24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

# Consolidating Modal Knowledge Bases

**Zied Bouraoui**[1] and **Jean-Marie Lagniez**[2] and **Pierre Marquis**[3] and **Valentin Montmirail**[4]

**Abstract.** This paper introduces a novel approach to the consolidation of knowledge bases represented as modal logic formulae. The objective is to turn the given knowledge base into another knowledge base such that the latter is consistent even when the former is not. Our approach follows a strategy that locally spots and iteratively consolidates inconsistent subformulae of the input knowledge base. Existing methods for consolidating a knowledge base typically consist in selecting some of its maximal consistent subbases. Such methods are suited to the case the input is a (conjunctively-interpreted) set of formulae. However, they are inadequate when the input consists of a single inconsistent modal formula since, in the modal case, a formula cannot always be turned into a conjunction of simpler formulae. Furthermore, such methods consolidate any base consisting of a single inconsistent formula into the empty base. Our approach does not suffer from such limitations and preserves more information in the general case. From a computational point of view, it ensures that the size of the consolidated base is bounded by the size of the input knowledge base. We present some empirical results demonstrating the practical feasibility of our approach.

## 1 INTRODUCTION

Logical deduction is the key inference mechanism to draw sound conclusions from a knowledge base (a conjunctively-interpreted set of formulae). It provides natural explanations for the consequences that can be derived. However, when the knowledge base under consideration is inconsistent, logical deduction is no longer appropriate, because it trivializes: every formula can be derived from an inconsistent base, including the conflicting information causing the inconsistency (*ex falso quodlibet sequitur*).

The problem of inconsistency management has received considerable attention in a wide variety of areas, including databases (e.g., [2, 26, 6]), ontology-based query answering (e.g., [11, 3]), multiagent systems (e.g., [19]), description logics (e.g., [17, 36]), belief merging and revision (e.g., [16, 33]).

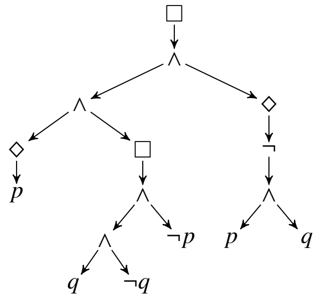
Many approaches have been designed so far to avoid the trivialization of inference when the knowledge base is inconsistent (e.g., [21, 13, 5]). Some of them, based on paraconsistent logics, either consider weakened proof systems, or non-classical semantical settings to ensure that an inconsistent formula has some models even if none of them is classical (e.g., [32, 30, 8, 18, 9]). Such approaches lead to inference relations that are not necessarily explosive when the input is an inconsistent formula, but they suffer from a couple of drawbacks: in general, they do not preserve the set of logical consequences of the input even when it is consistent and they do not guarantee that the set of consequences of the input is classically consistent. This prevents from reasoning classically from it.

Another family of approaches for reasoning under inconsistency gather approaches that weaken the input base instead of weakening the deduction relation. Several weakening mechanisms can be considered, the simplest one being formula inhibition (i.e., the removal of some formulae from the knowledge base when it is inconsistent). Such approaches also have some pros and some cons. From the positive side, they lead to knowledge bases that are classically consistent. As such, they can be used for the consolidation purpose. However, they do not avoid the trivialization problem when the knowledge base consists of a single formula. Indeed, if this formula is inconsistent, removing it from the base leads to a consolidated base, which is consistent, but empty. Accordingly, every piece of information contained in the input base is lost and the set of consequences of the consolidated base consists only of the valid formulae. It must be noted that the single formula situation cannot always be avoided, for two main reasons. On the one hand, though in classical logic settings, a formula can be decomposed into the conjunction of simpler formulae (especially, a conjunction of clauses), this is not the case in every logic, especially in modal logics [10]. On the other hand, replacing in a knowledge base the conjunction of two subformulae by the subformulae themselves is not neutral from the inference point of view [20]. Formulae that “come together” (i.e., that are linked by a conjunction in the knowledge base) may come from the same source of information so that they should not always be split into independent pieces of information (in some cases, they must be kept or removed as a whole).

A simple approach from this family consists in keeping only in the consolidated base the “free formulae” of the input base, i.e., those formulae that are not involved in any conflict. Said differently, this approach consists in removing all formulae participating in an inconsistency. Doing so, the loss of information could be considerable (which has a direct impact on the set of conclusions that can be drawn from the consolidated base), but from a computational point of view the size of the consolidated base remains bounded by the size of the input base, which is a desirable feature. More sophisticated approaches, used for instance in default reasoning [31], consist in focusing on maximal consistent subsets (i.e., maximal with respect to set inclusion) of the input base, and in taking advantage of all / some of them (the preferred ones, often characterized using a plausibility ordering over formulae) to draw conclusions. In this case, the consolidated base is equivalent to the disjunction of all preferred maximal consistent subsets). Compared to the “free formulae” approach, such approaches lead to preserve more information from the input base, but do not offer any polynomial guarantee on the size of the consolidated base. Indeed, a knowledge base can have exponentially many maximal consistent subsets.

[1] CRIL, Univ. Artois & CNRS, France, bouraoui@cril.fr
[2] Huawei Paris Research Center, France, jean-marie.lagniez@huawei.com
[3] CRIL, IUF, Univ. Artois & CNRS, France, marquis@cril.fr
[4] Avisto Telecom, IIoT Solutions Team, Vallauris, France, valentin.montmirail@avisto.com

**Figure 1**: $\varphi = \Box(\wedge(\wedge(\Diamond(p), \Box(\wedge(\wedge(q, \neg q), \neg p))), \Diamond(\neg(\wedge(p, q)))))$



In this paper, we present an approach to the consolidation of an inconsistent modal knowledge base. We focus on single-agent normal modal logics (e.g., [12]), the family of logics based on modal logic K. In this setting, a formula $\varphi$ cannot always be turned into a conjunctively-interpreted set $S$ of subformulae of $\varphi$ such that $S$ is equivalent to $\varphi$ (even though some existing works exist in this direction [28]). As a consequence, in the case when the knowledge base consists of a single inconsistent modal formula, approaches to consolidation based on free formulae or maximal consistent subsets (as mentioned above) will lead to an empty consolidated base.

Our approach to consolidation is parametrized by a weakening mechanism (the simplest one consisting in replacing subformulae by the Boolean constant always true) and it outputs a knowledge base that is not empty even when its input consists of a single, inconsistent formula. In a nutshell, our approach consists in consolidating subformulae of the input formula in an iterative fashion until recovering consistency. When the weakening mechanism that is exploited outputs a weakened formula that is of smaller size than the size of its input, our approach guarantees that the size of the consolidated knowledge base to be generated is bounded by the size of the input formula.

**Example 1.** *As a matter of illustration, consider the formula $\varphi$ represented in functional notation by $\varphi = \Box(\wedge(\wedge(\Diamond(p), \Box(\wedge(\wedge(q, \neg q), \neg p))), \Diamond(\neg(\wedge(p, q)))))$ and depicted hereafter ($p$ and $q$ are propositional symbols).*

*When interpreted as a formula from* K *(the basic modal logic [12]), $\varphi$ is inconsistent. It can be observed that the subformula $\wedge(\Diamond(p), \Box(\wedge(\wedge(q, \neg q), \neg p)))$ is inconsistent since at the same time $p$ is possible and $\wedge(\wedge(q, \neg q), \neg p) \equiv \bot$ is necessary, which cannot hold. Actually, this inconsistency impacts the whole formula $\varphi$, rendering it inconsistent as well. On the other hand, the subformula $\Diamond(\neg(\wedge(p, q)))$ is not involved in any inconsistency. If the free formula approach or the maximal consistent subsets approach is used for consolidating the knowledge base $K = \{\varphi\}$, then every piece of information will be lost ($\varphi$ is not a free formula of $K$, hence the consolidated base will be the empty set of formulae).*

*An alternative, yet more desirable result would consist in replacing $\varphi$ by its "free subformula" $\Diamond(\neg(\wedge(p, q)))$. Doing so, more information would be preserved by the consolidation process though avoiding the size of the consolidated base to blow up.*

## 2 BACKGROUND

In this section, we give some background on normal modal logics (for more details see [12]). We use here the standard modal logic syntax and Kripke semantics [15]. In particular, the *language* $\mathcal{L}$ of modal logic is based on $\mathbb{P}$ a countably infinite set of propositional variables ($p, q, \dots$) and the logical symbols (connectives and modalities) from $\{\neg, \wedge, \vee, \Box, \Diamond, \top, \bot\}$. We use Greek letters $\alpha, \beta, \dots$ to denote modal logic formulae. We also use the standard abbreviation $\bot \overset{\text{def}}{=} \neg\top$ and $\Diamond\varphi \overset{\text{def}}{=} \neg\Box\neg\varphi$. The size of a formula $\varphi$, denoted $\mathtt{size}(\varphi)$, is defined inductively as follows:

$$\begin{aligned}
&\mathtt{size}(\top) = \mathtt{size}(p) = 1\\
&\mathtt{size}(\varphi_1 \wedge \varphi_2) = \mathtt{size}(\varphi_1 \vee \varphi_2) = 1 + \mathtt{size}(\varphi_1) + \mathtt{size}(\varphi_2)\\
&\mathtt{size}(\Box\varphi_1) = \mathtt{size}(\Diamond\varphi_1) = \mathtt{size}(\neg\varphi_1) = 1 + \mathtt{size}(\varphi_1)
\end{aligned}$$

The satisfiability/unsatisfiability of a modal formula is defined in terms of "model" and "satisfaction relation". A "model" of a modal logic formula is a Kripke structure that satisfies the formula [22]. A *Kripke structure* is a triple $\mathcal{K} = \langle W, R, V\rangle$, where $W$ is a non-empty set of possible worlds, $R \subseteq W \times W$ is a binary accessibility relation and $V : \mathbb{P} \to 2^W$ is a valuation function which associates, with each $p \in \mathbb{P}$, the set of possible worlds from $W$ where $p$ is true. A *pointed Kripke structure* is a pair $\langle\mathcal{K}, \omega\rangle$, where $\mathcal{K}$ is a Kripke structure and $\omega$ is a possible world in $W$. Throughout the paper, in order to alleviate the phrasing, one uses the term "Kripke structure" to refer to "pointed Kripke structure".

The satisfaction relation $\models$ between Kripke structures and formulae in $\mathcal{L}$ is defined inductively as follows:

$$\begin{array}{lll}
\langle\mathcal{K}, \omega\rangle \models \top & & \\
\langle\mathcal{K}, \omega\rangle \models p & \text{iff} & \omega \in V(p)\\
\langle\mathcal{K}, \omega\rangle \models \neg\varphi & \text{iff} & \langle\mathcal{K}, \omega\rangle \not\models \varphi\\
\langle\mathcal{K}, \omega\rangle \models \varphi_1 \wedge \varphi_2 & \text{iff} & \langle\mathcal{K}, \omega\rangle \models \varphi_1 \text{ and } \langle\mathcal{K}, \omega\rangle \models \varphi_2\\
\langle\mathcal{K}, \omega\rangle \models \varphi_1 \vee \varphi_2 & \text{iff} & \langle\mathcal{K}, \omega\rangle \models \varphi_1 \text{ or } \langle\mathcal{K}, \omega\rangle \models \varphi_2\\
\langle\mathcal{K}, \omega\rangle \models \Box\varphi & \text{iff} & \forall\omega'\text{s.t.}(\omega, \omega') \in R \text{ implies } \langle\mathcal{K}, \omega'\rangle \models \varphi\\
\langle\mathcal{K}, \omega\rangle \models \Diamond\varphi & \text{iff} & \exists\omega'\text{s.t.}(\omega, \omega') \in R \text{ and } \langle\mathcal{K}, \omega'\rangle \models \varphi
\end{array}$$

A formula $\varphi \in \mathcal{L}$ is **valid** (denoted by $\models \varphi$) iff it is satisfied by all Kripke structures $\langle\mathcal{K}, \omega\rangle$. A formula $\varphi \in \mathcal{L}$ is **satisfiable** iff $\neg\varphi$ is not valid (denoted $\not\models \neg\varphi$). A Kripke structure that satisfies $\varphi$ is called a **model** of $\varphi$. A formula $\varphi \in \mathcal{L}$ is **unsatisfiable** iff $\varphi$ is not satisfiable.

We assume the reader familiar with the different schemata of modal logic (see [34, Table 25.2]). Let $\star$ be any of the 15 types of structures given in [34, Table 25.2]. A formula $\varphi \in \mathcal{L}$ is $K\star$-**valid** (noted $\models_{K\star} \varphi$) iff it is satisfied by all $K\star$-structures $\langle\mathcal{K}, \omega\rangle$. A formula $\varphi \in \mathcal{L}$ is $K\star$-**satisfiable** iff $\not\models_{K\star} \neg\varphi$. A $K\star$-structure that satisfies a formula $\varphi$ is called a $K\star$-**model** of $\varphi$.

We are now ready to introduce a number of definitions needed to present our consolidation approach, which is based on the notion of subformula. Let us first recall that a formula $\varphi$ from $\mathcal{L}$ can be represented as a tree, where each node is labelled by a symbol (that can be a modality, a connective among $\{\wedge, \vee, \neg\}$, $\top$ or a propositional variable). Each node $N$ of the tree is characterized by a (unique) path to be followed to reach it. This path is given by a word defined over the alphabet $\boldsymbol{N}^*$ of positive integers. In this paper, we consider the alphabet $\{1, 2\}$, which is enough given the arities of the connectives under consideration. A word is used to make precise the path followed from the root of the tree in order to reach a node. $\cdot$ denotes the concatenation of words.

**Definition 1** (Occurrence)**.** *Let $\varphi$ be a formula from $\mathcal{L}$. The* occurrence *of node $N$ in the tree associated with $\varphi$ is the word $o_N$ over $\boldsymbol{N}^*$ given by:*

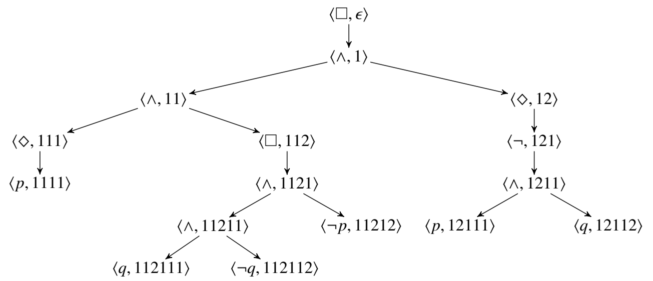
- *the root of the tree has occurrence $\epsilon$ (the empty word)*

- *if $N$ is not the root of the tree, then $N$ has a father node $M$ with occurrence $o_M$, and $o_N$ is defined by*
  - *if the label of $M$ is $\diamond$, $\Box$, or $\neg$, then $o_N = o_M.1$,*
  - *if the label of $M$ is $\wedge$ or $\vee$, then $o_N = o_M.1$ if $N$ corresponds to the first argument of the connective and $o_N = o_M.2$ otherwise.*

$O_\varphi$ denotes the set of occurrences in the tree-based representation of $\varphi$. If $o \in O_\varphi$, then $l(o, \varphi)$ denotes the label of the node at occurrence $o$ in $\varphi$, and $\varphi_o$ denotes the subformula of $\varphi$ rooted in $\varphi$ at occurrence $o$ (when $o \neq \epsilon$, $\varphi_o$ is said to be a strict subformula of $\varphi$).

**Example 2** (cont'ed). *Fig.2 depicts $\varphi$ with its occurrence information represented as an ordered pair associating with each node its label and its occurrence. We have $O_\varphi = \{\epsilon, 1, 11, 111, 1111, 112, 1121, 11211, 112111, 112112, 11212, 12, 121, 1211, 12111, 12112\}$. The label $l(112, \varphi)$ of the node of $\varphi$ at occurrence 112 is $\Box$. The root node of the subformula $\diamond(p)$ of $\varphi$ is at occurrence 111 in $\varphi$, and the root node of the subformula $\wedge(p, q)$ in $\varphi$ is at occurrence 1211. Stated otherwise, we have $\varphi_{111} = \diamond(p)$ and $\varphi_{1211} = \wedge(p, q)$.*

**Figure 2**: $\varphi$ with its occurrences.



We will also need the following notation. If $o$ is any word over $\boldsymbol{N}^*$ and $\{o_1, o_2, \ldots, o_k\}$ is a set of such words, then $o \times \{o_1, o_2, \ldots, o_k\}$ denotes the set of words $\{o.o_1, o.o_2, \ldots, o.o_k\}$.

## 3 AN APPROACH TO CONSOLIDATION

In this section, we present our approach to the consolidation of a modal formula. Consider a formula $\varphi$, the set $O_\varphi$ of occurrences can thus be partitioned into two subsets: i) $I(O_\varphi) = \{o \in O_\varphi \mid \varphi_o$ inconsistent$\}$, the subsets of occurrence of the nodes of $\varphi$ which are roots of inconsistent subformulae, and $I(O_\varphi)^c = O_\varphi \setminus I(O_\varphi)$ the complementary set of $I(O_\varphi)$. The set of inconsistent subformulae of $\varphi$ is then defined by $IS(\varphi) = \{\varphi_o \mid o \in I(O_\varphi)\}$.

**Example 3** (cont'ed). *We have $I(O_\varphi) = \{\epsilon, 1, 11, 112, 1121, 11211\}$. Thus $\varphi$ has six inconsistent subformulae: $\varphi = \varphi_\epsilon$ (the formula itself), $\varphi_1$, $\varphi_{11}$, $\varphi_{112}$, $\varphi_{1121}$ and $\varphi_{11211}$. The other subformulae of $\varphi$ are consistent.*

Notice that in some cases, $IS(\varphi)$ may be not empty even though $\varphi$ is consistent. Consider for instance the formula $\varphi = \vee(\wedge(p, \neg p), q)$. Clearly, $\varphi$ is consistent, but we have $IS(\varphi) = \wedge(p, \neg p)$. In this case, $\varphi$ can be simplified by replacing every inconsistent subformula by its equivalent formula $\neg\top$ while preserving logical equivalence. However, in this work, we are not primarily interested in formulae $\varphi$ from which inference does not trivialize, we rather focus on the case $\varphi$ is inconsistent.

Let us now explain how the input formula $\varphi$ can be consolidated by "partially removing" inconsistent subformulae. To do so, we assume that a **weakening** mechanism $wm$ is available. Given any formula $\varphi$, $wm(\varphi)$ returns a logical consequence of $\varphi$ such that $\texttt{size}(wm(\varphi)) \leq \texttt{size}(\varphi)$. Defining $wm^1(\varphi)$ as $wm(\varphi)$ and inductively $wm^{k+1}(\varphi)$ as $wm(wm^k(\varphi))$ for any integer $k \geq 0$, $wm$ is supposed to be such that for any formula $\varphi$ there exists an integer $n_\varphi$ such that $wm^{n_\varphi}(\varphi)$ is valid.

Many such weakening mechanisms can be defined. For the sake of simplicity, in the following, we consider the drastic weakening mechanism $dwm$ such that for any formula $\varphi$, we have $dwm(\varphi) = \top$.

Clearly enough, the inconsistent subformulae of the input formula that need to "removed" should be chosen with care. Thus, on the running example, if $\varphi$ itself is first replaced by $\top$, then the inconsistent subformulae $\varphi_1$ and $\varphi_{11}$ of $\varphi$ are not any longer subformulae of the result, hence the other replacement operations cannot take place. On the other hand, replacing $\varphi$ by $\top$ is often too drastic since it leads to lose all the information from $\varphi$: replacing $\varphi_{11} = \wedge(\Box(p), \diamond(\neg(p)))$ in $\varphi$ leads to the consolidated formula $\Box(\wedge(\top, \diamond(\neg(\wedge(p, q)))))$ which is consistent and contains more information than $\top$.

We now formally present our approach. Let us first recall that a *prefix* of a word $o$ is any word $o'$ such that there exists a word $o''$ satisfying $o = o' \cdot o''$, denoted $o' \sqsubseteq o$. A strict prefix $o'$ of $o$ is a prefix of $o$ such that $o'' \neq \epsilon$, denoted $o' \sqsubset o$. Clearly, $\sqsubseteq$ is a pre-order over $\boldsymbol{N}^*$ (i.e.,, a reflexive and transitive relation).

**Definition 2** (Prefix-Closed Set). *Let $\varphi$ be a formula from $\mathcal{L}$. A subset $S$ of $O_\varphi$ is* prefix-closed *iff for every $o \in S$ every prefix of $o$ also belongs to $S$.*

For instance, $\emptyset$ and $O_\varphi$ itself are prefix-closed. Given any subset $S$ of $\boldsymbol{N}^*$, $PC(S) = \{o \in S \mid$ every prefix of $o$ belongs to $S\}$ denotes the subset of $S$ containing occurrences such that all their prefixes belong to $S$. By construction, $PC(S)$ is prefix-closed.

**Example 4** (cont'ed). *$\{\epsilon, 1, 11, 111\}$ is prefix-closed but $\{\epsilon, 1, 11, 121\}$ is not. We have $PC(\{\epsilon, 1, 11, 121\}) = \{\epsilon, 1, 11\}$.*

We are now ready to define a notion of partial consolidation of a modal formula $\varphi$:

**Definition 3** (Partial Consolidation). *Let $\varphi$ be a formula from $\mathcal{L}$. Let*

$$R(\varphi) = max(PC(I(O_\varphi)), \sqsubseteq)$$

*be the set of occurrences to be "repaired". The* partial consolidation *of $\varphi$ for the weakening mechanism $wm$ is the formula $C_p(\varphi)$ of $\mathcal{L}$ obtained by replacing in $\varphi$ every subformula $\varphi_o$ occurring at some $o \in R(\varphi)$ by $wm(\varphi_o)$.*

**Example 5** (cont'ed). *Suppose that the weakening mechanism that is used is dwm. We have $PC(I(O_\varphi)) = \{\epsilon, 1, 11, 112, 1121, 11211\}$, hence $R(\varphi) = \{11211\}$. Accordingly, the partial consolidation of $\varphi$ for dwm is the formula $C_p(\varphi) = \Box(\wedge(\wedge(\diamond(p), \Box(\wedge(\top, \neg p))), \diamond(\neg(\wedge(p, q)))))$. Indeed, the only inconsistency pointed out by $R(\varphi)$ is $\wedge(q, \neg q)$, thus it is the only subformula of $\varphi$ replaced by $\top$ to keep as much information as possible.*

Intuitively, by only looking at the prefix-closed subset of $I(O_\varphi)$, we ensure that only the inconsistent subformulae of $\varphi$ that can be responsible for the inconsistency of $\varphi$, are considered. For instance, if $\varphi = \wedge(\vee(p, \wedge(q, \neg(q))), \wedge(r, \neg(r)))$, the inconsistent subformula $\wedge(q, \neg(q))$ is harmless, unlike the inconsistent subformula $\wedge(r, \neg(r))$, which makes $\varphi$ inconsistent. Replacing both inconsistent subformulae by $\top$ would lead to a valid formula, while the partial consolidation of $\varphi$ for $dwm$ is equivalent to $p$.

Considering in $PC(I(O_\varphi))$ only those occurrences which are maximal w.r.t. the prefix pre-order amounts to focusing in priority on the "deepest" inconsistent subformulae. The rationale for it is to keep as much information as possible in the partial consolidation.

Clearly enough, whenever $\varphi$ is consistent, we have $I(O_\varphi) = \emptyset$ so that $C_p(\varphi) = \varphi$. Now, in the general case when $\varphi$ is inconsistent, it is not always the case that the partial consolidation of a formula is a consolidation of $\varphi$ simply because it is not necessarily consistent. Indeed, repairing the ”deepest” inconsistent subformulae of $\varphi$ is not always enough to get a consistent formula. Stepping back to our running example, it is clear that the partial consolidation $\Box(\wedge(\wedge(\Diamond(p), \Box(\wedge(\top, \neg p))), \Diamond(\neg(\wedge(p, q)))))$ is still inconsistent. Thus, the partial consolidation process of $\varphi$ must be pursued until a consistent formula is reached. Formally:

**Definition 4** (Consolidation)**.** *Let $\varphi$ be a formula from $\mathcal{L}$. The* consolidation *of $\varphi$ for a weakening mechanism wm is the formula $C(\varphi)$ of $\mathcal{L}$ defined as $C_p^{min}(\varphi)$, where min is the least integer i such that $C_p^i(\varphi) = C_p^{i+1}(\varphi)$, and the sequence $(C_p^i(\varphi))_{i\in\mathbb{N}}$ is defined by*

- $C_p^0(\varphi) = \varphi$,
- $C_p^{i+1}(\varphi) = C_p(C_p^i(\varphi))$.

*where $C_p(.)$ computes the partial consolidation of its argument for wm.*

The following proposition holds.

**Proposition 1.** *Let $\varphi$ be a formula from $\mathcal{L}$. The consolidation $C(\varphi)$ of $\varphi$ for any weakening mechanism wm is a consistent formula.*

*Proof.* Given the condition imposed on *wm*, the fact that any valid is consistent and that the repair process achieved by $C_p$ consists in replacing some inconsistent subformulae of the input formula by a weakening of them, the termination condition $C_p^i(\varphi) = C_p^{i+1}(\varphi)$ is obtained when no replacement needs to be done in $C_p^i(\varphi)$, which implies that $I(O_{C_p^i(\varphi)}) = \emptyset$. In this case, $C_p^i(\varphi)$ is consistent hence so is $C(\varphi)$. □

Another important concept to be taken into account in a partial consolidation process is *monotonicity*. Monotone modal formulae from $\mathcal{L}$ are those not involving any occurrence of the negation connective. In the following, one needs a notion of monotone subset of occurrences of a formula:

**Definition 5** (Monotonicity)**.** *Let $\varphi$ be a formula from $\mathcal{L}$. A prefix-closed subset $S$ of $O_\varphi$ is* monotone *iff no element $o$ of $S$ is such that $l(o, \varphi) = \neg$.*

**Example 6** (cont'ed)**.** $\{\epsilon, 1, 11, 111\}$ *is prefix-closed but* $\{11, 111, 112\}$ *is not.* $\{\epsilon, 1, 11, 111\}$ *is monotone, while the prefix-closed subset of $O_\varphi$ given by* $\{\epsilon, 1, 12, 121\}$ *is not.*

It can be observed that weakening any strict subformula $\varphi'$ occurring at $o \neq \epsilon$ in a formula $\varphi$ such that $l(\epsilon, \varphi) = \neg$ does not ensure that the consistency of the overall formula has been restored when there exists a strict prefix $o'$ of $o$ such that $o' \notin I(O_\varphi)$. This is the case whatever the weakening mechanism at hand, especially when it is *dwm*. For instance, consider the inconsistent formula $\varphi = \neg(\vee(\neg\top, \top)$: $\varphi$ is inconsistent, $I(O_\varphi) = \{\epsilon, 11\}$, but the formula $\neg(\vee(\top, \top)$ obtained by replacing $\varphi_{11}$ by $\top$ in $\varphi$ still is inconsistent.

However, we can demonstrate that applying partial consolidation cannot produce such a spurious situation since a replacement can occur only at an occurrence belonging to a prefix-closed subset of the set of occurrences where inconsistent subformulae of $\varphi$ are rooted. Hence:

**Proposition 2.** *$\forall o \in max(PC(I(O_\varphi)), \sqsubseteq)$, $o$ has no strict prefix $o'$ such that $l(\varphi, o') = \neg$.*

*Proof.* By definition $R(\varphi)$ is a subset of $PC(I(O_\varphi))$. Let us show that $\nexists o \in PC(I(O_\varphi))$ such that $o' \sqsubset o$ and $l(\varphi, o') = \neg$. Towards a contradiction: let us suppose that such a word $o'$ exists. From the definition of a prefix-closed set, $PC(I(O_\varphi))$ is closed iff each strict prefix of $o$ is included in $PC(I(O_\varphi))$. Thus, $o'.1$ belongs to $PC(I(O_\varphi))$ and then to $I(O_\varphi)$. Consequently, $\varphi_{o'.1}$ is inconsistent, so that $\neg\varphi_{o'.1}$ is valid. This contradicts our assumption that $o' \in PC(I(O_\varphi))$. □

## 4 A CONSOLIDATION ALGORITHM

In this section, we present a two-step algorithm to consolidate an inconsistent modal formula $\varphi$. The first step consists in computing the set of occurrences $R(\varphi)$ that need to be repaired. A naive way to generate this set would consist in first computing $I(O_\varphi)$, then filtering out from it the set $PC(I(O_\varphi))$, and the set of occurrences of $PC(I(O_\varphi))$ which are maximal w.r.t. the prefix-order. However, in practice, checking the consistency of a modal formula is often computationally expensive, hence it makes sense to avoid testing the consistency of every subformula of $\varphi$. Especially, from the previous sections, one knows that testing the consistency of any subformula rooted at an occurrence not belonging to a prefix-closed subset of $I(O_\varphi)$ is useless.

These observations allow us to design a depth-first search procedure (Algorithm 1) to recursively computing the set of occurrences that need to be repaired (i.e., those corresponding to the subformulae to be weakened). The following proposition states that RS returns precisely the set of all the inconsistent prefix-closed occurrences of its input formula.

**Proposition 3.** *Let $\varphi \in \mathcal{L}$.* RS*($\varphi$) = $R(\varphi)$.*

*Proof.* Let us prove that RS($\varphi$) computes $R(\varphi)$ by induction on $\varphi$.

**Base case:** Suppose that $\varphi$ is consistent. In this case, $I(O_\varphi) = \emptyset$ and then $R(\varphi)$ is empty. In the case when $\varphi$ is consistent, RS($\varphi$) also returns an empty set (line 1) so that RS($\varphi$) = $R(\varphi)$.

**Induction step:** Now let us reason on the structure of an inconsistent $\varphi$ and let us consider the different cases:

1. If $\varphi = \neg(\varphi_1)$ is inconsistent, then $R(\varphi) = \{\epsilon\}$. Indeed, $\neg(\varphi_1)$ is inconsistent iff $\varphi_1$ is valid. In such a case, the occurrence 1 does not belong to $I(O_\varphi)$ and then $PC(I(O_\varphi))$ cannot contain an occurrence that starts by 1 (otherwise, it would not be closed). Since all occurrences of $\varphi$ start by 1, the only remaining occurrence is $\{\epsilon\}$. As we can see RS($\varphi$) also returns $\{\epsilon\}$ (line 2) and then RS($\varphi$) = $R(\varphi)$;
2. If $\varphi = \Box(\varphi_1)$ or $\varphi = \Diamond(\varphi_1)$ are inconsistent, then two cases have to be considered:
   (a) The first case is quite similar to the previous one: if $\varphi_1$ is consistent then 1 does not belong to $I(O_\varphi)$ so $R(\varphi) = \{\epsilon\}$ (by construction all occurrences of $\varphi$ start by 1). Now, if we observe the algorithm we can see that RS($\varphi$) also equals $\{\epsilon\}$ (line 4);
   (b) Now, if $\varphi_1$ is inconsistent, then we have to show that $R(\varphi)$ can be generated as the set of occurrences $1.o$ where $o$ belongs to $R(\varphi_1)$. First, by construction of $I$, we have that if $o' \in I(O_{\varphi_1})$, then $1.o' \in I(O_\varphi)$. Conversely, $\forall o \in I(O_\varphi) \setminus \{\epsilon\}$, we have that $o = 1.o'$ and $o' \in I(O_{\varphi_1})$. Indeed, any inconsistent sub-formula of $\varphi_1$ rooted at $o'$ also is a sub-formula of $\varphi$ rooted at $1.o'$. Since

$1 \in I(O_\varphi)$, if $o' \in PC(I(O_{\varphi_1}))$, then $1.o' \in PC(I(O_\varphi))$. Towards a contradiction: if $1.o' \notin PC(I(O_\varphi))$, then $\exists 1.o'' \sqsubseteq 1.o'$ such that $1.o'' \notin I(O_\varphi)$. Then, we have shown that $o'' \in I(O_{\varphi_1})$ if and only if $1.o'' \in I(O_\varphi)$, thus $o'' \notin I(O_\varphi)$. Because $o'' \sqsubseteq o'$, by definition of a prefix-closed set, we get that $o' \notin PC(I(O_{\varphi_1})$, that contradicts our assumption. Similarly, we can show that $\forall o' \in R(\varphi_1)$ we have $1.o' \in R(\varphi)$. Indeed, if two occurrences $o_1$ and $o_2$ are such that $o_1 \sqsubseteq o_2$, then $1.o_1 \sqsubseteq 1.o_2$. Consequently, if $o'$ is a maximal element in $PC(I(O_{\varphi_1}))$ w.r.t. $\sqsubseteq$, then $1.o'$ is a maximal element of $PC(I(O_\varphi))$ w.r.t. $\sqsubseteq$. If we consider again the algorithm we can see that $\mathrm{RS}(\varphi) = 1 \times \mathrm{RS}(\varphi_1)$ (line 5), then by induction hypothesis $\mathrm{RS}(\varphi_1) = R(\varphi_1)$ and then $\mathrm{RS}(\varphi) = 1 \times R(\varphi_1) = R(\varphi)$ which is the expected result;

3. If $\varphi = \vee(\varphi_1, \varphi_2)$ is inconsistent, then at least one of the two sub-formulae $\varphi_1$ and $\varphi_2$ must be repaired. To avoid making an arbitrary choice, we have decided to repair both of them.[5] Using a similar reasoning process as the one considered for modalities, we can show that $\forall o' \in R(\varphi_1)$ we have $1.o' \in R(\varphi)$ and $\forall o' \in R(\varphi_2)$ we have $2.o' \in R(\varphi)$. First, $\forall o' \in I(O_{\varphi_1})$ we have $1.o' \in I(O_\varphi)$ and $\forall o' \in I(O_{\varphi_2})$ we have $2.o' \in I(O_\varphi)$. Then, since $\{1, 2\} \subseteq I(O_\varphi)$, $\forall o' \in PC(I(O_{\varphi_1}))$ we have $1.o' \in PC(I(O_\varphi))$ and $\forall o' \in PC(I(O_{\varphi_2}))$ we have $2.o' \in PC(I(O_\varphi))$. Because $((1 \times PC(I(\varphi_1)))) \cap ((2 \times PC(I(\varphi_2)))) = \emptyset$, the maximal occurrences w.r.t. $\sqsubseteq$ are the union of those taken in each set independently. Thus,
$$\begin{aligned} R(\varphi) &= max((1 \times PC(I(O_{\varphi_1}))) \cup (2 \times PC(I(O_{\varphi_2}))), \sqsubseteq) \\ &= max((1 \times PC(I(O_{\varphi_1}))), \sqsubseteq) \cup max((2 \times PC(I(O_{\varphi_2}))), \sqsubseteq) \\ &= (1 \times R(\varphi_1)) \cup (2 \times R(\varphi_2)); \end{aligned}$$
If we consider the algorithm we have $\mathrm{RS}(\varphi) = (1 \times \mathrm{RS}(\varphi_1)) \cup (2 \times \mathrm{RS}(\varphi_2))$ (line 7). By induction hypothesis $\mathrm{RS}(\varphi_i) = R(\varphi_i)$ and then $\mathrm{RS}(\varphi) = R(\varphi)$;
4. If $\varphi = \wedge(\varphi_1, \varphi_2)$ is inconsistent, then three cases have to be considered:
   (a) The first case is when both subformulae $\varphi_1$ and $\varphi_2$ are consistent. In this case, we have $(\{1, 2\} \cap I(O_\varphi)) = \emptyset$ so that $PC(I(O_\varphi)) = \{\epsilon\}$. Thus $PC(I(O_\varphi)) = \{\epsilon\}$ necessary implies that $R(\varphi) = \{\epsilon\}$. Now, let us consider the algorithm. If both subformulae are satisfiable, then both conditions line 10 and line 11 are not satisfied. Consequently, since $w$ is initially set to $\emptyset$ and it does not change, then $w = \emptyset$ and the condition line 12 is satisfied. Thus $\mathrm{RS}(\varphi) = \{\epsilon\} = R(\varphi)$;
   (b) The second case is when only one of the two subformulae $\varphi_1$ and $\varphi_2$ is inconsistent. Without loss of generality, let us suppose that $\varphi_1$ is inconsistent. If $\varphi_1$ is inconsistent, then $\forall o' \in I(O_{\varphi_1})$ we have $1.o' \in I(O_\varphi)$. Similarly to what we did for the modalities, since $1 \in I_{O_\varphi}$, it is the case that $\forall o' \in PC(I(O_{\varphi_1}))$ we have $1.o' \in PC(I(O_\varphi))$ (see case 3). We can conclude that $o = 1.o' \in R(\varphi)$ if and only if $o' \in R(\varphi_1)$. If we consider the value returned by the algorithm RS we can see that, since the condition line 10 ($\varphi_2$ is inconsistent) is satisfied and the condition line 11 ($\varphi_2$ is consistent) and line 12 ($w$ cannot be empty since it is set to $1 \times \mathrm{RS}(\varphi_1)$ line 10) are not satisfied, then $\mathrm{RS} = 1 \times \mathrm{RS}(\varphi_1)$. By induction hypothesis, we have $\mathrm{RS}(\varphi_1) = R(\varphi_1)$. Thus, $\mathrm{RS}(\varphi) = 1 \times \mathrm{RS}(\varphi_1) = 1 \times R(\varphi_1) = R(\varphi)$. A similar reasoning can be done for the case when $\varphi_2$ is the sole subformula that is inconsistent.
   (c) Finally, let us consider the case when both subformulae $\varphi_1$ and $\varphi_2$ are inconsistent. As for the previous case, because both subformulae correspond to occurrences that start with different initials, we can conclude that:
$$\begin{aligned} R(\varphi) &= max((1 \times PC(I(O_{\varphi_1}))), \sqsubseteq) \cup max((2 \times PC(I(O_{\varphi_2}))) \\ &= (1 \times R(\varphi_1)) \cup (2 \times R(\varphi_2)) \end{aligned}$$
   Again, if we consider the algorithm, since both subformulae are consistent the conditions line 10 and line 11 are satisfied $\varphi$ and condition line 12 is not satisfied ($w$ is set to something different to $\emptyset$ line 10 and line 11), then $\mathrm{RS} = (1 \times \mathrm{RS}(\varphi_1)) \cup (2 \times \mathrm{RS}(\varphi_2))$. By induction hypothesis, we have $\mathrm{RS}(\varphi_1) = R(\varphi_1)$ and $\mathrm{RS}(\varphi_2) = R(\varphi_2)$. Consequently, $\mathrm{RS}(\varphi) = (1 \times \mathrm{RS}(\varphi_1)) \cup (2 \times \mathrm{RS}(\varphi_2)) = (1 \times R(\varphi_1)) \cup (2 \times R(\varphi_2)) = R(\varphi)$.

□

**Algorithm 1:** Inconsistent prefix-closed set extraction

```
RS($\varphi$ a formula from $\mathcal{L}$): a set of words
1  if $\varphi$ is satisfiable then return $\emptyset$;
2  if $\varphi = \neg(\varphi_1)$ then return $\{\epsilon\}$;
3  if $\varphi = \Box(\varphi_1)$ or $\varphi = \Diamond(\varphi_1)$ then
4      if $\varphi_1$ is consistent then return $\{\epsilon\}$;
5      return $1 \times$ RS ($\varphi_1$)
6  if $\varphi = \vee(\varphi_1, \varphi_2)$ then
7      return $(1 \times$ RS $(\varphi_1)) \cup (2 \times$ RS $(\varphi_2))$
8  if $\varphi = \wedge(\varphi_1, \varphi_2)$ then
9      $\omega \leftarrow \emptyset$;
10     if $\varphi_1 \equiv \bot$ then $\omega \leftarrow 1 \times$ RS $(\varphi_1)$;
11     if $\varphi_2 \equiv \bot$ then $\omega \leftarrow \omega \cup (2 \times$ RS $(\varphi_2))$;
12     if $\omega = \emptyset$ then return $\{\epsilon\}$;
13     return $\omega$;
```

**Algorithm 2:** Consolidation

```
Repair($\varphi$ a formula from $\mathcal{L}$, wm a weakening mechanism): a
formula from $\mathcal{L}$
1  $\varphi' \leftarrow \varphi$;
2  while $\varphi'$ is inconsistent do
3      $\omega \leftarrow$ RS $(\varphi')$;
4      foreach $o \in \omega$ do
5          $\varphi' \leftarrow \varphi'$ where $\varphi'_o$ has been replaced by $wm(\varphi'_o)$;
6  return $\varphi'$;
```

Once the set of occurrences that need to be repaired has been computed, it is easy to design an iterative method that computes a consolidated formula. This method, given by Algorithm 2, starts by copying the formula $\varphi$ in $\varphi'$ (line 1). Then, while $\varphi$ is inconsistent, a repairing set $\omega$ is computed using the Repair-Set function (line 3) and each subformula $\varphi'_o$, with $o \in \omega$, is weakened using $wm$ (lines 4–5). Finally, the consolidated formula $\varphi'$ is returned (line 6) after a finite number of calls to RS. The following proposition shows that the algorithm terminates and returns the consolidation $C(\varphi)$.

**Proposition 4.** Repair($\varphi$, $wm$) *terminates and returns the consolidation $C(\varphi)$ of $\varphi$ for wm.*

*Sketch.* Let us show that Repair($\varphi$, $wm$) terminates after a finite number of steps and returns a consolidation of $\varphi$. The algorithm termi-

[5] Other repair strategies can be considered, for instance, by first repairing $\varphi_1$, and then $\varphi_2$ only if needed, or vice-versa.

nates once the condition at line 2 is false, that is $\varphi'$ is no more inconsistent. Because Prop.3, the operations done at lines 4–5 exactly consist in computing $C_p(\varphi')$ for the weakening mechanism *wm*. Then, at each step $C_p(\varphi')$ is stored in $\varphi'$ and the process is repeated. Consequently, if we consider $C^i_p(\varphi')$ the formula obtained at the $i^{th}$ iteration of the while loop, then the next iteration will compute $C_p(C^i_p(\varphi'))$. Because, $\varphi'$ is set to $\varphi$ ($C^0_p(\varphi')$ in Def.4), after *min* steps we exactly compute the partial consolidation of $\varphi$ for *wm* ($C^{min}_p(\varphi')$ in Def.4). Prop.1 ensures that the resulting formula will be consistent and then the algorithm will terminate after a finite number of steps and it will return the consolidation of $\varphi$ for the weakening mechanism *wm*. □

Let us now present some properties offered by the consolidated base computed using Repair.

**Proposition 5.** *Let* $\psi = \textsc{Repair}(\varphi, wm)$*.*

1. *If $\varphi$ is consistent, then $\psi = \varphi$*
2. $\textsc{Repair}(\alpha \wedge \beta, wm) \equiv \textsc{Repair}(\beta \wedge \alpha, wm)$
3. $\textsc{Repair}(\alpha \vee \beta, wm) \equiv \textsc{Repair}(\beta \vee \alpha, wm)$

*Proof.* Let us consider each point separately:

1. Since $\varphi' = \varphi$ (line 1), if $\varphi$ is consistent then Repair does not consider the while loop (lines 2–5). In such a case, the function returns $\varphi' = \varphi = \psi$ (line 6);
2. Given $S$ an non-empty prefix-closed subset of $I(O_\varphi)$, let $C_S(\varphi)$ be the formula obtained by replacing in $\varphi$ every subformula occurring at some $o \in S$ by $\top$. To prove the statement, it is enough to show that $C_{\mathrm{RS}(\varphi_1 \wedge \varphi_2)}(\varphi_1 \wedge \varphi_2) \equiv C_{\mathrm{RS}(\varphi_2 \wedge \varphi_1)}(\varphi_2 \wedge \varphi_1)$. Five cases have to be considered:

   (a) $\varphi_1 \wedge \varphi_2$ is consistent: in this case, $\mathrm{RS}(\varphi_1 \wedge \varphi_2) = \emptyset = \mathrm{RS}(\varphi_2 \wedge \varphi_1)$. Consequently, $C_{\mathrm{RS}(\varphi_1 \wedge \varphi_2)}(\varphi_1 \wedge \varphi_2) \equiv C_{\mathrm{RS}(\varphi_2 \wedge \varphi_1)}(\varphi_2 \wedge \varphi_1)$ holds;

   (b) $\varphi_1$ is consistent and $\varphi_2$ is consistent. In such a case, $\mathrm{RS}(\varphi_1 \wedge \varphi_2) = \{\epsilon\} = \mathrm{RS}(\varphi_2 \wedge \varphi_1)$. Consequently, $C_{\mathrm{RS}(\varphi_1 \wedge \varphi_2)}(\varphi_1 \wedge \varphi_2) \equiv C_{\mathrm{RS}(\varphi_2 \wedge \varphi_1)}(\varphi_2 \wedge \varphi_1)$ holds;

   (c) $\varphi_1$ is consistent and $\varphi_2$ is not consistent. In such a case, $\mathrm{RS}(\varphi_1 \wedge \varphi_2) = 2 \times \mathrm{RS}(\varphi_2)$ and $\mathrm{RS}(\varphi_2 \wedge \varphi_1) = 1 \times \mathrm{RS}(\varphi_2)$. Thus, $C_{\mathrm{RS}(\varphi_1 \wedge \varphi_2)}(\varphi_1 \wedge \varphi_2) = \varphi_1 \wedge C_{\mathrm{RS}(\varphi_2)}(\varphi_2)$ since $\mathrm{RS}(\varphi_1 \wedge \varphi_2)$ does not contain occurrences of sub-formulae of $\varphi_1$. Similarly, we have $C_{\mathrm{RS}(\varphi_2 \wedge \varphi_1)}(\varphi_2 \wedge \varphi_1) = C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \wedge \varphi_1$. Consequently, since $\wedge$ is commutative then $C_{\mathrm{RS}(\varphi_1 \wedge \varphi_2)}(\varphi_1 \wedge \varphi_2) \equiv \varphi_1 \wedge C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \equiv C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \wedge \varphi_1 \equiv C_{\mathrm{RS}(\varphi_2 \wedge \varphi_1)}(\varphi_2 \wedge \varphi_1)$;

   (d) $\varphi_1$ is not consistent and $\varphi_2$ is consistent: similar to the previous case;

   (e) $\varphi_1$ and $\varphi_2$ are inconsistent: in this case, $\mathrm{RS}(\varphi_1 \wedge \varphi_2) = (1 \times \mathrm{RS}(\varphi_1)) \cup (2 \times \mathrm{RS}(\varphi_2))$ and $\mathrm{RS}(\varphi_2 \wedge \varphi_1) = (1 \times \mathrm{RS}(\varphi_2)) \cup (2 \times \mathrm{RS}(\varphi_1))$. Consequently, $C_{\mathrm{RS}(\varphi_1 \wedge \varphi_2)}(\varphi_1 \wedge \varphi_2) \equiv C_{\mathrm{RS}(\varphi_1)}(\varphi_1) \wedge C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \equiv C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \wedge C_{\mathrm{RS}(\varphi_1)}(\varphi_1) \equiv C_{\mathrm{RS}(\varphi_2 \wedge \varphi_1)}(\varphi_2 \wedge \varphi_1)$.

   The previous result shows that at each iteration in the while loop, whatever $\varphi_1 \wedge \varphi_2$ or $\varphi_2 \wedge \varphi_1$ is considered as input, an equivalent $\varphi'$ is obtained as an output. Consequently, we can conclude that $\textsc{Repair}(\alpha \wedge \beta, dwm) \equiv \textsc{Repair}(\beta \wedge \alpha, dwm)$ holds;
3. Similarly to the previous point, to prove this statement it is enough to show that $C_{\mathrm{RS}(\varphi_1 \vee \varphi_2)}(\varphi_1 \vee \varphi_2) \equiv C_{\mathrm{RS}(\varphi_2 \vee \varphi_1)}(\varphi_2 \vee \varphi_1)$. Here only two cases have to be considered:

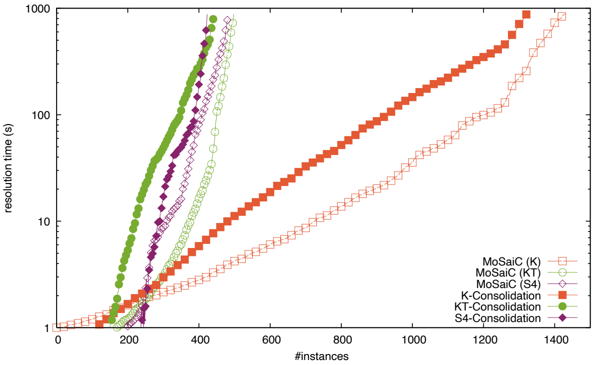
   (a) if $\varphi_1 \vee \varphi_2$ is consistent then $\mathrm{RS}(\varphi_1 \vee \varphi_2) = \emptyset = \mathrm{RS}(\varphi_2 \vee \varphi_1)$. Consequently, $C_{\mathrm{RS}(\varphi_1 \vee \varphi_2)}(\varphi_1 \vee \varphi_2) \equiv C_{\mathrm{RS}(\varphi_2 \vee \varphi_1)}(\varphi_2 \vee \varphi_1)$ holds;

   (b) if $\varphi_1 \vee \varphi_2$ is not consistent then $\mathrm{RS}(\varphi_1 \vee \varphi_2) = (1 \times RS(\varphi_1)) \cup (2 \times RS(\varphi_2))$ and $\mathrm{RS}(\varphi_2 \vee \varphi_1) = (1 \times RS(\varphi_2)) \cup (2 \times RS(\varphi_1))$. Consequently, since $\vee$ is commutative then $C_{\mathrm{RS}(\varphi_1 \vee \varphi_2)}(\varphi_1 \vee \varphi_2) \equiv C_{\mathrm{RS}(\varphi_1)}(\varphi_1) \vee C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \equiv C_{\mathrm{RS}(\varphi_2)}(\varphi_2) \vee C_{\mathrm{RS}(\varphi_1)}(\varphi_1) \equiv C_{\mathrm{RS}(\varphi_2 \vee \varphi_1)}(\varphi_2 \vee \varphi_1)$.

   Consequently, we can conclude that $\textsc{Repair}(\alpha \vee \beta, dwm) \equiv \textsc{Repair}(\beta \vee \alpha, dwm)$ holds.

□

**Figure 3**: Runtime of Comté and MoSaiC



Let us now present some additional properties offered by the consolidated base computed using Repair when the drastic weakening mechanism *dwm* is used.

**Proposition 6.** *Let* $\psi = \textsc{Repair}(\varphi, dwm)$*.*

1. $O_\psi \subseteq O_\varphi$
2. *If $o \in O_\varphi \setminus O_\psi$, then $o \in I(O_\varphi)$, $o$ has not strict prefix $o'$ such that $l(\varphi, o') = \neg$ and $o$ has not strict prefix $o''$ such that $l(\psi, o'') = \top$*

*Proof.*
1. As stated by Prop.3, $\mathrm{RS}(\varphi) = R(\varphi) = max(PC(I(O_\varphi)), \sqsubseteq)$ which is an inconsistent prefix-closed set $\varphi$. Since the function Repair only replaces subformulae from $\varphi'$ occurring at some $o \in R(\varphi')$, then we can only remove occurrences from $\varphi'$. Consequently, since $\varphi'$ is set to $\varphi$ (line 1) and $\psi = \varphi'$ at the end of the procedure, $O_\psi \subseteq O_\varphi$ holds;
2. Because RS compute a partial consolidation (see Prop.3) and Prop.2, it is clear that if $o \in O_\varphi \setminus O_\psi$, then $o \in I(O_\varphi)$, $o$ has not strict prefix $o'$ such that $l(\varphi, o') = \neg$ and $o$ has not strict prefix $o''$ such that $l(\psi, o'') = \top$;

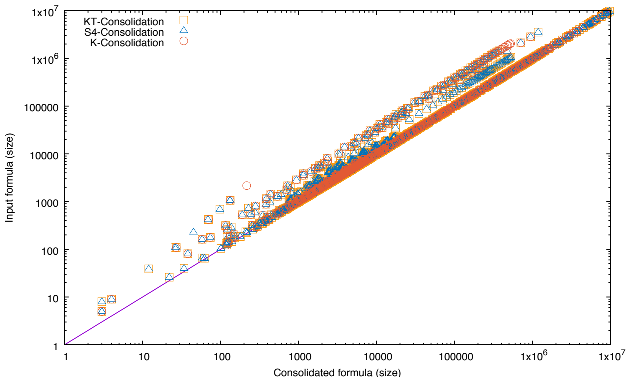
□

It must be kept in mind here that no full syntax independence can be obtained whatever the approach when one wants to avoid the trivialization of inference; otherwise, any inconsistent $\varphi$ could be replaced by the equivalent formula $\bot$ from which, obviously, no relevant conclusion can be drawn).

## 5 EXPERIMENTAL EVALUATION

In this section, we report and comment some empirical results about the performance of our consolidation method. The datasets we considered in our experiments are the benchmarks reported in [24, 25]. They consist of modal logic formulae, that are inconsistent when interpreted in K, KT, or S4. These benchmarks are known in the community under the names: Logic WorkBench (LWB) [4], MQBF [27], and 3CNF [29]. They have already been split into consistent/inconsistent formulae. Furthermore, it is known that inconsistency in K implies inconsistency in KT and S4 and that inconsistency in KT implies inconsistency in S4. In total, 2454 benchmarks consisting of

**Figure 4**: Size of the consolidation vs. size of the input formula



inconsistent formulae in modal logic have been considered in our empirical study. The hardness of a consistency test does not only lie in the number of variables involved, but the way modalities are nested has a huge impact. For example, the MQBF family consists of formulae with only one variable, while the minimum (resp. maximum, average±std) modal depth in the family is 19 (resp. 225, $69.2 \pm 47.5$).

We have developed a C++ tool, called Comté (COnsolidation Modal Tool), that implements our consolidation method. [6] Comté can deal with any normal modal logic as long as a modal logic $K\star$ oracle for deciding the consistency of any formula is provided. We have chosen to take advantage of MoSaiC [25, 23] as a modal logic K⋆ oracle. The experiments have been carried out on a Xeon 4-core, 3.3 GHz, running CentOS 7.0, with a memory limit set to 32GiB. The runtime limit was set to 900 seconds per benchmark.

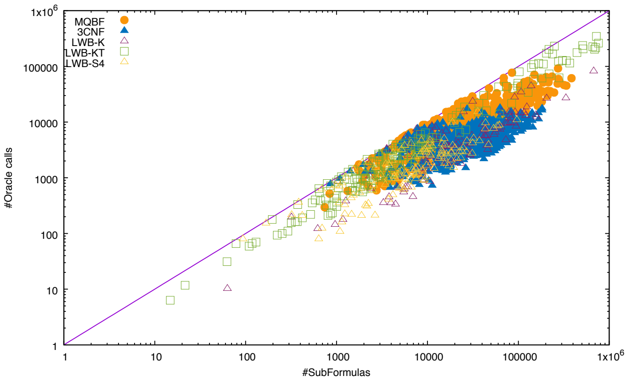
Figures 3 is about the computation time required to test the K-consistency (resp. KT-consistency, S4-consistency) of a formula using MoSaiC and to consolidate it using Comté. For each task, the number of instances (on the x-axis) that have been "solved" given an amount of time (reported on the y-axis) is reported. In general, the time needed to consolidate a formula is significantly larger than the time needed to decide the consistency of the instance.

Each dot in Figure 4 represents an instance where the x-coordinate indicates the size of the initial formula and the y-coordinate gives the size of the resulting consolidated formula. Logarithmic scales are used here for both coordinates. A dot on the diagonal means that the consolidation process did not change the formula. It can be observed that all dots are extremely close to the diagonal. This shows that in the input instances, only few inconsistencies occur and they are "far from the root" (they correspond to subformulae which occur deeply in the input). Once these subformulae are spotted and repaired, the formula becomes consistent.

The ratio between the the size of the input formula and the consolidated formulae can be viewed as an inconsistency measure, i.e., a quantity which is meant to tell how inconsistent the formula is, similarly to what has been done in [7] in the case of propositional logic (see [1, 14, 35] for more information about inconsistency measures).

The minimal ratio $\frac{(consolidated-size)}{(input-size)}$ is equal to 0.4491%. This means that after consolidation, recovering consistency may require the loss of much information from the input formula. However, the instances for which this minimal ratio is obtained are rare, namely the first quartile, the median value and the third quartile are respectively equal to 98.74%, 98.21% and 99.99%. This confirms the intuition that the inconsistency of the formulae considered in the experiments is typically caused by few and very small inconsistent subformulae. For these cases, our consolidation approach permits to preserve a lot of information from the input. Obviously, there is no formula leading to a ratio equal to 100%, because only inconsistent formulae have been considered in the experiments.

**Figure 5**: Number of oracle calls



| | #Oracle | #Repairs | Runtime (s) |
|---|---|---|---|
| 3CNF | 58.32 | 42.39 | 52.61 |
| MQBF | 71.41 | 45.09 | 53.25 |
| LWB-K | 41.26 | 18.88 | 84.07 |
| LWB-KT | 38.74 | 21.03 | 81.54 |
| LWB-S4 | 39.95 | 25.20 | 41.09 |

**Table 1**: Number of calls to the consistency oracle, number of repairs and runtime for computing the consolidation (average values).

Figure 5 reports a scatter plot that gives the number of oracle calls performed by Comté (y-axis) against the number of calls that would be performed by a naive procedure that would have considered all the subformulae (x-axis). Logarithmic scales are used for both coordinates. It can be observed that all dots are way below the diagonal, which means that the computation of the consolidated formula requires with very few calls to MoSaiC. Table 1 gives additional statistics regarding the experiments.

## 6 CONCLUSION

We have presented a new approach for consolidating modal logic formulae. This approach consists in weakening some of the inconsistent subformulae of the input formula in an iterative fashion. It ensures that the the consolidated base is classically consistent. It also guarantees this base to coincide with the input when the latter is consistent. Large-scale experiments have been conducted, showing that the approach is practical for instances of reasonable sizes.

In our approach, many weakening mechanisms can be used. Though we focused on the drastic one in the experiments (for the sake of simplicity), other choices, preserving more information, could have been considered instead. For instance, a weakening mechanism suited to KT and to S4 consists in replacing in an inconsistent formula $\varphi$ the connective occurring at $\epsilon$ by $\vee$ when $l(\epsilon, \varphi) = \wedge$, the modality occurring at $\epsilon$ by $\Diamond$ when $l(\epsilon, \varphi) = \Box$, and the formula $\varphi$ by $\top$ otherwise. Clearly enough, this weakening mechanism is less drastic than *dwm*. A perspective for further research consists in experimenting our approach to consolidation when equipped with such weakening mechanisms in order to determine the extent to which they lead to preserve more information in practice than our approach equipped with *dwm*.

[6] Comté is available on http://www.cril.fr/~montmirail/modal-consolidation/.

## REFERENCES

[1] Meriem Ammoura, Badran Raddaoui, Yakoub Salhi, and Brahim Oukacha, 'On measuring inconsistency using maximal consistent sets', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 13th European Conference, ECSQARU 2015, Compiègne, France, July 15-17, 2015. Proceedings*, eds., Sébastien Destercke and Thierry Denoeux, volume 9161 of *LNCS*, pp. 267–276. Springer, (2015).

[2] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki, 'Consistent query answers in inconsistent databases', in *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania, USA*, eds., Victor Vianu and Christos H. Papadimitriou, pp. 68–79. ACM Press, (1999).

[3] Jean-Francois Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia, 'Inconsistency-tolerant query answering: Rationality properties and computational complexity analysis', in *Logics in Artificial Intelligence - 15th European Conference, JELIA 2016, Larnaca, Cyprus, November 9-11, 2016, Proceedings*, eds., Loizos Michael and Antonis C. Kakas, volume 10021 of *LNCS*, pp. 64–80, (2016).

[4] Peter Balsiger, Alain Heuerding, and Stefan Schwendimann, 'A Benchmark Method for the Propositional Modal Logics K, KT, S4', *J. of Automated Reasoning*, **24**(3), 297–317, (2000).

[5] Salem Benferhat, Didier Dubois, Jérôme Lang, Henri Prade, Alessandro Saffiotti, and Philippe Smets, 'A general approach for inconsistency handling and merging information in prioritized knowledge bases', in *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998.*, eds., Anthony G. Cohn, Lenhart K. Schubert, and Stuart C. Shapiro, pp. 466–477. Morgan Kaufmann, (1998).

[6] Leopoldo E. Bertossi, *Database Repairing and Consistent Query Answering*, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2011.

[7] Philippe Besnard, 'Forgetting-based inconsistency measure', in *Scalable Uncertainty Management - 10th International Conference, SUM 2016, Nice, France, September 21-23, 2016, Proceedings*, eds., Steven Schockaert and Pierre Senellart, volume 9858 of *LNCS*, pp. 331–337. Springer, (2016).

[8] Philippe Besnard and Anthony Hunter, *Introduction to Actual and Potential Contradictions*, 1–11, Springer Netherlands, Dordrecht, 1998.

[9] Philippe Besnard and Torsten Schaub, 'Signed systems for paraconsistent reasoning', *J. Autom. Reasoning*, **20**(1), 191–213, (1998).

[10] Meghyn Bienvenu, 'Prime implicates and prime implicants: From propositional to modal logic', *J. Artif. Intell. Res.*, **36**, 71–128, (2009).

[11] Meghyn Bienvenu and Riccardo Rosati, 'Tractable approximations of consistent query answering for robust ontology-based data access', in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, ed., Francesca Rossi, pp. 775–781. IJCAI/AAAI, (2013).

[12] Patrick Blackburn, Johan van Benthem, and Frank Wolter, *Handbook of Modal Logic*, volume 3, Elsevier, 2006.

[13] Anthony Finkelstein, Dov M. Gabbay, Anthony Hunter, Jeff Kramer, and Bashar Nuseibeh, 'Inconsistency handling in multperspective specifications', *IEEE Trans. Software Eng.*, **20**(8), 569–578, (1994).

[14] John Grant and Anthony Hunter, 'Distance-based measures of inconsistency', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 12th European Conference, ECSQARU 2013, Utrecht, The Netherlands, July 8-10, 2013. Proceedings*, ed., Linda C. van der Gaag, volume 7958 of *LNCS*, pp. 230–241. Springer, (2013).

[15] Joseph Y. Halpern and Yoram Moses, 'A guide to completeness and complexity for modal logics of knowledge and belief', *Artif. Intell.*, **54**(2), 319–379, (1992).

[16] Sven Ove Hansson, *A Textbook of Belief Dynamics: Solutions to Exercises*, Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[17] Matthew Horridge, Bijan Parsia, and Ulrike Sattler, 'Laconic and precise justifications in OWL', in *Proc. of ISWC*, eds., Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, volume 5318 of *LNCS*, pp. 323–338. Springer, (2008).

[18] Anthony Hunter, 'Handbook of defeasible reasoning and uncertainty management systems', chapter Paraconsistent Logics, 11–36, Kluwer Academic Publishers, Norwell, MA, USA, (1998).

[19] Anthony Hunter, Simon Parsons, and Michael J. Wooldridge, 'Measuring inconsistency in multi-agent systems', *KI*, **28**(3), 169–178, (2014).

[20] Sébastien Konieczny, Jérôme Lang, and Pierre Marquis, 'Reasoning under inconsistency: the forgotten connective', in *Proc. of IJCAI*, pp. 484–489, (2005).

[21] Sarit Kraus, Daniel J. Lehmann, and Menachem Magidor, 'Nonmonotonic reasoning, preferential models and cumulative logics', *Artificial Intelligence*, **44**(1-2), 167–207, (1990).

[22] Saul Kripke, 'A Completeness Theorem in Modal Logic', *J. of Symbolic Logic*, **24**(1), 1–14, (1959).

[23] Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail, 'A recursive shortcut for CEGAR: application to the modal logic K satisfiability problem', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, ed., Carles Sierra, pp. 674–680. ijcai.org, (2017).

[24] Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail, 'An assumption-based approach for solving the minimal s5-satisfiability problem', in *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Oxford, UK, July 14-17, 2018, Proceedings*, eds., Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, volume 10900 of *LNCS*, pp. 1–18. Springer, (2018).

[25] Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail, 'A sat-based approach for PSPACE modal logics', in *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.*, eds., Michael Thielscher, Francesca Toni, and Frank Wolter, pp. 651–652. AAAI Press, (2018).

[26] Xiang Lian, Lei Chen, and Shaoxu Song, 'Consistent query answers in inconsistent probabilistic databases', in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, eds., Ahmed K. Elmagarmid and Divyakant Agrawal, pp. 303–314. ACM, (2010).

[27] Fabio Massacci and Francesco M. Donini, 'Design and Results of TANCS-2000 Non-classical (Modal) Systems Comparison', in *Proc. of TABLEAUX'00*, pp. 52–56, (2000).

[28] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon, 'A modal-layered resolution calculus for K', in *Automated Reasoning with Analytic Tableaux and Related Methods - 24th International Conference, TABLEAUX 2015, Wrocław, Poland, September 21-24, 2015. Proceedings*, ed., Hans de Nivelle, volume 9323 of *LNCS*, pp. 185–200. Springer, (2015).

[29] Peter F. Patel-Schneider and Roberto Sebastiani, 'A New General Method to Generate Random Modal Formulae for Testing Decision Procedures', *J. of Artificial Intelligence Research*, **18**, 351–389, (2003).

[30] Gadi Pinkas and Ronald Prescott Loui, 'Reasoning from inconsistency: A taxonomy of principles for resolving conflict', in *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, USA, October 25-29, 1992.*, eds., Bernhard Nebel, Charles Rich, and William R. Swartout, pp. 709–719. Morgan Kaufmann, (1992).

[31] Raymond Reiter, 'A logic for default reasoning', *Artificial intelligence*, **13**(1), 81–132, (1980).

[32] N. Rescher and R. Manor, 'On inference from inconsistent premises', *Theory and Decision*, **1**, 179–219, (1970).

[33] Nicolas Schwind and Pierre Marquis, 'On consensus in belief merging', in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, eds., Sheila A. McIlraith and Kilian Q. Weinberger, pp. 1949–1956. AAAI Press, (2018).

[34] Roberto Sebastiani and Armando Tacchella, 'SAT techniques for modal and description logics', in *Handbook of Satisfiability*, eds., Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, volume 185 of *Frontiers in Artificial Intelligence and Applications*, 781–824, IOS Press, (2009).

[35] Matthias Thimm and Johannes Peter Wallner, 'Some complexity results on inconsistency measurement', in *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, eds., Chitta Baral, James P. Delgrande, and Frank Wolter, pp. 114–124. AAAI Press, (2016).

[36] Kewen Wang, Zhe Wang, Rodney W. Topor, Jeff Z. Pan, and Grigoris Antoniou, 'Eliminating concepts and roles from ontologies in expressive descriptive logics', *Computational Intelligence*, **30**(2), 205–232, (2014).