A Resolution Method for Modal Logic S5

Yakoub Salhi and Michael Sioutis

Université d'Artois, CRIL, CNRS UMR 8188
Rue Jean Souvraz, SP-18
62307, Lens Cedex 3
{salhi,sioutis}@cril.fr

Abstract

In this work, we aim to define a resolution method for the modal logic S5. We first propose a conjunctive normal form (S5-CNF) which is mainly based on using labels referring to semantic worlds. In a sense, S5-CNF can be seen as a generalization of the conjunctive normal form in propositional logic by including the modal connective of necessity and labels in the clause structure. We show that every S5 formula can be transformed into an S5-CNF formula using a linear encoding. Then, in order to show the suitability of our normal form, we describe a modeling of the graph coloring problem. Finally, we introduce a simple resolution method for S5, composed of three deductive rules, and we show that it is sound and complete. Our deductive rules can be seen as adaptations of Robinson's resolution rule to the possible-worlds semantics.

1 Introduction

Robinson's resolution method forms the keystone of logic programming and several automated theorem-provers. It corresponds to a proof rule leading to a decision procedure for deciding whether a formula is unsatisfiable in propositional logic. In this work, we are interested in defining a resolution method for modal logic S5.

The normal modal logic S5 is among the most studied modal logics. It is well-known that it can be considered as an epistemic logic in the sense that it is suitable for representing and reasoning about the knowledge of an agent [8]. In [12], Ladner proved that the complexity of the satisfiability problem in S5 is NP-complete.

In the literature, the majority of decision algorithms for modal logics are based on either the use of the formalism style in structural proof theory called sequent calculus and its variant called tableau method [9, 6], or encodings in first order logic [13]. There are also several works proposing resolution methods for modal logics which are mainly defined using two approaches, described as follows. The first approach is based on performing resolution inside modal connectives without integrating semantic information (see for instance [1, 7, 4]). As pointed out in [2], this leads to complex systems with a proliferation of deductive rules. The second approach is based on using semantic pieces of information, such as labels referring to semantic worlds, for direct resolution methods avoiding proliferation of deductive rules [2]. In this work, we adopt the second approach by using the notions of nominals and satisfaction connectives stemming from hybrid logics to define a simple and elegant system for S5. Hybrid logics were mainly introduced to explicitly express the relativity of truth in modal logics [5, 3]. This relativity is obtained by adding a new kind of propositional symbols, called nominals, which are used to refer to specific worlds in a model. In this context, new connectives having nice logical properties, called satisfaction operators, are added to allow one to jump to worlds named by nominals.

Our resolution system differs from the existing ones for S5 mainly in the following two positive aspects. Firstly, we use without loss of generality a normal form very close to the conjunctive normal form in propositional logic. Indeed, a formula in our normal form corresponds to a conjunction of clauses where each clause does not contain the connectives $\wedge$ and $\Diamond$, and the negation connective is only applied to propositional variables. By comparison, in the existing clausal normal form for S5, a clause can contain all the connectives of S5 [7]. Secondly, our system contains only three deductive rules which can be seen as adaptations of Robinson's resolution rule to the possible-worlds semantics. Intuitively, the two first rules can be seen as Robinson's resolution rule restricted to a single world, while the third rule concerns all the worlds. As a summary, the contribution of this paper is at least twofold: (*i*) we introduce an intermediate language between the language of S5 and that of its hybrid version, a conjunctive normal form in S5 similar to that in propositional logic, in particular, a conjunction of conjunction-free clauses, (*ii*) we provide a fairly simple and natural resolution method based on a small set of rules similar to Robinson's rule.

The rest of this paper is organized as follows. We provide an overview of the syntax and the semantics of modal logic S5 in Section 2. In Section 3, we define our conjunctive normal form for S5 and we describe a linear method for translating the S5 formulæ to this normal form. In order to show the suitability of our normal form, we describe in Section 4 a modeling of the graph coloring problem. In Section 5, we introduce a resolution method and show its soundness and completeness. Section 6 concludes with some directions for future work.

2 Modal Logic S5

In this section, we first describe the syntax and the semantics of modal logic S5. The set of formulae of S5 is defined by a denumerable set of propositional variables, denoted by $\mathcal{P}$ (we use $p, q, r, \ldots$ to range over $\mathcal{P}$), by extending the propositional language with the modal connectives $\Diamond$ and $\Box$. In particular, the language is defined by the following grammar:

$$A ::= p \mid \neg A \mid A \wedge A \mid A \vee A \mid A \to A \mid \Box A \mid \Diamond A$$

Given an S5 formula A, we use $Var(A)$ to denote the set of propositional variables occurring in A.

A Hilbert axiomatic system for S5 is given by the following axioms and rules:
0. Any substitution instance of a propositional tautology.
$\mathcal{K}$. $\Box(A \to B) \to (\Box A \to \Box B)$
$\mathcal{T}$. $\Box A \to A$
$\mathcal{B}$. $A \to \Box\Diamond A$
4. $\Box A \to \Box\Box A$

$$\frac{A \to B \qquad A}{B}\ [mp] \quad \text{and} \quad \frac{A}{\Box A}\ [nec]$$

It is well-known that S5 can be considered as an epistemic logic in the sense that it is suitable for representing and reasoning about the knowledge of a single agent [8]. A formula of the form $\Box A$ can be read as "the agent knows A". For instance, axiom 4 expresses the fact that if an agent knows A then she knows that she knows A (the positive introspection axiom).

Given an S5 formula A, the set of subformulæ of A is defined inductively as follows:

- A is a subformula of A;
- if $B \odot C$ is a subformula of A, then so are B and C, for $\odot = \wedge, \vee, \to$;

- if $\bigcirc B$ is a subformula of A, then so is B, for $\bigcirc = \neg, \Box, \Diamond$.

An S5 formula is in *S5 Negation Normal Form* (S5-NNF formula) if the negation operator is only applied to propositional variables and the allowed connectives are $\neg$, $\wedge$, $\vee$, $\Diamond$ and $\Box$. Using De Morgan's laws, which are valid in S5, and the equivalence $\neg\Box A \equiv \Diamond\neg A$, we know that every S5 formula can be written as an S5-NNF formula. We use $\mathcal{F}_{S5}$ to denote the set of S5 formulæ in S5 negation normal form.

There are two versions of possible-worlds semantics for S5, one in which accessibility is an equivalence relation, and one in which there is no accessibility relation (see, e.g., [10]). Intuitively, the latter semantics is obtained from the former by restricting our attention to one equivalence class in a model. In this paper, we consider the semantics in which there is no accessibility relation.

Definition 1 (S5-Interpretation). *Given an S5 formula A, an* S5-interpretation *of A is a pair $\mathcal{M} = (W, V)$ where W is a non-empty set (of worlds) and V is a function from $Var(A)$ to 2^W, where 2^W is the powerset of W, i.e., its set of subsets.*

Definition 2 (Satisfaction Relation). *The* satisfaction relation $\models$ *between an S5-interpretation $\mathcal{M} = (W, V)$, a world $w \in W$, and an S5 formula A, written $\mathcal{M}, w \models A$, is defined inductively as follows:*

$\mathcal{M}, w \models p$ *iff* $w \in V(p)$;
$\mathcal{M}, w \models \neg p$ *iff* $w \notin V(p)$;
$\mathcal{M}, w \models A \wedge B$ *iff* $\mathcal{M}, w \models A$ *and* $\mathcal{M}, w \models B$;
$\mathcal{M}, w \models A \vee B$ *iff* $\mathcal{M}, w \models A$ *or* $\mathcal{M}, w \models B$;
$\mathcal{M}, w \models \Diamond A$ *iff* $\exists w' \in W,\ \mathcal{M}, w' \models A$;
$\mathcal{M}, w \models \Box A$ *iff* $\forall w' \in W,\ \mathcal{M}, w' \models A$.

Definition 3 (S5 Satisfiability Problem). *Given an S5 formula A, determine whether there exists an S5-interpretation $\mathcal{M} = (W, V)$ of A with a world $w \in W$ s.t. $\mathcal{M}, w \models A$. If A is satisfied in $\mathcal{M}$, we say that $\mathcal{M}$ is an* S5-model *of A.*

The S5 satisfiability problem is NP-complete [12].

Consider the two S5 formulæ $\Diamond\neg p$ and $\Box(p \vee q)$. Clearly, these formulæ infer that there exists a world where p is false and q is true. This fact can be expressed by formula $\Diamond(\neg p \wedge q)$. However, since our aim is to use a conjunctive normal form close to that of propositional formulae with conjunction-free clauses, the subformulæ in the scope of the occurrences of $\Diamond$ have to be literals, i.e., propositional variables or negated propositional variables. Furthermore, the S5 formula $(\Diamond\neg p) \wedge (\Diamond q)$ does not express that p is false and q is true in the same world. Indeed, this formula may be satisfied by a model where $\neg p \wedge q$ is not true in any world. As a consequence, to define the conjunctive normal form for our resolution method we need to refer to possible worlds in S5 formulæ. To this end, we consider a syntactical fragment of a *hybrid version* of S5 which includes S5.

The hybrid version of S5, denoted by HybS5, is obtained by adding a new kind of symbols to the syntax of S5, called *nominals*, which are used to refer to specific worlds in a model [5, 3]. New connectives related to the nominals, called *satisfaction connectives*, are also added. They allow one to jump to the worlds named by nominals. In our work, we consider a syntactical fragment of HybS5 where nominals appear only with satisfaction connectives. We use $\mathcal{N}$ to denote the denumerable set of nominals (we use $a, b, c, ...$ to range over $\mathcal{N}$). The syntax of the fragment of HybS5 that we consider, denoted by FHybS5, is defined using the following grammar:

$$A ::= p \mid \neg p \mid A \wedge A \mid A \vee A \mid \Box A \mid \Diamond A \mid a : A$$

Clearly, every formula in $\mathcal{F}_{S5}$ is an FHybS5 formula. Given an S5-interpretation $\mathcal{M} = (W, V)$ and a HybS5 formula A, an *$\mathcal{M}$-assignment* of A is a function that assigns to each nominal in A a world in W. A *HybS5-interpretation* of A is a pair $(\mathcal{M}, f)$ where $\mathcal{M}$ is an S5-interpretation and f an $\mathcal{M}$-assignment of A. The *satisfaction relation* for the considered fragment, $(\mathcal{M}, f), w \models A$, is defined in the same manner as for S5 in the case of the common connectives, and in the following manner in the case of the satisfaction connectives: $(\mathcal{M}, f), w \models a : A$ iff $(\mathcal{M}, f), f(a) \models A$.

3 S5-CNF Formulæ

In this section, we first define a *Conjunctive Normal Form* for the S5 formulæ (S5-CNF in short). This normal form is obtained by generalizing the conjunctive normal form in propositional logic through the introduction of nominals and the modal connective of necessity in the clause structure. Then, we show that every formula in $\mathcal{F}_{S5}$ admits an equi-satisfiable S5-CNF formula using a linear encoding.

3.1 S5 Conjunctive Normal Form

In order to define our conjunctive normal form, we start by defining the notions of *literal*, *basic clause*, and *hyb-literal*. A *literal* is defined as a propositional variable or a negated propositional variable. We denote by $\bar{l}$ the complementary literal of l. More precisely, if $l = p$ then $\bar{l}$ is $\neg p$ and if $l = \neg p$ then $\bar{l}$ is p. A *basic clause* is defined as a disjunction of literals. Using the two aforementioned notions, a *hyb-literal* is defined as an FHybS5 formula of one of the forms $a : l$ or $\Box c$, where l is a literal, a is a nominal, and c is a basic clause.

Definition 4. *A* hyb-clause *is a disjunction of hyb-literals and an* S5-CNF formula *is a conjunction of hyb-clauses.*

An S5-CNF formula can also be seen as a set of hyb-clauses, and a hyb-clause as a set of hyb-literals; we refer to this representation when set operators are used, such as $\cup$ and $\cap$.

Example 1. *The following formula is in S5 conjunctive normal form:*

$$(a : p \vee b : \neg q \vee a : r) \wedge (b : \neg r \vee \Box(\neg p \vee \neg r) \vee a : q)$$

Definition 5 (S5-SAT)**.** *The S5-SAT problem consists of checking whether an S5-CNF formula is satisfiable or not.*

As a notational convention, we use the following symbols with subscripts and superscripts in what follows: l to denote literals, c to denote basic clauses, h to denote hyb-literals, α and β to denote hyb-clauses, and Δ to denote S5-CNF formulæ.

Proposition 1. *Let Δ be an S5-CNF formula. If Δ is satisfiable, then there exists a HyS5-model $(\mathcal{M}, f)$ of Δ such that $\mathcal{M}$ has $max(1, |Nom(\Delta)|)$ worlds.*

Proof. Let $((W, V), f)$ be a HyS5-model of Δ. We define the S5-interpretation $\mathcal{M} = (W', V')$ as follows:

- $W' = \begin{cases} \{f(a) \mid a \in Nom(\Delta)\} & if|Nom(\Delta)| > 0 \\ \{w\}\ for\ w \in W & otherwise \end{cases}$
- V' is the restriction of V to W'.

We now show that $(\mathcal{M}, f)$ is a HyS5-model of Δ. Let us assume that there exists a hyb-clause $a_1 : l_1 \vee \cdots \vee a_m : l_m \vee \Box c_1 \vee \ldots \vee \Box c_n$ which is not satisfied by $(\mathcal{M}, f)$. Then, since $W' \subseteq W$, for all $i \in \{1, \ldots, m\}$, $(W, V), f(a_i) \not\models l_i$ holds. Moreover, for all $j \in \{1, \ldots, n\}$, there exists a world $w \in W$ such that $(W, V), w \not\models c_j$ holds. Thus, we get a contradiction since $((W, V), f)$ is a HyS5-model of Δ. □

Theorem 1. *S5-SAT is NP-complete.*

Proof. Using Proposition 1, we know that S5-SAT is in NP. The NP-hardness stems from the fact that we can encode the propositional satisfiability problem (SAT) in S5-SAT. Indeed, we have $\Delta \equiv (l_1^1 \vee \cdots \vee l_{n_1}^1) \wedge \cdots \wedge (l_1^k \vee \cdots \vee l_{n_1}^k)$ is satisfiable iff $(a : l_1^1 \vee \cdots \vee a : l_{n_1}^1) \wedge \cdots \wedge (a : l_1^k \vee \cdots \vee a : l_{n_1}^k)$ is satisfiable, where a is a nominal. □

Given a hyb-clause α, we use $NHL(\alpha)$ to denote the number of hyb-literals occurring in α. $NHL(\cdot)$ is extended to the S5-CNF formulæ as follows: $NHL(\alpha_1 \wedge \cdots \wedge \alpha_n) = NHL(\alpha_1) + \cdots + NHL(\alpha_n)$. Consider for instance the S5-CNF formula Δ described in Example 1, we have $NHL(\Delta) = 6$. This measure is used to inductively prove the completeness of our resolution method.

3.2 A Transformation into S5-CNF

We will now describe a transformation which takes as input an arbitrary S5 formula and produces an equi-satisfiable S5-CNF formula. Similarly to Tseitin's well-known encoding [14], the key idea consists of substituting subformulæ with new propositional variables.

Given two S5 formulæ A and C, and a subformula B of A, we use $A[B/C]$ to denote the result of substituting B with C.

Proposition 2. *Let $A \in \mathcal{F}_{S5}$ and B a subformula of A. Then, A is satisfiable in S5 iff $A[B/q] \wedge \Box(q \leftrightarrow B)$ is satisfiable in S5, where q is a fresh variable, i.e., $q \notin Var(A)$.*

Proof. The "if" part is an obvious consequence of the fact that B is equivalent to the fresh variable q in any world of any model of $A[B/q] \wedge \Box(q \leftrightarrow B)$. Let us now consider the "only if" part. Let $\mathcal{M} = (W, V)$ be an S5-model of A. Then, we define the S5-interpretation $\mathcal{M}'$ of $A[B/q] \wedge \Box(q \leftrightarrow B)$ as the ordered pair (W, V') where V' is defined as follows:

$$V'(p) = \begin{cases} V(p) & if \quad p \neq q \\ \{w \in W \mid \mathcal{M}, w \models B\} & otherwise \end{cases}$$

Clearly, $\mathcal{M}'$ satisfies formula $\Box(q \leftrightarrow B)$ since $V(q) = \{w \in W \mid \mathcal{M}, w \models B\}$. Furthermore, we know that there exists $w \in W$ such that $\mathcal{M}, w \models A$. As a consequence, we have $\mathcal{M}', w \models A[B/q]$ since q is equivalent to B in every world in $\mathcal{M}'$. Thus, $\mathcal{M}', w \models A[B/q] \wedge \Box(q \leftrightarrow B)$ holds. □

It is interesting to note that each S5 formula can be represented by a binary rooted directed acyclic graph (DAG), where each leaf node is labelled with a literal and each internal node is labelled with a logical connective ($\wedge$ and $\vee$ (resp. $\Box$ and $\Diamond$) have two children (resp. a single

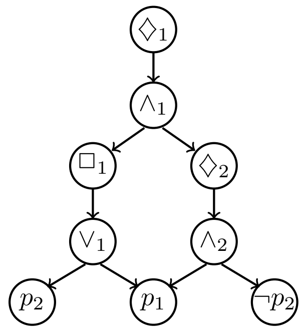

Figure 1: The DAG corresponding to formula $A = \Diamond(\Box(p_1 \vee p_2) \wedge \Diamond(p_1 \wedge \neg p_2))$

child)). An example of such a representation is depicted in Figure 1.

Our approach for transforming an S5 formula into an equi-satisfiable S5-CNF formula consists of the following two successive steps:

1. applying a Tseitin-like transformation;
2. removing the occurrences of $\Diamond$ and introducing nominals.

First step. Let $A \in \mathcal{F}_{S5}$ (we consider the DAG representation). In the first step, we start by associating with each internal node v of A a fresh propositional variable, denoted by q_v. We use $R(A)$ to denote the DAG obtained from A by replacing the label of each internal node with its corresponding fresh variable. We start with an empty set denoted by S_A. For all internal nodes v of A, if v is labelled with a binary logical connective $\otimes$ (i.e., $\wedge$ or $\vee$), we add formula $\Box(q_v \leftrightarrow x \otimes y)$ to S_A where x and y are the labels of the children of v in $R(A)$, and if v is labelled with a unary connective $\bigcirc$ ($\Box$ or $\Diamond$), we add formula $\Box(q_v \leftrightarrow \bigcirc x)$ to S_A where x is the label of the child of v in $R(A)$. Using Proposition 2, we know that $q_r \wedge \bigwedge_{B \in S_A} B$ is equivalent to A w.r.t. S5 satisfiability where q_r is the label of the root node in $R(A)$. After building S_A, we replace each formula in this set using the following equivalences:

- $\Box(q_v \leftrightarrow x \vee y) \equiv \Box(\neg q_v \vee x \vee y) \wedge \Box(\overline{x} \vee q_v) \wedge \Box(\overline{y} \vee q_v)$
- $\Box(q_v \leftrightarrow x \wedge y) \equiv \Box(q_v \vee \overline{x} \vee \overline{y}) \wedge \Box(x \vee \neg q_v) \wedge \Box(y \vee \neg q_v)$
- $\Box(q_v \leftrightarrow \Box(x)) \equiv (\Box(\neg q_v) \vee \Box(x)) \wedge (\Diamond(\overline{x}) \vee \Box(q_v))$
- $\Box(q_v \leftrightarrow \Diamond(x)) \equiv (\Box(\neg q_v) \vee \Diamond(x)) \wedge (\Box(\overline{x}) \vee \Box(q_v))$

These equivalences are obtained from valid equivalences in propositional logic (and also in S5) and the following valid equivalences in S5:

(*i*) $\Box(A \wedge B) \equiv \Box(A) \wedge \Box(B)$

(*ii*) $\Box(A \vee \Diamond(B)) \equiv \Box(A) \vee \Diamond(B)$

(*iii*) $\Box(A \vee \Box(B)) \equiv \Box(A) \vee \Box(B)$

(*iv*) $\Diamond(A) \equiv \neg\Box(\neg A)$

Let us consider, for instance, formula $\Box(q_v \leftrightarrow \Diamond(x))$. Using equivalences in propositional logic, we have $\Box(q_v \leftrightarrow \Diamond(x)) \equiv \Box((\neg q_v \vee \Diamond(x)) \wedge (\neg\Diamond(x) \vee q_v))$. Using equivalence (i), $\Box(q_v \leftrightarrow \Diamond(x)) \equiv \Box(\neg q_v \vee \Diamond(x)) \wedge \Box(\neg\Diamond(x) \vee q_v)$ holds. Then, using equivalence (iv), $\Box(q_v \leftrightarrow \Diamond(x)) \equiv \Box(\neg q_v \vee \Diamond(x)) \wedge \Box(\Box(\overline{x}) \vee q_v)$ holds. Finally, using equivalences (ii) and (iii), we obtain $\Box(q_v \leftrightarrow \Diamond(x)) \equiv (\Box(\neg q_v) \vee \Diamond(x)) \wedge (\Box(\overline{x}) \vee \Box(q_v))$.

Second step. We define an *i-literal* as an S5 formula having one of the forms l, $\Diamond l$, $\Box c$, where l is a literal and c is a basic clause. An *i-clause* is defined as a disjunction of i-literals.

It is worth noting that, using the first step in our transformation, each $A \in \mathcal{F}_{S5}$ can be transformed into an equi-satisfiable formula $I(A)$ consisting of a conjunction of i-clauses. Our aim in the second step is to transform each i-clause $I(A)$ into a hyb-clause. To this end, we use a function g associating with each subformula of the form $\Diamond l$ a nominal such that $g(\Diamond l) = g(\Diamond l')$ iff $l = l'$. We use $T(A)$ to denote the formula obtained from $I(A)$ by (i) replacing each occurrence of an i-literal of the form $\Diamond l$ with $g(\Diamond l) : l$, and (ii) replacing each literal l which is not in the scope of a modal connective with $a : l$ where a is a fresh nominal, i.e., there is no $\Diamond l$ such that $g(\Diamond l) = a$. Clearly, $T(A)$ is an S5-CNF formula.

Example 2. *Let us consider formula A described in Figure 1. We first have $S_A = \{\Box(q_{\Diamond_1} \leftrightarrow \Diamond q_{\wedge_1}), \Box(q_{\wedge_1} \leftrightarrow (q_{\Box_1} \wedge q_{\Diamond_2})), \Box(q_{\Box_1} \leftrightarrow \Box q_{\vee_1}), \Box(q_{\Diamond_2} \leftrightarrow \Diamond q_{\wedge_2}), \Box(q_{\vee_1} \leftrightarrow (p_1 \vee p_2)), \Box(q_{\wedge_2} \leftrightarrow (p_1 \wedge \neg p_2))\}$. After the first step, $I(A)$ corresponds to the following formula:*

$$\begin{gathered} q_{\Diamond_1} \wedge (\Box\neg q_{\Diamond_1} \vee \Diamond q_{\wedge_1}) \wedge (\Box q_{\Diamond_1} \vee \Box\neg q_{\wedge_1}) \wedge (\Box(\neg q_{\wedge_1} \vee q_{\Box_1})) \wedge (\Box(\neg q_{\wedge_1} \vee q_{\Diamond_2}))\wedge \\ (\Box(q_{\wedge_1} \vee \neg q_{\Box_1} \vee \neg q_{\Diamond_2})) \wedge (\Box\neg q_{\Box_1} \vee \Box q_{\vee_1}) \wedge (\Box q_{\Box_1} \vee \Diamond\neg q_{\vee_1}) \wedge (\Box\neg q_{\Diamond_2} \vee \Diamond q_{\wedge_2})\wedge \\ (\Box q_{\Diamond_2} \vee \Box\neg q_{\wedge_2}) \wedge (\Box(\neg q_{\vee_1} \vee p_1 \vee p_2)) \wedge (\Box(q_{\vee_1} \vee \neg p_1)) \wedge (\Box(q_{\vee_1} \vee \neg p_2))\wedge \\ (\Box(\neg q_{\vee_1} \vee p_1)) \wedge (\Box(\neg q_{\vee_1} \vee \neg p_2)) \wedge (\Box(q_{\vee_1} \vee \neg p_1 \vee p_2)) \end{gathered}$$

Then, using the second step in our transformation, $T(A)$ corresponds to the following formula:

$$\begin{gathered} \underline{a : q_{\Diamond_1}} \wedge (\Box\neg q_{\Diamond_1} \vee \underline{b_1 : q_{\wedge_1}}) \wedge (\Box q_{\Diamond_1} \vee \Box\neg q_{\wedge_1}) \wedge (\Box(\neg q_{\wedge_1} \vee q_{\Box_1})) \wedge (\Box(\neg q_{\wedge_1} \vee q_{\Diamond_2}))\wedge \\ (\Box(q_{\wedge_1} \vee \neg q_{\Box_1} \vee \neg q_{\Diamond_2})) \wedge (\Box\neg q_{\Box_1} \vee \Box q_{\vee_1}) \wedge (\Box q_{\Box_1} \vee \underline{b_2 : \neg q_{\vee_1}}) \wedge (\Box\neg q_{\Diamond_2} \vee \underline{b_3 : q_{\wedge_2}})\wedge \\ (\Box q_{\Diamond_2} \vee \Box\neg q_{\wedge_2}) \wedge (\Box(\neg q_{\vee_1} \vee p_1 \vee p_2)) \wedge (\Box(q_{\vee_1} \vee \neg p_1)) \wedge (\Box(q_{\vee_1} \vee \neg p_2))\wedge \\ (\Box(\neg q_{\vee_1} \vee p_1)) \wedge (\Box(\neg q_{\vee_1} \vee \neg p_2)) \wedge (\Box(q_{\vee_1} \vee \neg p_1 \vee p_2)) \end{gathered}$$

The underlined items are simply meant to emphasize hyb-literals governed by a nominal.

Proposition 3. *An S5 formula A is satisfiable iff $T(A)$ is satisfiable.*

Proof. Using Proposition 2, we know that A is satisfiable iff $I(A)$ is satisfiable. Thus, it suffices to show that $I(A)$ is satisfiable iff $T(A)$ is satisfiable. Let us first consider the "if" part. Let $(\mathcal{M}, f)$ be a HybS5-interpretation satisfying $T(A)$. Without loss of generality, let $\alpha = l_1 \vee \cdots \vee l_k \vee \Diamond l'_1 \vee \cdots \vee \Diamond l'_m \vee \Box c_1 \vee \cdots \vee \Box c_n$ be a conjunct in $I(A)$, i.e., it is an i-clause. Then $\alpha' = a : l_1 \vee \cdots \vee a : l_k \vee a_1 : l'_1 \vee \cdots \vee a_m : l'_m \vee \Box c_1 \vee \cdots \vee \Box c_n$ is a hyb-clause in $T(A)$. From the definition of the satisfaction relation in HybS5, $(\mathcal{M}, f), f(a) \models \alpha$ holds. As a consequence, $\mathcal{M}$ is a HybS5-model of $I(A)$. We prove the "only if" part in the same way. Let $\mathcal{M} = (W, V)$ be an S5-model of $I(A)$. Then, there exists $w \in W$ s.t. $\mathcal{M}, w \models I(A)$. We define an $\mathcal{M}$-assignment f of $T(A)$ as follows. If a is the nominal associated with the literals which are not in the scope of a modal connective using g (see *second step*), then $f(a) = w$. Further, for every nominal b associated with an i-literal $\Diamond l$ using g, if there exists $\mathcal{M}, w' \models l$ then $f(b) = w'$, and $f(b) = w''$ for a world $w'' \in W$ otherwise. From the definition of f, one can easily see that $(\mathcal{M}, f)$ is a HybS5-model of $T(A)$ since $\mathcal{M}, w \models I(A)$. $\Box$

4 An Example of Modeling in S5-SAT

In this section, we use an example in order to show that S5-SAT is suitable for modeling NP-complete decision problems. In particular, compared to SAT, we show that it allows one to use fewer propositional variables and clauses for modelling such a problem. The considered problem in our example is the graph coloring problem which is well-known to be NP-hard [11].

Let k be a strictly positive integer and $G = (V, E)$ an undirected graph such that V is the set of vertices and E is the set of edges. A k-coloring of G corresponds to a partition of V into k sets $P = \{V_1, \ldots, V_k\}$ such that no two vertices in the same subset are adjacent, i.e., for all $1 \leq i \leq k$ and for all $v, v' \in V_i$ with $v \neq v'$ we have $(v, v') \notin E$. Each subset in P corresponds to a different color. The *chromatic number* of a graph G, denoted by $\chi(G)$, is defined as the smallest number of colors needed to color G. Given a graph G, the minimum graph coloring problem for G consists of finding a k-coloring such that $k = \chi(G)$.

Given a strictly positive integer k and an undirected graph $G = (V, E)$, we associate with each vertex v in V a distinct propositional variable, denoted by p_v, and with each integer $i \in \{1, \ldots, k\}$ a distinct nominal, denoted by a_i. The following S5-CNF formula expresses that all the vertices of V have to be colored:

$$\bigwedge_{v \in V} \bigvee_{i=1}^{k} a_i : p_v \tag{1}$$

Further, the following S5-CNF formula expresses that no two adjacent vertices have the same color:

$$\bigwedge_{(v,v') \in E} \Box(\neg p_v \vee \neg p_{v'}) \tag{2}$$

In other words, the first S5-CNF formula expresses that, for all $v \in V$, the propositional variable p_v has to be true in at least one world from the worlds named by the nominals (colors) $a_1, \ldots, a_n$, and the second S5-CNF formula expresses that there is no world (color) where two propositional variables associated with two adjacent vertices are both true. Clearly, the conjunction of these two S5-CNF formulæ is satisfiable if and only if G admits a k-coloring.

The graph coloring problem can be encoded in SAT in the same way as in S5-SAT. However, we have to associate with each vertex $v \in V$ and each color $i \in \{1, \ldots, k\}$ a propositional variable to express that node v has color i. This leads to the use of $|V| \times k$ propositional variables in SAT. On the other hand, we only need to use $|V|$ propositional variables in our S5-SAT encoding. Moreover, we have to use $|E| \times k$ clauses to express that there are no two adjacent vertices having the same color in SAT. Again, we only need to use $|E|$ clauses in our S5-SAT encoding. This example illustrates the interest in using S5-SAT for modeling NP-complete problems.

5 Resolution Rules

In propositional logic, the resolution method is a decision procedure for deciding whether a formula in conjunctive normal form is unsatisfiable. It is based on the following inference rule:

$$\frac{\alpha \vee l \qquad \alpha' \vee \bar{l}}{\alpha \vee \alpha'} \ [Res]$$

The clause $\alpha \vee \alpha'$ is called the *resolvent* of $\alpha \vee l$ and $\alpha' \vee \bar{l}$. The resolution inference rule states that the resolvent $\alpha \vee \alpha'$ is a logical consequence of the set of clauses $\{\alpha \vee l, \alpha' \vee \bar{l}\}$. A *resolution derivation* of a conclusion C (a clause) from a set of premises Δ (a set of clauses) is a tree[1] such that the root node is labelled with C, the labels at the immediate successors of a node n are the premises of an instance of the resolution rule having the label at n as a resolvent, and the leaf nodes are labelled with clauses in Δ. The resolution method is mainly used to show unsatisfiability of CNF formulæ. Indeed, a set of clauses Δ is unsatisfiable if and only if there is a resolution derivation of the empty clause from Δ.

In the same vein, we define here a resolution method for modal logic S5 using the S5 conjunctive normal form defined earlier. Our method is based on the rules described in Figure 2. The premises and the conclusion of each rule are hyb-clauses. The empty hyb-clause is denoted by $\{\}$. It is worth noting that if a hyb-clause contains a hyb-literal with an empty set of literals in the scope of $\Box$, then this hyb-literal is removed.

$$\frac{\alpha \vee a : l \qquad \alpha' \vee a : \bar{l}}{\alpha \vee \alpha'}\ [Res1] \qquad \frac{\alpha \vee a : l \qquad \alpha' \vee \Box(l_1 \vee \cdots \vee l_n \vee \bar{l})}{\alpha \vee \alpha' \vee a : l_1 \vee \cdots \vee a : l_n}\ [Res2]$$

$$\frac{\alpha \vee \Box(\beta \vee l) \qquad \alpha' \vee \Box(\beta' \vee \bar{l})}{\alpha \vee \alpha' \vee \Box(\beta \vee \beta')}\ [Res3]$$

Figure 2: Resolution rules for S5

Example 3. *We provide here a resolution derivation of the formula* $(\Box(\neg p \vee q)) \wedge (\Box(\neg p \vee r)) \wedge (\Box p) \wedge (a : \neg q \vee b : \neg r)$*:*

$$\dfrac{\dfrac{\dfrac{\Box p \quad \Box(\neg p \vee q)}{\Box q}\,[Res3] \qquad a : \neg q \vee b : \neg r}{b : \neg r}\,[Res2] \qquad \dfrac{\Box p \quad \Box(\neg p \vee r)}{\Box r}\,[Res3]}{\{\}}\,[Res2]$$

Theorem 2 (Soundness)**.** *The rules $Res1$, $Res2$, and $Res3$ are sound.*

Proof. The soundness of the rules $Res1$, $Res2$, and $Res3$ means that, for each rule, the resolvent is a logical consequence of the premises. In the three rules, the soundness stems from the fact that we can not satisfy a literal and its complementary literal in any world. A more formal proof can be obtained by assuming that the premises are satisfied by a model $\mathcal{M}$ and showing that $\mathcal{M}$ satisfies also the resolvent. □

Using the soundness of $Res1$, $Res2$, and $Res3$, we know that if there is a resolution derivation of the empty hyb-clause from a set of hyb-clauses, then the set of hyb-clauses is unsatisfiable.

Given a resolution derivation $\mathcal{D}$ and a node v in $\mathcal{D}$, we use $leng(v, \mathcal{D})$ to denote the number of nodes between the root and v.

The following proposition is used to show completeness. Intuitively, it states that if there is a resolution proof showing that an S5-CNF formula is unsatisfiable, then, for every S5-CNF formula obtained by using the modal connective $\Box$ instead of nominals, there is also a resolution

[1] In the literature, a resolution derivation is also defined as a DAG in which each clause appears once.

proof showing that this formula is unsatisfiable. In what follows, symbol $\uplus$ denotes the disjoint union operator.

Proposition 4. *If there is a resolution derivation of the empty hyb-clause from* $\Delta \uplus \{\alpha \vee a : c\}$, *then there is a resolution derivation of the empty hyb-clause from* $\Delta \uplus \{\alpha \vee \Box c\}$, *where for* $c = l_1 \vee \cdots \vee l_k$, $a : c$ *denotes* $a : l_1 \vee \cdots \vee a : l_k$.

Proof. Let $\mathcal{D}$ be a resolution derivation of the empty hyb-clause from $\Delta \uplus \{\alpha \vee a : c\}$. Our proof is obtained by showing the following property: ($Prop$) for each node v in $\mathcal{D}$ labelled with $\gamma = \beta \vee a : l_1 \vee \cdots \vee a : l_k$, where a does not appear in β, there exists $S' \subseteq S = \{l_1, \ldots, l_k\}$ such that there is a resolution derivation of $\beta \vee (\bigvee_{l \in S \setminus S'} a : l) \vee \Box(\bigvee_{l \in S'} l)$ from $\Delta \uplus \{\alpha \vee \Box c\}$. We show this property by induction on $leng(v, \mathcal{D})$. If $leng(v, \mathcal{D}) = 1$ then v is a leaf node labelled with either a hyb-clause in Δ or $\alpha \vee a : c$. In the case where v is labelled with $\alpha \vee a : c$, $Prop$ is obtained using the hyb-clause $\alpha \vee \Box c$. Regarding the case where $leng(v, \mathcal{D}) > 1$ (v is an internal node), since the label of v is a resolvent in an instance $IRes$ of a resolution rule, we need to distinguish the three cases of the resolution rules. We first have to apply the induction hypothesis on the premises of $IRes$. Then, we apply a resolution rule according to the complementary literals used to perform resolution in $IRes$. $\Box$

Proposition 5. *Let* Δ *be a set of hyb-literals.* Δ *is unsatisfiable iff there is a resolution derivation of the empty hyb-clause from* Δ.

Proof. The "if" part is obtained from Theorem 2. To show the "only if" part we use the completeness of the resolution rule in propositional logic. Without loss of generality, we consider Δ to be equal to $\{a_1 : l^1_1, \ldots, a_1 : l^1_{k_1}, \ldots, a_n : l^n_1, \ldots, a_n : l^n_{k_n}, \Box c_1, \ldots, \Box c_m\}$, where for all $1 \leq i \neq j \leq n$, $a_i \neq a_j$. Clearly, Δ is unsatisfiable iff one of the following sets of clauses is unsatisfiable: $\{l^1_1, \ldots, l^1_{k_1}, c_1, \ldots, c_m\}, \ldots, \{l^n_1, \ldots, l^n_{k_n}, c_1, \ldots, c_m\}$. Indeed, every set of interpretations satisfying all the previous sets of clauses can be used to build an interpretation satisfying Δ. Let us assume that the set of clauses $P_i = \{l^i_1, \ldots, l^i_{k_i}, c_1, \ldots, c_m\}$ is unsatisfiable for $i \in \{1, \ldots, n\}$. Thus, using the completeness of the resolution method in propositional logic, there is a resolution derivation $\mathcal{D}$ of the empty clause from P_i. Using the resolution rule $Res1$, we know that $\mathcal{D}$ can be directly transformed into a resolution derivation $\mathcal{D}$ of the empty hyb-clause from $\{a_i : l^i_1, \ldots, a_i : l^i_{k_i}, a_i : c_1, \ldots, a_i : c_m\}$. Thus, using Proposition 4, there is a resolution derivation of the empty hyb-clause from Δ_i. $\Box$

Theorem 3 (Completeness)**.** *Let* Δ *be a set of hyb-clauses. If* Δ *is unsatisfiable then there is a resolution derivation of the empty hyb-clause from* Δ.

Proof. Our proof of completeness is obtained by induction on the value $NHL(\Delta)$. The proof regarding the base case where $NHL(\Delta) = 0$ is a direct consequence of Proposition 5. Let us consider the case where $NHL(\Delta) > 0$. Then, there exists a clause α in Δ such that $NHL(\alpha) \geq 2$. Let h be a hyb-literal in α ($\alpha \equiv h \vee \beta$). Since Δ is unsatisfiable, we know that the two S5-CNF formulæ $\Delta_1 = (\Delta \setminus \alpha) \cup \{h\}$ and $\Delta_2 = (\Delta \setminus \alpha) \cup \{\beta\}$ are unsatisfiable. Moreover, we have $NHL(\Delta_1) < NHL(\Delta)$ and $NHL(\Delta_2) < NHL(\Delta)$. By applying the induction hypothesis, there are resolution derivations $\mathcal{D}_1$ and $\mathcal{D}_2$ of the empty hyb-clause from Δ_1 and Δ_2, respectively. We use $V(\mathcal{D}_2)$ to denote the set of the nodes of $\mathcal{D}_2$ belonging to a path from the root node to a node labelled with β. Let $\mathcal{D}'_2$ be the tree obtained from $\mathcal{D}$ by adding the hyb-literal h to the hyb-clauses of the nodes in $V(\mathcal{D}_2)$. From the structure of the resolution rules, we have that $\mathcal{D}'_2$ is a resolution derivation of either the empty hyb-clause or h from Δ. In the case where the root node is labelled with the empty hyb-clause, $\mathcal{D}'_2$ allows us to show our property. Otherwise, Let $\mathcal{D}'_1$ be the resolution tree obtained from $\mathcal{D}_1$ by replacing the leaf nodes labelled with h with $\mathcal{D}_2'$. As a consequence, $\mathcal{D}_1'$ is a resolution derivation of the empty hyb-clause from Δ. □

6 Conclusion and Perspectives

The contribution of this work consists of defining a conjunctive normal form for S5, analogous to that for propositional logic, and providing a fairly simple resolution method. A formula in our normal form consists of a conjunction of conjunction-free and $\Diamond$-free clauses. Further, our conjunctive normal form for S5 allows for a more compact encoding of propositional variables and clauses for modelling NP-complete problems than the one obtained in SAT. Our resolution method comprises three resolution rules which are defined in the same way as Robinson's resolution rule.

For future work, we intend to implement an automated theorem-prover for S5 based on our resolution method. To this end, we plan to propose different strategies for rule instances selection. We also plan to extend our approach for defining resolution methods for other normal modal logics as well.

References

[1] M. Abadi and Z. Manna. Modal theorem proving. In *8th International Conference on Automated Deduction*, volume 230 of *LNCS*, pages 172–189. Springer, 1986.

[2] C. Areces, M. de Rijke, and H. de Nivelle. Resolution in modal, description and hybrid logic. *J. Log. Comput.*, 11(5):717–736, 2001.

[3] C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*. Elsevier, 2006.

[4] Y. Auffray, P. Enjalbert, and J.-J. Hébrard. Strategies for modal resolution: Results and problems. *J. Autom. Reasoning*, 6(1):1–38, 1990.

[5] P. Blackburn. Internalizing labelled deduction. *J. Log. Comput.*, 10(1):137–168, 2000.

[6] M. D'Agostino, D. M. Gabbay, R. Hähnle, and J. Posegga, editors. *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.

[7] P. Enjalbert and L. F. del Cerro. Modal resolution in clausal form. *Theoretical Computer Science*, 65(1):1 – 33, 1989.

[8] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press Books. The MIT Press, 2003.

[9] M. Fitting. *Proof methods for modal and intuitionistic logics*, volume 169 of *Synthese Library*. Kluwer, 1983.

[10] M. Fitting. A simple propositional S5 tableau system. *the Annals of Pure and Applied Logic*, 96:107–115, 1999.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

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

[13] H. J. Ohlbach. Semantics-based translation methods for modal logics. *Journal of Logic and Computation*, 1(5):691–746, 1991.

[14] G. S. Tseitin. On the complexity of derivations in the propositional calculus. In H.A.O. Slesenko, editor, *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.