF$ (55)	54	19476.2	62.8M	0	223919.6	109.7M	0	40669.2	76.1M	0	238946.2	250.1M	126804.3	2.35G	–	–
$3CNF_{mu}$(945)	939	7776.2	59.9M	0	207520.2	100.7M	0	48329.9	55.5M	0	187422.1	174.0M	290205.9	64G	–	–
LWB_k (42)	42	154.5	46.8M	0	180.6	51.5M	0	166.7	48.8M	0	176.4	59.4M	81326.9	64G	147467.9	64G
LWB_kt (105)	105	7.8	48.1M	0	15.6	55.1M	0	14.2	55.9M	0	18.4	62.8M	30916.2	64G	61103.1	58G
LWB_s4(105)	105	40.4	6.8M	0	47.6	7.1M	0	46.5	7.1M	0	51.8	7.7M	38299.4	24.5G	57393.4	62G
$qbfS$ (177)	177	1.59	3.0M	0	5.78	4.8M	0	3.19	3.0M	0	3.24	12.4M	591.44	84.0M	–	–

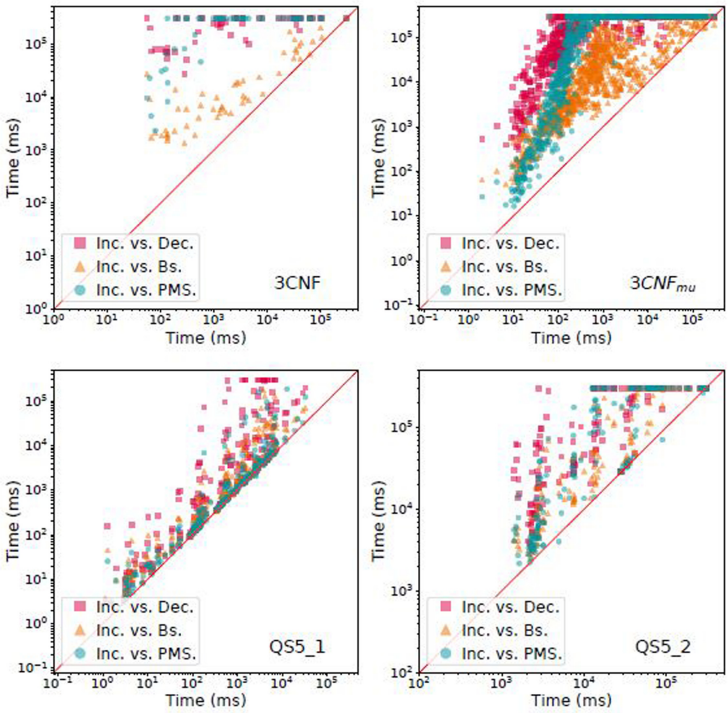

Fig. 3. The comparison of running time on $3CNF$ and $QS5$.(The x-axis corresponds to the time used by incremental method and the y-axis corresponds to the time used by other methods. The axes are in logarithmic scale. The point above the line y=x means that the incremental method consumes less time on this instance.)

be very large. To our best knowledge, there is no literature about the minimal number of possible worlds of modal logics. So, we try to figure out whether a minimal S5 Kripke model is so small frequently.

Suppose the input formula ϕ is an S5-NF, and there are m S5-clauses containing at least one D-literal. Based on the definition of diamond degree, we know that $dd(\phi)=m$. So the satisfiability of ϕ can be decided with at most $m+1$

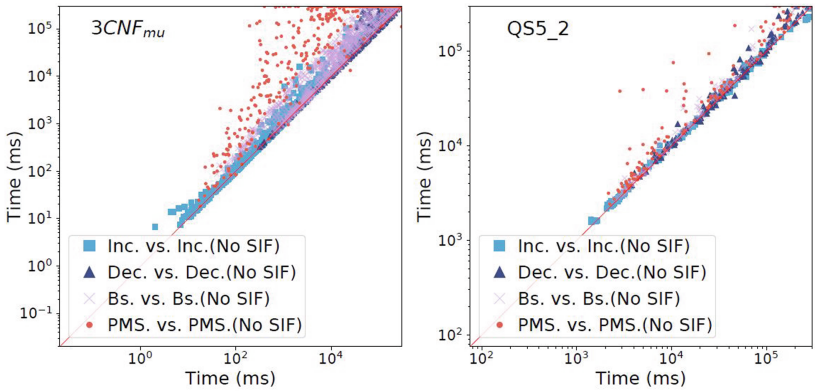

Fig. 4. The comparison of running time on $3CNF_mu$ and $QS5\_2$.(The x-axis corresponds to the time used by methods with SIF and the y-axis corresponds to the time used by methods without SIF. The axes are on a logarithmic scale. The point above the line y=x means the method with SIF consumes less time on this instance.)

Table 2. The minimal number of possible worlds VS. estimated upper bounds.

Ins	MinW	$\chi+1$	$dd(\theta)+1$	$m+1$
$QMLTP$	1.31	1.75	6.87	242.73
$QS5\_1$	1.00	16.05	2826.70	8333.36
$QS5\_2$	1.92	40.93	1329.41	35095.44
$3CNF$	7.03	125.45	152.82	427.85
$3CNF_{mu}$	6.89	233.47	316.50	797.16
LWB_k	1.50	3.47	440.92	1049.48
LWB_kt	1.40	4.00	217.89	1035.52
LWB_s4	1.20	2.57	236.22	1130.41
$qbfS$	2.00	3.00	45.07	1637.87

possible worlds. If ϕ is satisfiable then at least one S5-literal in each S5-clause must be true in its minimal model and we call that S5-literal as the "key S5-literal". Actually, the number of possible worlds is decided by all key D-literals. We can build a graph G for these "key D-literals". The vertices set are key D-literals and the edges are used to denote whether there is a conflict (they cannot be realized by the same world) between two key D-literals. Figure 5 is a schematic. We assume that P_k is the probability of that the key S5-literal is a D-literal in an S5-clause or whether a vertex appears in the graph. We use P_c to denote the probability of the appearance of a conflicting edge between two key D-literals. In this context, if the minimal chromatic number of the graph coloring problem of G is $\chi(G)$, then the minimal size of the model is either $\chi(G)$ or $\chi(G)$+1. According to literature [24], the chromatic number upper bound is

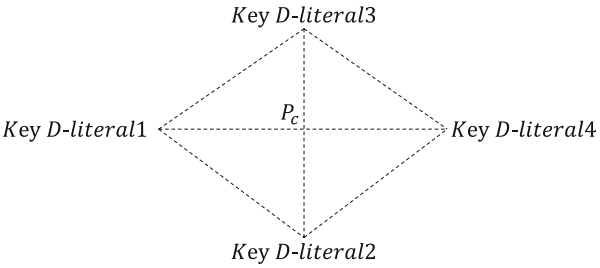

Fig. 5. The conflict graph of four key D-literals.

$\chi(G) \leq \delta = max_{i \in V} min(d_i + 1, i)$, where d_i denotes the degree of vertex i and $d_1 \geq d_2 \geq ... \geq d_{|V|}$. Then the probability that the chromatic number less than some constant H is:

$$P(\delta \leq H) = P(|\{d_i | d_i \geq H\}| \leq H)$$

We simplify the model and use independent random variable $X_{ij} (i, j \in \{1, 2, ..., m\})$ to denote the appearance of edge between two vertexes i and j. $X_{ij} = 1$ means that vertex i appears in graph otherwise, $X_{ij} = 0$.

$$P(d_i \geq H) = P(\sum_{j=1}^{m} X_{ij} \geq H) = P(\frac{\sum_{j=1}^{m} X_{ij} - mp_k^2 p_c}{\sqrt{mp_k^2 p_c (1 - p_k^2 p_c)}} \geq \frac{H - mp_k^2 p_c}{\sqrt{mp_k^2 p_c (1 - p_k^2 p_c)}})$$

Based on central limit theorem, we have:

$$Z = \frac{\sum_{j=1}^{m} X_{ij} - mp_k^2 p_c}{\sqrt{mp_k^2 p_c (1 - p_k^2 p_c)}} \sim N(0, 1)$$

So, $P(d_i \geq H) = \Phi(1 - \frac{H - mp_k^2 p_c}{\sqrt{mp_k^2 p_c (1 - p_k^2 p_c)}})$ where Φ is the cumulative distribution function (CDF) of normal distribution $N(0, 1)$.

We simply use independent random variable Y_i to denote whether $d_i \geq H$ and $P(Y_i = 1) = P(d_i \geq H) = p_d$. Then we have:

$$P(|\{d_i | d_i \geq H\}| \leq H) = P(\sum_{i=1}^{m} Y_i \leq H) = P(\frac{\sum_{i=1}^{m} Y_i - mp_d}{\sqrt{p_d (1 - p_d)}} \leq \frac{H - mp_d}{\sqrt{p_d (1 - p_d)}})$$

Reuse the central limit theorem, we have:

$$P(\delta \leq H) = P(|\{d_i | d_i \geq H\}| \leq H) = \Phi(\frac{H - mp_d}{\sqrt{p_d (1 - p_d)}})$$

If $m = 200$, $P_k = \frac{1}{3}$ (There are 3 types of S5-literals in an S5-clause and they are chosen as key S5-literal with equal probability.) and $P_c = \frac{1}{2}$ (Simply assuming that the existence of an edge follows the uniform distribution.), then $P(\delta \leq \frac{m}{10} =$

20) = 99.99%. If $m = 300$, $P(\delta \leq \frac{m}{10}) \approx 100\%$. In practice, m is much smaller than $|\phi|$. Besides, the minimal chromatic number can be much smaller than the estimated upper bound δ. The upper bound χ+1 is estimated from conflicting relationship among D-literals, but one cannot know which D-literals are key D-literals and get the accurate conflicting relationships in advance. So the upper bound χ+1 can also be much larger than $\chi(G)$+1. So the minimal number of possible worlds can be many times smaller than estimated upper bound with a high probability. This also shows that there is room for improvement in the upper bound.

Intuitively, P_k is inversely proportional to the average number of S5-literals in each S5-clause. P_c is proportional to the length of the D-literal. In general, the key D-literal is not too long for a satisfiable formula, because its conjunctive structure determines that if it is too long, it will make the assignment of a certain world very demanding, which is easy to make the whole formula to be unsatisfiable. As P_k and P_c get smaller in some larger formulas, δ will be smaller than a smaller constant H with a higher probability. To some extent, this model can help us figure out why a minimal model can be so small and the incremental method is more efficient.

8 Conclusion

In this paper, we propose some efficient SAT-based methods with a symmetry-breaking technique to generate minimal Kripke models for modal logic S5 formulas. Extensive experiments show that our methods achieve state-of-the-art performances and the incremental method is more efficient and consumes less memory compared with other methods. We notice that the minimal S5 Kripke model is usually very small. So, we analyze the reason with a graph model and find that "a small Kripke S5 model is a high probability event". It explains why querying an SAT oracle incrementally performs best in most cases. In the future, we would like to migrate these methods to other modal logics such as S4 and KD45 which are similar to S5.

Acknowledgements. This work has been supported by the National Natural Science Foundation of China (NSFC) under grant No.61972384 and the Key Research Program of Frontier Sciences, Chinese Academy of Sciences under grant number QYZDJ-SSW-JSC036. Feifei Ma is also supported by the Youth Innovation Promotion Association CAS under grant No. Y202034. The authors would like to thank the anonymous reviewers for their comments and suggestions.

References

1. Abate, P., Goré, R., Widmann, F.: Cut-free single-pass tableaux for the logic of common knowledge. In: Workshop on Agents and Deduction at TABLEAUX. vol. 2007. Citeseer (2007)

2. Aguilera, J.P., Fernández-Duque, D.: Verification logic: an arithmetical interpretation for negative introspection. In: Advances in Modal Logic 11, proceedings of the 11th conference on Advances in Modal Logic, held in Budapest, Hungary, August 30 - September 2, 2016. pp. 1–20 (2016)
3. Audemard, G., Simon, L.: Predicting learnt clauses quality in modern SAT solvers. In: IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, 11–17 July 2009, pp. 399–404 (2009)
4. Balsiger, P., Heuerding, A., Schwendimann, S.: A benchmark method for the propositional modal logics k, kt, S4. J. Autom. Reason. **24**(3), 297–317 (2000)
5. Bienvenu, M., Fargier, H., Marquis, P.: Knowledge compilation in the modal logic S5. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11–15, 2010 (2010)
6. Caridroit, T., Lagniez, J., Berre, D.L., de Lima, T., Montmirail, V.: A SAT-based approach for solving the modal logic S5-satisfiability problem. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4–9 February 2017, San Francisco, California, USA. pp. 3864–3870 (2017)
7. Chu, Y., Luo, C., Cai, S., You, H.: Empirical investigation of stochastic local search for maximum satisfiability. Frontiers Comput. Sci. **13**(1), 86–98 (2019). https://doi.org/10.1007/s11704-018-7107-z
8. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.: Reasoning about knowledge. MIT press, Cambridge (2004)
9. Fitting, M.: A simple propositional S5 tableau system. Ann. Pure Appl. Log. **96**(1–3), 107–115 (1999)
10. Fitting, M.: Modality and databases. In: Dyckhoff, R. (ed.) TABLEAUX 2000. LNCS (LNAI), vol. 1847, pp. 19–39. Springer, Heidelberg (2000). https://doi.org/10.1007/10722086_2
11. Goranko, V., Otto, M.: Model theory of modal logic. In: Handbook of Modal Logic, pp. 249–329 (2007)
12. Grossi, D., Rey, S.: Credulous acceptability, poison games and modal logic. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2019, Montreal, QC, Canada, 13–17 May 2019, pp. 1994–1996 (2019)
13. Hella, L., et al.: Weak models of distributed computing, with connections to modal logic. Distrib. Comput. **28**(1), 31–53 (2013). https://doi.org/10.1007/s00446-013-0202-3
14. Huang, P., Liu, M., Wang, P., Zhang, W., Ma, F., Zhang, J.: Solving the satisfiability problem of modal logic S5 guided by graph coloring. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019. pp. 1093–1100 (2019)
15. Ignatiev, A., Morgado, A., Marques-Silva, J.: RC2: an efficient maxsat solver. J. Satisf. Boolean Model. Comput. **11**(1), 53–64 (2019)
16. Ladner, R.E.: The computational complexity of provability in systems of modal propositional logic. SIAM J. Comput. **6**(3), 467–480 (1977)
17. Lagniez, J.-M., Le Berre, D., de Lima, T., Montmirail, V.: An assumption-based approach for solving the minimal S5-satisfiability problem. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) IJCAR 2018. LNCS (LNAI), vol. 10900, pp. 1–18. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94205-6_1
18. Leuştean, I., Moangă, N., Şerbănuţă, T.F.: Operational semantics and program verification using many-sorted hybrid modal logic. In: Cerrito, S., Popescu, A. (eds.) TABLEAUX 2019. LNCS (LNAI), vol. 11714, pp. 446–476. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29026-9_25

19. Massacci, F.: Design and results of the tableaux-99 non-classical (Modal) systems comparison. In: Murray, N.V. (ed.) TABLEAUX 1999. LNCS (LNAI), vol. 1617, pp. 14–18. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48754-9_2
20. Niveau, A., Zanuttini, B.: Efficient representations for the modal logic S5. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, 9–15 July 2016. pp. 1223–1229 (2016)
21. Papacchini, F., Schmidt, R.A.: A tableau calculus for minimal modal model generation. Electron. Notes Theor. Comput. Sci. **278**, 159–172 (2011)
22. Papacchini, F., Schmidt, R.A.: Terminating minimal model generation procedures for propositional modal logics. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) IJCAR 2014. LNCS (LNAI), vol. 8562, pp. 381–395. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08587-6_30
23. Patel-Schneider, P.F., Sebastiani, R.: A new general method to generate random modal formulae for testing decision procedures. J. Artif. Intell. Res. **18**, 351–389 (2003)
24. Soto, M., Rossi, A., Sevaux, M.: Three new upper bounds on the chromatic number. Discret. Appl. Math. **159**(18), 2281–2289 (2011)
25. Wan, H., Yang, R., Fang, L., Liu, Y., Xu, H.: A complete epistemic planner without the epistemic closed world assumption. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 25–31 July 2015. pp. 3257–3263 (2015)