Symmetries in Modal Logics: A Coinductive Approach

Carlos Areces
FaMAF
Universidad Nacional de Córdoba
Córdoba, Argentina
CONICET, Argentina

Guillaume Hoffmann
FaMAF
Universidad Nacional de Córdoba
Córdoba, Argentina

Ezequiel Orbe
FaMAF
Universidad Nacional de Córdoba
Córdoba, Argentina
CONICET, Argentina

We generalize the notion of symmetries of propositional formulas in conjunctive normal form to modal formulas. Our framework uses the coinductive models introduced in [4] and, hence, the results apply to a wide class of modal logics including, for example, hybrid logics. Our main result shows that the symmetries of a modal formula preserve entailment: if σ is a symmetry of φ then $\varphi \models \psi$ if and only if $\varphi \models \sigma(\psi)$.

1 Symmetries in Automated Theorem Proving

Many concrete, real life problems present symmetries. For instance, if we want to know whether trying to place three pigeons in two pigeonholes results in two occupying the same nest, it does not really matter which of all pigeons gets in each pigeonhole. Starting by putting the first pigeon to the first pigeonhole is the same as if we put the second one in it. In mathematical and common-sense reasoning these kinds of symmetries are often used to reduce the difficulty of reasoning — one can analyze in detail only one of the symmetric cases and generalize the result to the others. The exact same is done in propositional theorem proving. Many problem classes and, in particular, those arising from real world applications, display a large number of symmetries; and current SAT solvers take into account these symmetries to avoid exploring duplicate branches of the search space. In the last years there has been extensive research in this area, focusing on how to define symmetries, how to detect them efficiently, and how SAT solvers can better profit from them [23].

Informally, we can define a symmetry of a discrete object as the permutation of its components that leaves the object, or some aspect of it, intact (think of the rotations of a spatial solid). In the context of SAT solving we can formally define a symmetry as a permutation of the variables (or literals) of a problem that preserves its structure and, in particular, its set of solutions. Depending on which aspect of the problem is kept invariant, symmetries are classified in the literature into semantic or syntactic [9]. Semantic symmetries are intrinsic properties of the Boolean function that are independent of any particular representation, i.e., a permutation of variables that does not change the value of the function under any variable assignment. Syntactic symmetries, on the other hand, correspond to the specific algebraic representation of the function, i.e., a permutation of variables or literals that does not change the representation. A syntactic symmetry is also a semantic symmetry, but the converse does not always hold.

In [20], Krishnamurthy used symmetries in the context of SAT solving. In this article, the notions of *global* and *local* symmetries as inference rules are used to strengthen resolution-based proof systems for propositional logic showing that they can shorten the proofs of certain difficult propositional problems, like the pigeonhole principle. Since then, many articles have investigated how to detect and exploit symmetries. Most of them can be grouped into two different approaches: static symmetry breaking and dynamic symmetry breaking. In the first approach [14, 15, 1], symmetries are detected and eliminated

To appear in EPTCS.

from the problem statement before the SAT solver is used. They work as a preprocessing step. In contrast, dynamic symmetry breaking [13, 9, 10] detects and breaks symmetries during the search space exploration. The first approach can be used with any theorem prover; but the second can take advantage of symmetries that emerge during search. Despite their differences they share the same goal: to identify symmetric branches of the search space and guide the SAT solver away from symmetric branches already explored. A third alternative was introduced in [8], which combines symmetry reasoning with clause learning [24] in Conflict-Driven Clause Learning SAT solvers [17]. The idea is to augment the clause learning process by using the symmetries of the problem to learn the symmetric equivalents of conflict-induced clauses. This approach is particularly appealing because it does not imply major modifications to the search procedure and the required modification to the clause learning process is minor.

Symmetries have been extensively investigated and successfully exploited for propositional logic SAT and some results involve other logics, see [6, 7, 16]. But, to the best of our knowledge, symmetries remain largely unexplored in automated theorem proving for modal logics.

In this paper, we generalize the notion of symmetries to modal formulas in conjunctive normal form for different modal logics including the basic modal language over different model classes (e.g., reflexive, linear or transitive models), and logics with additional modal operators (e.g., universal and hybrid operators). The main result of the article shows that symmetries of a modal formula preserve entailment: if σ is a symmetry of φ then $\varphi \models \psi$ if and only if $\varphi \models \sigma(\psi)$. In cases where the modal language has a tree model property, we can actually use a more flexible notion of symmetry that enables different permutations to be applied at each modal depth.

In order to tackle a broad range of modal languages that may or may not enjoy the tree model property, we use in our work the semantics provided by coinductive modal models [4] instead of the more familiar Kripke relational semantics. Coinductive modal models provide a homogeneous framework to investigate different modal languages at a greater level of abstraction. A consequence of this is that results obtained in the coinductive framework can be easily extended to concrete modal languages by just giving the appropiate definition of the model classes and fixing some parameters.

In Section 2 we present modal logics and coinductive modal models. In Section 3 we define modal symmetries, together with the appropriate notion of simulation we need to show that symmetries preserve modal entailment. In Section 4 we introduce layered permutations and show that they can be used when the modal logic has the adequate notion of the tree model property. We draw our conclusions and discuss future research in Section 5.

2 Modal Logics and Coinductive Models

In what follows, we will assume basic knowledge of classical modal logics and refer the reader to [11, 12] for technical details. The coinductive framework for modal logics was introduced in [4] to investigate normal forms for a wide number of modal logics. Its main characteristic is that it allows the representation of a wide range of modal logics in a homogeneous form. For a start, the set of formulas is as for the basic (multi) modal logic.

Definition 1 (Modal formula)**.** *A modal signature is a pair $\langle \mathsf{Atom}, \mathsf{Mod} \rangle$ where Atom and Mod are two countable, disjoint sets. We usually assume that Atom is infinite. The set of modal formulas over $\langle \mathsf{Atom}, \mathsf{Mod} \rangle$ is defined as*

$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid [m]\varphi,$$

for $a \in \mathsf{Atom}$, $m \in \mathsf{Mod}$. $\top$ and $\bot$ stand for an arbitrary tautology and contradiction, respectively. Connectives such as $\wedge, \rightarrow$ and $\langle m \rangle$, are defined as usual.

We will define a symmetry as a permutation of literals that preserve the structure of formulas in conjunctive normal form (CNF).

Definition 2 (Literals and modal CNF)**.** *A* literal l *is either an atom* a *or its negation* $\neg a$. *The set of literals over* Atom *is* $\mathsf{ALit} = \mathsf{Atom} \cup \{\neg a \mid a \in \mathsf{Atom}\}$.

A set of literals L *is* complete *if for each* $a \in \mathsf{Atom}$ *either* $a \in L$ *or* $\neg a \in L$. *It is* consistent *if for each* $a \in \mathsf{Atom}$ *either* $a \notin L$ *or* $\neg a \notin L$. *Any complete and consistent set of literals* L *defines a unique valuation* $v : \mathsf{Atom} \mapsto \{\top, \bot\}$ *as* $v(a) = \top$ *if* $a \in L$ *and* $v(a) = \bot$ *if* $\neg a \in L$. *For* $S \subseteq \mathsf{Atom}$, *the* consistent and complete set of literals generated by S *(notation* L_S*) is* $S \cup \{\neg a \mid a \in \mathsf{Atom} \backslash S\}$.

A modal formula is in conjunctive normal form (CNF) *if it is a conjunction of modal CNF clauses. A* modal CNF clause *is a disjunction of atom and modal literals. A* modal literal *is a formula of the form* $[m]C$ *or* $\neg[m]C$ *where* C *is a modal CNF clause. Every modal formula can be transformed into an equisatisfiable modal CNF formula in polynomial time (see [4, 22] for details).*

A modal CNF formula can be represented as a set of CNF clauses (which are to be interpreted conjunctively), and each clause can be represented as a set of atom and modal literals (which are to be interpreted disjunctively). With the set representation we can disregard the order and multiplicity in which clauses and literals appear. This will be important when we define symmetries below. In the rest of the paper we will assume that modal formulas are in modal CNF.

Example 1. *The modal formula* $\varphi = \langle m \rangle (p \wedge q \wedge p) \wedge [m]\neg r$ *is equisatisfiable to the CNF formula* $\varphi' = \{\{\neg[m]\{\neg p, \neg q\}\}, \{[m]\{\neg r\}\}\}$.

Up to now, we have not departed from the standard presentation of classical modal logic in important ways. The main change introduced by the coinductive approach is with the definition of model and semantic conditions.

Definition 3 (Models)**.** *Let* $\mathscr{S} = \langle \mathsf{Atom}, \mathsf{Mod} \rangle$ *be a modal signature and* W *be a fixed, non-empty set.* $\mathbf{Mods}_W$, *the class of all models with domain* W, *for the signature* $\mathscr{S}$, *is the class of all tuples* $\langle w, W, V, R \rangle$ *such that* $w \in W$, $V(v) \subseteq \mathsf{Atom}$ *for all* $v \in W$, *and*

$$R(m, v) \subseteq \mathbf{Mods}_W \text{ for } m \in \mathsf{Mod} \text{ and } v \in W.$$

Given a model $\mathscr{M} = \langle w, W, V, R \rangle$ *we will say that* w *is the point of evaluation and denote it as* $w^{\mathscr{M}}$, W *is the domain and denote it as* $|\mathscr{M}|$, V *is the valuation and denote it as* $V^{\mathscr{M}}$, *and* R *is the accessibility relation and denote it as* $R^{\mathscr{M}}$. $\mathbf{Mods}$ *denotes the class of all models over all domains,* $\mathbf{Mods} = \bigcup_W \mathbf{Mods}_W$.

Given $\mathscr{M} \in \mathbf{Mods}_W$, *let* $\mathrm{Ext}(\mathscr{M})$, *the* extension of $\mathscr{M}$, *be the smallest subset of* $\mathbf{Mods}_W$ *that contains* $\mathscr{M}$ *and is such that if* $\mathscr{N} \in \mathrm{Ext}(\mathscr{M})$, *then* $R^{\mathscr{N}}(m, v) \subseteq \mathrm{Ext}(\mathscr{M})$ *for all* $m \in \mathsf{Mod}$, $v \in W$.

The definition of a coinductive modal model is similar to the usual definition of a Kripke pointed model. The difference lies in the way the accessibility relation is defined. In particular, for each m and each state w, $R(m, w)$ is defined as the set of (potentially different) models accessible from w through the m modality. Observe that for each W, $\mathbf{Mods}_W$ is well-defined (coinductively), and so does $\mathbf{Mods}$, the class of all models. Our results will apply not only to $\mathbf{Mods}$ but to many of its subclasses. We will be interested in classes which are *closed under accessibility relations* (*closed classes* for short): $\mathscr{M} \in \mathscr{C}$ implies $\mathrm{Ext}(\mathscr{M}) \subseteq \mathscr{C}$. In the rest of the paper we will only consider classes of models closed under accessibility relations.

Example 2. *To illustrate the differences between coinductive modal models and Kripke models, consider the pointed Kripke model in Figure 1a, and its equivalent coinductive modal model in Figure 1b. The main difference is that the relation of a coinductive model leads to another coinductive model, whereas in a Kripke model the relation leads to another point of the same model.*

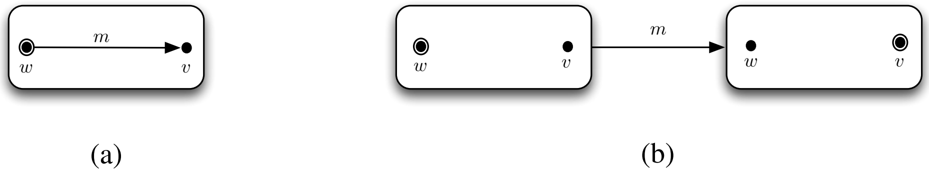


Figure 1: a) A Kripke model. b) Equivalent coinductive model.

We are now ready to introduce the definition of the satisfiability relation $\models$.

Definition 4 (Semantics)**.** *Let φ be a formula in modal CNF and $\mathscr{M} = \langle w, W, V, R \rangle$ a model in* **Mods***. We define $\models$ for CNF formulas, clauses and literals as*

$$
\begin{array}{lll}
\mathscr{M} \models \varphi & \textit{iff} & \text{for all clauses } C \in \varphi \text{ we have } \mathscr{M} \models C \\
\mathscr{M} \models C & \textit{iff} & \text{there is some literal } l \in C \text{ such that } \mathscr{M} \models l \\
\mathscr{M} \models a & \textit{iff} & a \in V(w) \text{ for } a \in \mathsf{Atom} \\
\mathscr{M} \models \neg a & \textit{iff} & a \notin V(w) \text{ for } a \in \mathsf{Atom} \\
\mathscr{M} \models [m]C & \textit{iff} & \mathscr{M}' \models C, \text{ for all } \mathscr{M}' \in R(m,w) \\
\mathscr{M} \models \neg[m]C & \textit{iff} & \mathscr{M} \not\models [m]C.
\end{array}
$$

For $\mathscr{C}$ a class of models, we write $\mathscr{C} \models \varphi$ whenever $\mathscr{M} \models \varphi$ for every $\mathscr{M}$ in $\mathscr{C}$, and we say that $\Gamma_{\mathscr{C}} = \{\varphi \mid \mathscr{C} \models \varphi\}$ is the logic *defined by $\mathscr{C}$.*

The set of models of a formula φ is the set $\mathsf{Mod}_{\mathscr{C}}(\varphi) = \{\mathscr{M} \mid \mathscr{M} \in \mathscr{C}$ and $\mathscr{M} \models \varphi\}$ (when $\mathscr{C}$ is clear from the context we will just write $\mathsf{Mod}(\varphi)$). We say that ψ can be inferred from *φ in $\mathscr{C}$ and write $\varphi \models_{\mathscr{C}} \psi$ if $\mathsf{Mod}_{\mathscr{C}}(\varphi) \subseteq \mathsf{Mod}_{\mathscr{C}}(\psi)$.*

As shown in [4], the logic $\Gamma_{\mathbf{Mods}}$ (generated by the class of all possible models) coincides with the basic multi-modal logic K. By properly restricting the model class we can capture different modal logics. Let us call a predicate P on models a *defining condition* for a class $\mathscr{C}$ whenever $\mathscr{C}$ is such that $\mathscr{M} \in \mathscr{C}$ if and only if $P(\mathscr{M})$ holds. Consider the signature $\mathscr{S} = \langle \mathsf{Atom}, \mathsf{Mod} \rangle$ where $\mathsf{Atom} = \mathsf{Prop} \cup \mathsf{Nom}$, $\mathsf{Mod} = \mathsf{Rel} \cup \{\mathsf{A}\} \cup \{@_i \mid i \in \mathsf{Nom}\}$; and $\mathsf{Prop} = \{p_1, p_2, \ldots\}$, $\mathsf{Nom} = \{n_1, n_2, \ldots\}$ and $\mathsf{Rel} = \{r_1, r_2, \ldots\}$ are mutually disjoint, countable infinite sets. In what follows, we will usually be interested in sub-languages of the language defined over $\mathscr{S}$ by Definition 1.

Class	Defining condition
$\mathscr{C}^{\mathsf{K}}_m$	$\mathscr{P}^{\mathsf{K}}_m(\mathscr{M}) \iff R^{\mathscr{M}}(m,w) \subseteq \{\langle v, \lvert\mathscr{M}\rvert, V^{\mathscr{M}}, R^{\mathscr{M}} \rangle \mid v \in \lvert\mathscr{M}\rvert\}, m \in \mathsf{Rel}$
$\mathscr{C}_{\mathsf{A}}$	$\mathscr{P}_{\mathsf{A}}(\mathscr{M}) \iff R^{\mathscr{M}}(\mathsf{A},w) = \{\langle v, \lvert\mathscr{M}\rvert, V^{\mathscr{M}}, R^{\mathscr{M}} \rangle \mid v \in \lvert\mathscr{M}\rvert\}$
$\mathscr{C}_{@_i}$	$\mathscr{P}_{@_i}(\mathscr{M}) \iff R^{\mathscr{M}}(@_i,w) = \{\langle v, \lvert\mathscr{M}\rvert, V^{\mathscr{M}}, R^{\mathscr{M}} \rangle \mid i \in V(v)\}, i \in \mathsf{Nom}$
$\mathscr{C}_{\mathsf{Nom}}$	$\mathscr{P}_{\mathsf{Nom}}(\mathscr{M}) \iff \{w \mid i \in V^{\mathscr{M}}(w)\}$ is a singleton, $\forall i \in \mathsf{Nom}$

Figure 2: Defining conditions for different modal logics

Figure 2 introduces a number of closed model classes by means of their defining conditions. Observe that $\mathscr{P}^{\mathsf{K}}_m$ is true for a model $\mathscr{M}$ if every successor of $w^{\mathscr{M}}$ is identical to $\mathscr{M}$ except perhaps on its point of evaluation. We call m a *relational modality* when it is interpreted in $\mathscr{C}^{\mathsf{K}}_m$ because over this class they behave as classical relational modalities [4].

We can capture different modal operators, like the ones from hybrid logics [2], by choosing the proper class of models. Predicates $\mathscr{P}_{\mathsf{A}}$ and $\mathscr{P}_{@_i}$, for instance, impose conditions on the point of evaluation of the accessible models restricting the evaluation to the class of models where the relation is, respectively, the total relation ($\forall xy.R(x,y)$) and the 'point to all i' relation ($\forall xy.R(x,y) \leftrightarrow i(y)$). Observe that whenever the atom i is interpreted as a singleton set, the 'point to all i' relation becomes the usual 'point to i' relation ($\forall xy.R(x,y) \leftrightarrow y = i$) of hybrid logics. Finally, predicate $\mathscr{P}_{\mathsf{Nom}}$ turns elements of Nom into nominals, i.e., true at a unique element of the domain of the model.

An interesting feature of this setting is that we can express the combination of modalities as the intersection of their respective classes. For example, $\mathscr{C}_{\mathscr{H}(@)}$, the class of models for the hybrid logic $\mathscr{H}(@)$, can be defined as follows:

$$\mathscr{C}_{\mathscr{H}(@)} = \mathscr{C}_{\mathsf{Nom}} \cap \mathscr{C}_{@} \cap \mathscr{C}_{\mathsf{Rel}}, \text{ where}$$
$$\mathscr{C}_{@} = \textstyle\bigcap_{i \in \mathsf{Nom}} \mathscr{C}_{@_i}, \text{ and } \mathscr{C}_{\mathsf{Rel}} = \textstyle\bigcap_{m \in \mathsf{Rel}} \mathscr{C}_m^{\mathsf{K}}.$$

The crucial characteristic of the coinductive approach is that all these different modal operators are captured using the same semantic condition introduced in Definition 4, all the details defining each particular operator are now introduced as properties of the accessibility relation. As a result, a unique notion of bisimulation is sufficient to cover all of them.

Definition 5 (Bisimulations)**.** *Given two models $\mathscr{M}$ and $\mathscr{M}'$ we say that $\mathscr{M}$* and *$\mathscr{M}'$* are bisimilar *(notation $\mathscr{M} \underline{\leftrightarrow} \mathscr{M}'$) if $\mathscr{M}\, Z\, \mathscr{M}'$ for some relation $Z \subseteq \mathrm{Ext}(\mathscr{M}) \times \mathrm{Ext}(\mathscr{M}')$ such that whenever $\langle w, W, V, R\rangle\, Z$ $\langle w', W', V', R'\rangle$ we have that:*

- ***Harmony:** $w \in V(a)$ iff $w' \in V'(a)$, for all $a \in \mathit{Atom}$.*
- ***Zig:** $\mathscr{N} \in R(m,w)$ implies $\mathscr{N}\, Z\, \mathscr{N}'$ for some $\mathscr{N}' \in R'(m,w')$.*
- ***Zag:**$\mathscr{N}' \in R'(m,w')$ implies $\mathscr{N}\, Z\, \mathscr{N}'$ for some $\mathscr{N} \in R(m,w)$.*

Such Z is called a bisimulation between *$\mathscr{M}$* and *$\mathscr{M}'$*.

The classic result of invariance of modal formulas under bisimulation [11] can easily be proved.

Theorem 1. *If $\mathscr{M} \underline{\leftrightarrow} \mathscr{M}'$, then $\mathscr{M} \models \varphi$ iff $\mathscr{M}' \models \varphi$, for all φ.*

As stated in [4], this general notion of bisimulation works for every modal logic definable as a closed subclass of **Mods**.

3 Modal Symmetries

We will show that consistent symmetries for modal formulas behave similarly as in the propositional case and, hence, could assist in modal theorem proving. Let us start by introducing the basic notions.

Definition 6 (Permutation)**.** *A* permutation *is a bijective function $\sigma : \mathit{ALit} \mapsto \mathit{ALit}$. For L a set of literals, $\sigma(L) = \{\sigma(l) \mid l \in L\}$.*

We will be mostly interested in permutations that only involve a finite number of elements (i.e., $\sigma(l) = l$ except for a finite number of literals). In these cases we can succinctly define a permutation using cycle notation, e.g., $\sigma = (p\ \neg q)(\neg p\ q)$ is the permutation that makes $\sigma(p) = \neg q$, $\sigma(\neg q) = p$, $\sigma(\neg p) = q$ and $\sigma(q) = \neg p$; and $\sigma = (p\ q\ r)(\neg p\ \neg q\ \neg r)$ is the permutation $\sigma(p) = q$, $\sigma(q) = r$ and $\sigma(r) = p$ and similarly for the negations.

Because, in our language, atoms may occur in some modalities (like $@_i$) we should take some care when we apply permutations to modal formulas. We will say that a modality is *indexed by atoms* if its definition depends on the value of an atom. If m is indexed by an atom a we will sometimes write $m(a)$.

Definition 7 (Permutation of a formula)**.** *Let φ be a formula in modal CNF and σ a permutation. We define $\sigma(\varphi)$ recursively. If φ is a literal l then $\sigma(\varphi)$ is simply $\sigma(l)$. For the other cases:*

$$\begin{array}{rcll} \sigma(C) & = & \{\sigma(A) \mid A \in C\} & \textit{for } C \textit{ a modal CNF clause or formula} \\ \sigma([m]C) & = & [\sigma(m)]\sigma(C) & \\ \sigma(\neg[m]C) & = & \neg[\sigma(m)]\sigma(C) & \end{array}$$

where $\sigma(m) = \sigma(m(a)) = m(\sigma(a))$ if m is indexed by a, and $\sigma(m) = m$ otherwise.

Definition 8. *We say that a permutation σ is* consistent *if for every literal l, $\sigma(\neg l) = \neg\sigma(l)$. We say that a permutation σ is a* symmetry *for φ if $\varphi = \sigma(\varphi)$, when conjunctions and disjunctions in φ are represented as sets.*

Example 3. *Trivially, the identity permutation $\sigma(l) = l$ is a consistent symmetry of any formula φ. More interestingly, consider $\varphi = \{\{\neg p, r\}, \{q, r\}, \{r, [m]\{\neg p, q\}\}\}$, then the permutation $\sigma = (p\ \neg q)(\neg p\ q)$ is a consistent symmetry of φ.*

Definition 9 (Permutation of a model)**.** *Let σ be a permutation and $\mathscr{M} = \langle w, W, V, R\rangle$ a model. Then $\sigma(\mathscr{M}) = \langle w, W, V', R'\rangle$, where,*

$$\begin{array}{rcll} V'(v) & = & \sigma(L_{V(v)}) \cap \mathsf{Atom} & \textit{for all } v \in W, \textit{ and,} \\ R'(m, v) & = & \{\sigma(\mathscr{N}) \mid \mathscr{N} \in R(\sigma(m), v)\} & \textit{for all } m \in \mathsf{Mod} \textit{ and } v \in W. \end{array}$$

For M a set of models, $\sigma(M) = \{\sigma(\mathscr{M}) \mid \mathscr{M} \in M\}$.

The main ingredient to prove that symmetries preserve entailment is the relation between models that we call σ-simulation.

Definition 10 (σ-simulation)**.** *Let σ be a permutation. A* σ-simulation *between models $\mathscr{M} = \langle w, W, V, R\rangle$ and $\mathscr{M}' = \langle w', W', V', R'\rangle$ is a non-empty relation $Z \subseteq \mathrm{Ext}(\mathscr{M}) \times \mathrm{Ext}(\mathscr{M}')$ that satisfies the following conditions:*

- ***Root:*** *$\mathscr{M} Z \mathscr{M}'$.*
- ***Atomic Harmony:*** *$l \in L_{V(w)}$ iff $\sigma(l) \in L_{V'(w')}$.*
- ***Zig:*** *$\mathscr{M} Z \mathscr{M}'$ and $\mathscr{N} \in R(m, w)$ then $\mathscr{N} Z \mathscr{N}'$ for some $\mathscr{N}' \in R'(\sigma(m), w')$.*
- ***Zag:*** *$\mathscr{M} Z \mathscr{M}'$ and $\mathscr{N}' \in R'(m, w')$ then $\mathscr{N} Z \mathscr{N}'$ for some $\mathscr{N} \in R(\sigma^{-1}(m), w)$.*

We say that two models $\mathscr{M}$ and $\mathscr{M}'$ are σ-similar (notation $\mathscr{M} \underline{\to}_\sigma \mathscr{M}'$) if there is a σ-simulation Z between them.

Notice that while $\mathscr{M} \underline{\to}_\sigma \mathscr{M}'$ implies $\mathscr{M}' \underline{\to}_{\sigma^{-1}} \mathscr{M}$, $\underline{\to}_\sigma$ is not a symmetric relation (in particular σ might differ from σ^{-1}). From the definition of σ-simulations it intuitively follows that while they do not preserve validity of modal formulas (as is the case with bisimulations) they do preserve validity of *permutations* of formulas.

Proposition 1. *Let σ be a consistent permutation, φ a modal CNF formula and $\mathscr{M} = \langle w, W, V, R\rangle$, $\mathscr{M}' = \langle w', W', V', R'\rangle$ models such that $\mathscr{M} \underline{\to}_\sigma \mathscr{M}'$. Then $\mathscr{M} \models \varphi$ iff $\mathscr{M}' \models \sigma(\varphi)$.*

Proof. The proof is by induction on φ. Base Case: Suppose $\varphi = a$ then, $\mathscr{M} \models a$ iff $a \in V(w)$ iff $a \in L_{V(w)}$ iff, by definition of σ-simulation, $\sigma(a) \in L_{V'(w')}$ iff $\mathscr{M}' \models \sigma(a)$.

Suppose $\varphi = \neg a$ then, $\mathscr{M} \models \neg a$ iff $a \notin V(w)$ iff $\neg a \in L_{V(w)}$ iff, by definition of σ-simulation, $\sigma(\neg a) = \neg\sigma(a) \in L_{V'(w')}$ iff $\sigma(a) \notin V'(w')$ iff $\mathscr{M}' \models \neg\sigma(a)$.

Inductive Step: Suppose $\varphi = [m]\psi$. Then $\mathscr{M} \models [m]\psi$ iff $\mathscr{N} \models \psi$ for all $\mathscr{N} \in R(m,w)$. Given that $\mathscr{M} \underrightarrow{\to}_\sigma \mathscr{M}'$, by Zig we know that for all $\mathscr{N}$ exist $\mathscr{N}'$ such that $\mathscr{N} \underrightarrow{\to}_\sigma \mathscr{N}'$ and $\mathscr{N}' \in R'(\sigma(m),w')$. Then, by inductive hypothesis, $\mathscr{N}' \models \sigma(\psi)$ for all $\mathscr{N}' \in R'(\sigma(m),w')$. Then $\mathscr{M}' \models [\sigma(m)]\sigma(\psi) = \sigma([m]\psi)$. The converse follows by the same argument, using Zag and the inductive hypothesis. □

An easily verifiable consequence of Definitions 9 and 10 is that $\mathscr{M}$ and $\sigma(\mathscr{M})$ are always σ-similar.

Proposition 2. *Let σ be a consistent permutation and $\mathscr{M} = \langle w,W,V,R\rangle$ a model. Then $\mathscr{M} \underrightarrow{\to}_\sigma \sigma(\mathscr{M})$.*

Proof. Let us define the relation $Z = \{(\mathscr{N}, \sigma(\mathscr{N})) \mid \mathscr{N} \in \mathrm{Ext}(\mathscr{M})\}$ and show that it is a σ-simulation between $\mathscr{M}$ and $\sigma(\mathscr{M})$. The *Zig* and *Zag* conditions are trivial by definition of $\sigma(\mathscr{M})$.

For *Atomic Harmony*, we have to check that $l \in L_{V(w)}$ iff $\sigma(l) \in L_{V'(w)}$. From the definition of $\sigma(\mathscr{M})$, $L_{V'(w)} = \sigma(L_{V(w)})$, hence if $l \in L_{V(w)}$ then $\sigma(l) \in \sigma(L_{V(w)})$. Moreover, $\sigma(L_{V(w)})$ is a complete set of literals because $L_{V(w)}$ is a complete set of literals and σ is a consistent permutation, and hence the converse also follows. □

Interestingly if σ is a symmetry of φ then for any model $\mathscr{M}$, $\mathscr{M}$ is a model of φ if and only if $\sigma(\mathscr{M})$ is. This will be a direct corollary of the following proposition in the particular case when σ is a symmetry and hence $\sigma(\varphi) = \varphi$.

Proposition 3. *Let σ be a consistent permutation, $\mathscr{M}$ a model and φ a modal CNF formula. Then $\mathscr{M} \models \varphi$ iff $\sigma(\mathscr{M}) \models \sigma(\varphi)$.*

Proof. This lemma follows directly from Proposition 2 ($\mathscr{M} \underrightarrow{\to}_\sigma \sigma(\mathscr{M})$) and Proposition 1. □

Corollary 1. *If σ is a symmetry of φ then $\mathscr{M} \in \mathsf{Mod}(\varphi)$ iff $\sigma(\mathscr{M}) \in \mathsf{Mod}(\varphi)$.*

To clarify the implications of the Corollary 1, consider the following example.

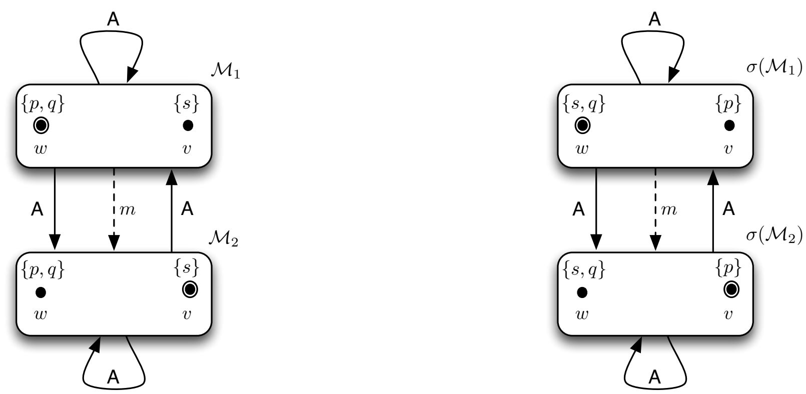
Example 4. *Let $\varphi = (p \vee q \vee r) \wedge (s \vee q \vee r) \wedge (\neg p \vee \neg s) \wedge \langle m\rangle(p \vee s) \wedge [\mathsf{A}](\neg r)$. Then $\sigma = (p\ s)(\neg p\ \neg s)$ is a symmetry of φ. Now consider the model $\mathscr{M}_1$ of Figure 3a. It is easy to verify that $\mathscr{M}_1 \models \varphi$. Now, given that σ is a symmetry of φ, by Corollary 1 $\sigma(\mathscr{M}_1) \models \varphi$, which is easy to verify in the model of Figure 3b.*

The notion of σ-simulation in coinductive modal models is general enough to be applicable to a wide range of modal logics. Notice though, that our definition of σ-simulation makes no assumption about the models being in the same class. Consider, for example, a model $\mathscr{M} \in \mathscr{C}_{\mathscr{H}(@)}$ and a permutation $\sigma = (i\ p)(\neg i\ \neg p)$ for $i \in \mathsf{Nom}$, $p \in \mathsf{Prop}$. By the defining condition $\mathscr{C}_{\mathscr{H}(@)}$, nominals in $\mathscr{M}$ are true at a unique element in the domain, but this does not necessary hold for $\sigma(\mathscr{M})$, and hence $\sigma(\mathscr{M})$ might not be in $\mathscr{C}_{\mathscr{H}(@)}$. Hence, when working with subclasses of **Mods** we will often have to require additional conditions of a permutation σ to ensure that for every $\mathscr{M}$, $\sigma(\mathscr{M})$ is in the intended class.

Definition 11. *Let σ be a permutation and $\mathscr{C}$ a closed class of models. We say that $\mathscr{C}$ is* closed under σ *if for every $\mathscr{M} \in \mathscr{C}$, $\sigma(\mathscr{M}) \in \mathscr{C}$.*

Example 5 (σ-simulation in Hybrid Logic)**.** *Consider the class $\mathscr{C}_{\mathscr{H}(@)}$ again. As we mentioned $\mathscr{C}_{\mathscr{H}(@)}$ is not closed under arbitrary permutations, but it is closed under permutations that send nominals to nominals.*

Everything is now in place to show that modal entailment is preserved under symmetries.



(a) (b)

Figure 3: a) Model $\mathscr{M}_1$. b) Model $\sigma(\mathscr{M}_1)$.

Theorem 2. *Let φ and ψ be modal formulas, let σ be a consistent symmetry of φ and $\mathscr{C}$ a class of models closed under σ. Then $\varphi \models_{\mathscr{C}} \psi$ if and only if $\varphi \models_{\mathscr{C}} \sigma(\psi)$.*

Proof. We first show that under the hypothesis of the theorem the following property holds

Claim: $\mathsf{Mod}_{\mathscr{C}}(\varphi) = \sigma(\mathsf{Mod}_{\mathscr{C}}(\varphi))$.

[$\supseteq$] Let $\mathscr{N} \in \sigma(\mathsf{Mod}_{\mathscr{C}}(\varphi))$ and $\mathscr{M} \in \mathsf{Mod}_{\mathscr{C}}(\varphi)$ be such that $\mathscr{N} = \sigma(\mathscr{M})$. Then $\mathscr{M} \models \varphi$ and by Corollary 1, $\sigma(\mathscr{M}) \models \varphi$. Given that $\mathscr{C}$ is closed under σ, $\sigma(\mathscr{M}) \in \mathscr{C}$ and, hence, $\sigma(\mathscr{M}) \in \mathsf{Mod}_{\mathscr{C}}(\varphi)$.

[$\subseteq$] Let $\mathscr{M} \in \mathsf{Mod}_{\mathscr{C}}(\varphi)$, then $\mathscr{M} \models \varphi$. By Corollary 1, $\sigma(\mathscr{M}) \models \varphi$ and, given that $\mathscr{C}$ is closed under σ, $\sigma(\mathscr{M}) \in \mathscr{C}$. Therefore, $\sigma(\mathscr{M}) \in \mathsf{Mod}_{\mathscr{C}}(\varphi)$. Because σ is a permutation there is $n \in \mathbb{N}$, $n \geq 1$, such that $\sigma^n(\mathscr{M}) = \mathscr{M}$ and hence $\mathscr{M} \in \sigma(\mathsf{Mod}_{\mathscr{C}}(\varphi))$.

Now, we have to prove that $\varphi \models \psi$ if and only if $\varphi \models \sigma(\psi)$. By definition, $\varphi \models \psi$ iff and only if $\mathsf{Mod}_{\mathscr{C}}(\varphi) \models \psi$. By Proposition 3, this is the case if and only if $\sigma(\mathsf{Mod}_{\mathscr{C}}(\varphi)) \models \sigma(\psi)$.

Given that σ is a symmetry of φ, by the Claim above, $\sigma(\mathsf{Mod}_{\mathscr{C}}(\varphi)) \models \sigma(\psi)$ if and only if $\mathsf{Mod}_{\mathscr{C}}(\varphi) \models \sigma(\psi)$, which by definition means $\varphi \models \sigma(\psi)$. $\square$

Theorem 2 provides an inexpensive inference mechanism that can be used in every situation where entailment is involved during modal automated reasoning. Indeed applying a permutation on a formula is a calculation that is arguably computationally cheaper than a tableau expansion or a resolution step thereby, new formulas obtained by this mean may reduce the total running time of an inference task. In the case of propositional logic, the strengthening of the learning mechanism has already shown its results in [8]. In the case of modal logic, it remains to see when cases of $\varphi \models \psi$ occur during a decision procedure, and how to better take advantage of them.

4 Layered Permutations

In this section we present the notion of layered permutations. First, we present a definition of the tree model property [11] for coinductive modal models that will be useful when introducing layered permutations.

Given a model $\mathscr{M}$, a (finite) *path rooted* at $\mathscr{M}$ is a sequence $\pi = (\mathscr{M}_0, m_1, \mathscr{M}_1, \dots, m_k, \mathscr{M}_k)$, for $m_i \in$ Mod where $\mathscr{M}_0 = \mathscr{M}$, $k \geq 0$, and $\mathscr{M}_i \in R(m_i, w^{\mathscr{M}_{i-1}})$ for $i = 1, \dots, k$. For a path $\pi = (\mathscr{M}_0, m_1, \mathscr{M}_1, \dots, m_k, \mathscr{M}_k)$ we define $\mathit{first}(\pi) = \mathscr{M}_0$, $\mathit{last}(\pi) = \mathscr{M}_k$, and $\mathit{length}(\pi) = k$. We denote the set of all paths rooted at $\mathscr{M}$ as $\Pi[\mathscr{M}]$. A coinductive tree model is a model that has a unique path to every reachable model (every model in $\mathrm{Ext}(\mathscr{M})$). Formally we can define the class of all coinductive tree models, $\mathscr{C}_{Tree}$, with the following defining condition:

$$\mathscr{C}_{Tree} := P_{Tree}(\mathscr{M}) \iff \mathit{last} : \Pi[\mathscr{M}] \mapsto \mathrm{Ext}(\mathscr{M}) \text{ is bijective.}$$

For example, the *unravelling* construction (in its version for coinductive modal models) shown below always defines a model in $\mathscr{C}_{Tree}$.

Definition 12 (Model Unravelling)**.** *Given a model $\mathscr{M} = \langle w, W, V, R\rangle$, the unravelling of $\mathscr{M}$, (notation $\mathscr{T}(\mathscr{M})$), is the rooted coinductive model $\mathscr{T}(\mathscr{M}) = \langle (\mathscr{M}), \Pi[\mathscr{M}], V', R'\rangle$ where*

$$\begin{aligned} V'(\pi) &= V(w^{\mathit{last}(\pi)}), \text{ for all } \pi \in \Pi[\mathscr{M}], \\ R'(m,\pi) &= \{\langle \pi', \Pi[\mathscr{M}], V', R'\rangle \mid \mathit{last}(\pi') \in R(m, \mathit{last}(\pi))\}, \quad \text{for } m \in \mathit{Mod}, \pi \in \Pi[\mathscr{M}]. \end{aligned}$$

It is easy to verify that given a model $\mathscr{M}$, its unravelling $\mathscr{T}(\mathscr{M})$ is a tree ($\mathscr{T}(\mathscr{M}) \in \mathscr{C}_{Tree}$) and, as expected, $\mathscr{M}$ and $\mathscr{T}(\mathscr{M})$ are bisimilar.

In what follows, we will use trees to define a more flexible family of symmetries that we call *layered symmetries*. The following will give a sufficient condition ensuring that layered symmetries also preserve entailment.

Definition 13 (Tree model closure property)**.** *We say that a class $\mathscr{C}$ of models is* closed under trees *if for every model $\mathscr{M} \in \mathscr{C}$ there is a tree model $\mathscr{T} \in \mathscr{C}$ such that $\mathscr{M} \underline{\leftrightarrow} \mathscr{T}$.*

From this definition, it follows that a class of models $\mathscr{C}$ closed under unravellings ($\mathscr{T}(\mathscr{M}) \in \mathscr{C}$ for all $\mathscr{M} \in \mathscr{C}$) is also closed under trees.

Example 6. *Trivially the class* **Mods** *(i.e., the basic modal logic) is closed under trees, and so does the class $\mathscr{C}_{\mathsf{KAlt}_1}$ of models where the accessibility relation is a partial function. Many classes like $\mathscr{C}_{\mathsf{A}}$, $\mathscr{C}_{@_i}$ and $\mathscr{C}_{\mathsf{Nom}}$ fail to be closed under trees.*

Logics defined over classes closed under trees have an interesting property: there is a direct correlation between the syntactical modal depth of the formula and the depth in a tree model satisfying it. In tree models, a notion of layer is induced by the depth (distance from the root) of the nodes in the model. Similarly,in modal formulas, a notion of layer is induced by the nesting of the modal operators. A consequence of this correspondence is that literals occurring at different formula layers are semantically independent of each other (see [3] for further discussion), i.e., at different layers the same literal can be assigned a different value.

Example 7. *Consider the formula $\varphi = (p \vee q) \wedge (r \vee \neg\Box(\neg p \vee q \vee \Box\neg r))$ and a tree model $\mathscr{M}$ of the formula. Figure 4 shows the layers induced by the modal depth of the formula and the corresponding depth in $\mathscr{M}$.*

The independence between literals at different layers enables us to give a more flexible notion of a permutation that we will call *layered permutation*. Key to the notion of layered permutation is that of a *permutation sequence*.

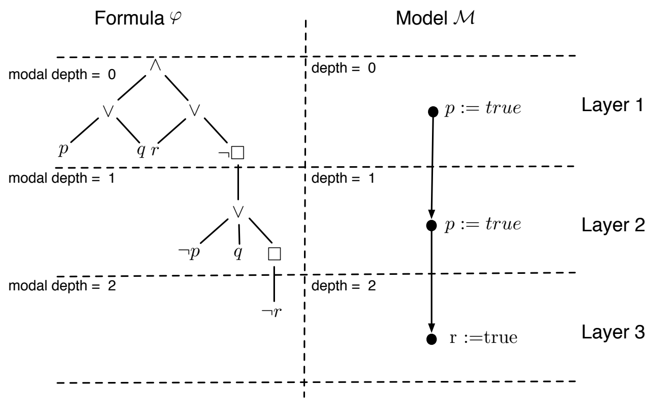


Figure 4: Induced layering on a model and a formula.

Definition 14 (Permutation Sequence)**.** *We define a finite permutation sequence $\bar{\sigma}$ as either $\bar{\sigma} = \langle\rangle$ (i.e., $\bar{\sigma}$ is the empty sequence) or $\bar{\sigma} = \sigma : \bar{\sigma}_2$ with σ a permutation and $\bar{\sigma}_2$ a permutation sequence. Alternatively we can use the notation $\bar{\sigma} = \langle \sigma_1, \dots, \sigma_n \rangle$ instead of $\bar{\sigma} = \sigma_1 : \dots : \sigma_n : \langle\rangle$.*

Let $|\bar{\sigma}| = n$ be the length of $\bar{\sigma}$ ($\langle\rangle$ has length 0). For $1 \leq i \leq n$, we write $\bar{\sigma}_i$ for the subsequence that starts from the i^{th} element of $\bar{\sigma}$. For $i \geq n$, we define $\bar{\sigma}_i = \langle\rangle$. In particular $\bar{\sigma} = \bar{\sigma}_1$. Given a permutation sequence $\sigma_1 : \bar{\sigma}_2$ we define $head(\sigma_1 : \bar{\sigma}_2) = \sigma_1$ and $head(\langle\rangle) = \sigma_{Id}$, where σ_{Id} is the identity permutation. We say that a permutation sequence is consistent *if all of its permutations are consistent.*

Applying a permutation sequence to a formula in modal CNF can be defined as follows:

Definition 15 (Layered permutation of a formula)**.** *Let φ be a formula in modal CNF and $\bar{\sigma}$ a permutation sequence. We define $\bar{\sigma}(\varphi)$ recursively:*

$$\begin{array}{rcll} \langle\rangle(\varphi) & = & \varphi & \\ (\sigma_1 : \bar{\sigma}_2)(l) & = & \sigma_1(l) & \text{for } l \in \mathsf{ALit} \\ (\sigma_1 : \bar{\sigma}_2)([m]C) & = & [\sigma_1(m)]\bar{\sigma}_2(C) & \\ \bar{\sigma}(C) & = & \{\bar{\sigma}(A) \mid A \in C\} & \text{for } C \text{ a clause or a formula.} \end{array}$$

Notice that this definition is well defined even if the modal depth of the formula is greater than the size of the permutation sequence. Layered permutations let us use a different permutation at each modal depth. This enables symmetries (layered symmetries) to be found, that wouldn't be found otherwise.

Example 8. *Consider the formula $\varphi = (p \vee [m](p \vee \neg r)) \wedge (\neg q \vee [m](\neg p \vee r))$. If we stick to the non-layered symmetry definition, this formula has no symmetries. However, the permutation sequence $\langle \sigma_1, \sigma_2 \rangle$ generated by $\sigma_1 = (p\ \neg q)$ and $\sigma_2 = (p\ \neg r)$ is a symmetry.*

From now on we can mostly repeat the work we did in the previous section to arrive to a result similar to Theorem 2 but involving permutation sequences, with one caveat: the obvious extension of the notion of permutated model $\sigma(\mathscr{M})$ to layered permutations is ill defined if $\mathscr{M}$ is not a tree. Hence, we need the additional requirement that the class $\mathscr{C}$ of models is closed under trees for the result to go through.

Definition 16 (Layered Permutation of a model)**.** *Let $\bar{\sigma}$ be a permutation sequence and $\mathscr{M} = \langle w, W, V, R \rangle$ a tree model. Then $\bar{\sigma}(\mathscr{M}) = \langle w, W, V', R' \rangle$, where,*

$$\begin{array}{rcll} V'(v) & = & head(\bar{\sigma})(L_{V(v)}) \cap \mathsf{Atom} & \text{for all } v \in W \text{, and,} \\ R'(m,v) & = & \{\bar{\sigma}_2(\mathscr{N}) \mid \mathscr{N} \in R(head(\bar{\sigma})(m), v)\} & \text{for all } m \in \mathsf{Mod} \text{ and } v \in W. \end{array}$$

For M a set of tree models, $\bar{\sigma}(M) = \{\bar{\sigma}(\mathscr{M}) \mid \mathscr{M} \in M\}$.

We can now extend the notion of σ-simulation to permutation sequences.

Definition 17 ($\bar{\sigma}$-simulation)**.** *Let $\bar{\sigma}$ be a permutation sequence. A* $\bar{\sigma}$-simulation *between models $\mathscr{M} = \langle w, W, V, R\rangle$ and $\mathscr{M}' = \langle w', W', V', R'\rangle$ is a family of relations $Z_{\bar{\sigma}_i} \subseteq \mathrm{Ext}(\mathscr{M}) \times \mathrm{Ext}(\mathscr{M}')$, $1 \leq i$, that satisfies the following conditions:*

- ***Root:*** *$\mathscr{M} Z_{\bar{\sigma}_1} \mathscr{M}'$.*
- ***Atomic Harmony:*** *If $wZ_{\bar{\sigma}_i}w'$ then $l \in L_{V(w)}$ iff $head(\bar{\sigma}_i)(l) \in L_{V'(w')}$.*
- ***Zig:*** *$\mathscr{M} Z_{\bar{\sigma}_i} \mathscr{M}'$ and $\mathscr{N} \in R(m,w)$ then $\mathscr{N} Z_{\bar{\sigma}_{i+1}} \mathscr{N}'$ for some $\mathscr{N}' \in R'(head(\bar{\sigma}_i)(m), w')$.*
- ***Zag:*** *$\mathscr{M} Z_{\bar{\sigma}_i} \mathscr{M}'$ and $\mathscr{N}' \in R'(m,w')$ then $\mathscr{N} Z_{\bar{\sigma}_{i+1}} \mathscr{N}'$ for some $\mathscr{N} \in R(head(\bar{\sigma}_i)^{-1}(m), w)$.*

We say that two models $\mathscr{M}$ and $\mathscr{M}'$ are $\bar{\sigma}$-similar (notation $\mathscr{M} \underline{\rightarrow}_{\bar{\sigma}} \mathscr{M}'$), if there is a $\bar{\sigma}$-simulation between them.

An important remark about the previous definition is that it does not make any assumption about the size of the permutation sequence. In fact, it is well defined even if the permutation sequence at hand is the empty sequence. In that case, it just behave as the identity permutation at each layer, thus the relation defines a bisimulation between the models.

Given a closed class of tree models $\mathscr{C}$ and $\bar{\sigma}$ a permutation sequence, we say that $\mathscr{C}$ is *closed under* $\bar{\sigma}$ if for every $\mathscr{M} \in \mathscr{C}$, $\bar{\sigma}(\mathscr{M}) \in \mathscr{C}$.

Now we are ready to prove the main result concerning layered symmetries and entailment.

Theorem 3. *Let φ and ψ be modal formulas and let $\bar{\sigma}$ be a consistent permutation sequence, and let $\mathscr{C}$ be a class of models closed under trees and $\mathscr{C} \cap \mathscr{C}_{Tree}$ closed under $\bar{\sigma}$. If $\bar{\sigma}$ is a symmetry of φ then for any ψ we have that $\varphi \models_{\mathscr{C}} \psi$ if and only if $\varphi \models_{\mathscr{C}} \bar{\sigma}(\psi)$.*

Proof. We first show that under the hypothesis of the theorem the following two properties hold.

Claim 1: $\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi) = \bar{\sigma}(\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi))$.

The argument is the same as for the Claim in Theorem 2 but using permutation sequences.

Claim 2: $\mathsf{Mod}_{\mathscr{C}}(\varphi) \models_{\mathscr{C}} \varphi$ iff $\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi) \models_{\mathscr{C}} \varphi$.

The left-to-right direction is trivial by the fact that $\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi) \subseteq \mathsf{Mod}_{\mathscr{C}}(\varphi)$. For the other direction, assume $\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi) \models_{\mathscr{C}} \varphi$ and $\mathsf{Mod}_{\mathscr{C}}(\varphi) \not\models_{\mathscr{C}} \varphi$. Then there is $\mathscr{M} \in \mathsf{Mod}_{\mathscr{C}}(\varphi)$ such that $\mathscr{M} \not\models_{\mathscr{C}} \varphi$. But we know that $\mathscr{M} \underline{\leftrightarrow} \mathscr{T}$, and $\mathscr{T} \in \mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi)$. Hence $\mathscr{T} \models_{\mathscr{C}} \varphi$ which contradicts our assumption.

It rest to prove that $\varphi \models_{\mathscr{C}} \psi$ if and only if $\varphi \models_{\mathscr{C}} \bar{\sigma}(\psi)$. By definition, $\varphi \models_{\mathscr{C}} \psi$ if and only if $\mathsf{Mod}_{\mathscr{C}}(\varphi) \models_{\mathscr{C}} \psi$. By Claim 2, this is case if and only if $\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi) \models_{\mathscr{C}} \psi$. By the layered version of Proposition 3, this is the case if and only if $\bar{\sigma}(\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi)) \models_{\mathscr{C}} \bar{\sigma}(\psi)$. Given that $\bar{\sigma}$ is a symmetry of φ, by Claim 1, $\bar{\sigma}(\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi)) \models_{\mathscr{C}} \bar{\sigma}(\psi)$ if and only if $\mathsf{Mod}_{\mathscr{C} \cap \mathscr{C}_{Tree}}(\varphi) \models_{\mathscr{C}} \bar{\sigma}(\psi)$, which by Claim 2 is the case if and only if $\mathsf{Mod}_{\mathscr{C}}(\varphi) \models_{\mathscr{C}} \bar{\sigma}(\psi)$ which by definition means that $\varphi \models_{\mathscr{C}} \bar{\sigma}(\psi)$. □

5 Conclusions and further work

The notion of symmetry has been well studied in propositional logic, and various optimizations of decision procedures based on it are known. In this article, we extend the notion of syntactic and semantic symmetries to many different modal logics using the framework of coinductive models. The main contribution is that a global symmetry σ preserves entailments whenever the class is closed by σ. For example, arbitrary global symmetries preserve entailments in the basic modal logic, but for the hybrid logic $\mathcal{H}(@)$ we can only consider symmetries that maps nominals to nominals. The second contribution of the paper is to show that if the class of models is closed under trees, then the more flexible notion of layered symmetry also preserve entailments.

To arrive at the previous results, we defined the concept of σ-simulation and show that it preserves σ-permutation of formulas. We then presented permutation sequences $\bar{\sigma}$, and $\bar{\sigma}$-simulations. Permutation sequences are relevant in those classes of models that are closed under trees. This property enables the use of layered symmetries, a notion that can capture more symmetries than the ordinary symmetry definition. Indeed, layered symmetries can be detected independently within atoms at each modal depth of a formula. $\bar{\sigma}$-simulations extend the notion of σ-simulations to permutation sequences, and enabled us to prove that layered symmetries also preserve entailment.

Preliminary results on modal symmetries were discussed in [21], were we also developed an efficient algorithm to detect symmetries for the basic modal logic, and empirically verified that many modal problems (both randomly and hand generated) contain symmetries.

Our ongoing research focuses on the incorporation of symmetry information into a modal tableau calculi such as [19, 18] or modal resolution calculi such as [5]. One promising theme that we will investigate in the future is permutations involving also modal literals. Such a general notion of permutation has more convoluted semantic consequences, since an adequate definition of σ-simulation between models will have to map literals to models.

References

[1] F. Aloul, A. Ramani, I. Markov & K. Sakallah (2003): *Solving difficult instances of Boolean satisfiability in the presence of symmetry*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 22(9), pp. 1117–1137, doi:10.1109/TCAD.2003.816218.

[2] C. Areces & B. ten Cate (2006): *Hybrid Logics*. In P. Blackburn, F. Wolter & J. van Benthem, editors: Handbook of Modal Logics, Elsevier, pp. 821–868.

[3] C. Areces, R. Gennari, J. Heguiabehere & M. de Rijke (2000): *Tree-Based Heuristics in Modal Theorem Proving*. In: Proceedings of ECAI'2000, Berlin, Germany, pp. 199–203.

[4] C. Areces & D. Gorín (2010): *Coinductive models and normal forms for modal logics (or how we learned to stop worrying and love coinduction)*. Journal of Applied Logic 8(4), pp. 305–318.

[5] C. Areces & D. Gorín (2011): *Resolution with Order and Selection for Hybrid Logics*. Journal of Automated Reasoning 46(1), pp. 1–42.

[6] G. Audemard (2002): *Reasoning by symmetry and function ordering in finite model generation*. In: Proceedings of CADE-18, pp. 226–240.

[7] G. Audemard, B. Mazure & L. Sais (2004): *Dealing with Symmetries in Quantified Boolean Formulas*. In: Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT'04), pp. 257–262.

[8] B. Benhamou, T. Nabhani, R. Ostrowski & M. Saidi (2010): *Enhancing Clause Learning by Symmetry in SAT Solvers*. In: Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 329–335.

[9] B. Benhamou & L. Sais (1992): *Theoretical Study of Symmetries in Propositional Calculus and Applications*. In: Proceedings of CADE-11, pp. 281–294.

[10] B. Benhamou & L. Sais (1994): *Tractability Through Symmetries in Propositional Calculus*. *Journal of Automated Reasoning* 12(1), pp. 89–102.

[11] P. Blackburn, M. de Rijke & Y. Venema (2001): *Modal Logic*. Cambridge University Press.

[12] P. Blackburn, J. van Benthem & F. Wolter (2006): *Handbook of Modal Logic*. *Studies in Logic and Practical Reasoning* 3, Elsevier Science Inc., New York, NY, USA.

[13] C. Brown, L. Finkelstein & P. Purdom, Jr. (1996): *Backtrack searching in the presence of symmetry*. *Nordic Journal of Computing* 3(3), pp. 203–219.

[14] J. Crawford (1992): *A Theoretical Analysis of Reasoning By Symmetry in First-Order Logic*. In: *Proceedings of AAAI Workshop on Tractable Reasoning*, San Jose, CA, pp. 17–22.

[15] J. Crawford, M. Ginsberg, E. Luks & A. Roy (1996): *Symmetry-Breaking Predicates for Search Problems*. In: *Proceedings of KR 1996*, pp. 148–159.

[16] D. Dharbe, P. Fontaine, S. Merz & B. Woltzenlogel Paleo (2011): *Exploiting Symmetry in SMT Problems*. In: *Proceedings of CADE-23*, *Lecture Notes in Computer Science* 6803, Springer Berlin Heidelberg, pp. 222–236.

[17] N. Een & N. Sörensson (2003): *An Extensible SAT-solver*. In: *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT'03)*, pp. 502–518.

[18] G. Hoffmann (2010): *Lightweight Hybrid Tableaux*. *Journal of Applied Logic* 8(4), pp. 397–408.

[19] M. Kaminski & G. Smolka (2009): *Terminating Tableau Systems for Hybrid Logic with Difference and Converse*. *Journal of Logic, Language and Information* 18(4), pp. 437–464.

[20] B. Krishnamurthy (1985): *Short Proofs for Tricky Formulas*. *Acta Informatica* 22(3), pp. 253–275.

[21] E. Orbe, C. Areces & G. Infante-López (2012): *A Note about Modal Symmetries*. Technical Report, FaMAF, UNC. Available at `http://www.famaf.unc.edu.ar/publicaciones/publicaciones.html#publ_serie_a_compu`.

[22] P. Patel-Schneider & R. Sebastiani (2003): *A New General Method to Generate Random Modal Formulae for Testing Decision Procedures*. *Journal of Artificial Intelligence Research* 18, pp. 351–389.

[23] M. Prasad, A. Biere & A. Gupta (2005): *A survey of recent advances in SAT-based formal verification*. *International Journal on Software Tools for Technology Transfer* 7, pp. 156–173.

[24] L. Ryan (2004): *Efficient Algorithms For Clause-Learning SAT Solvers*. Master's thesis, Simon Fraser University.