Relation-Changing Modal Operators

Carlos Areces[1,2], Raul Fervari[1,2] and Guillaume Hoffmann[1,2,3]

[1] FaMAF, Universidad Nacional de Córdoba
[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
[3] Universidad Blas Pascal
{areces, fervari, hoffmann}@famaf.unc.edu.ar

Abstract. We study dynamic modal operators that can change the accessibility relation of a model during the evaluation of a formula. In particular, we extend the basic modal language with modalities that are able to delete, add or swap an edge between pairs of elements of the domain. We define a generic framework to characterize this kind of operations. First, we investigate relation-changing modal logics as fragments of classical logics. Then, we use the new framework to get a suitable notion of bisimulation for the logics introduced, and we investigate their expressive power. Finally, we show that the complexity of the model checking problem for the particular operators introduced is PSpace-complete, and we study two subproblems of model checking: formula complexity and program complexity.

Keywords: modal logic, dynamic logics, expressivity, complexity.

1 Introduction

Modal logics [10,11] are particularly well suited to *describe* graphs, and this is fortunate as many situations can be modeled using graphs: an algebra, a database, the execution flow of a program or, simply, the arbitrary relations between a set of elements. This explains why modal logics have been used in many, diverse fields. They offer a well balanced trade-off between expressivity and computational complexity (the problem of model checking the basic modal language $\mathcal{ML}$ has only polynomial complexity, while the complexity of its satisfiability problem is PSpace-complete). Moreover, the range of modal logics known today is extremely wide, so that it is usually possible to pick and choose the right modal logic for a particular application.

But if we want to describe and reason about *dynamic aspects* of a given situation, e.g., how the relations between a set of elements *evolve* through time or through the application of certain operations, the use of modal logics (or actually, any kind of logic with classical semantics) becomes less clear. We can always resort to modeling the whole space of possible evolutions of the system as a graph, but this soon becomes unwieldy. It would be more elegant to use truly dynamic modal logics with operators that can mimic the changes that the structure will undergo. This is not a new idea, and an early example of this kind of logics is the *sabotage logic* introduced by Johan van Benthem in [40].

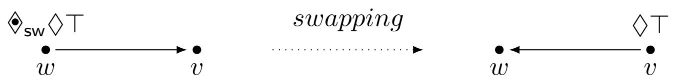
Consider the following *sabotage game*. It is played on a graph with two players, Runner and Blocker. Runner can move on the graph from node to accessible node, starting from a designated point, and with the goal of reaching a given final point. He should move one edge at a time. Blocker, on the other hand, can delete one edge from the graph, every time it is his turn. Of course, Runner wins if he manages to move from the origin to the final point in the graph, while Blocker wins otherwise. Van Benthem discusses in [40] how to transform the sabotage game into a modal logic. This original idea has been studied in several other works [28,34,21] where the semantics of the (global) sabotage operator $\blacklozenge_{\mathsf{gsb}}$ is defined as:

$$\mathcal{M}, w \models \blacklozenge_{\mathsf{gsb}}\varphi \quad \text{iff} \quad \text{there is a pair } (u,v) \text{ of } \mathcal{M} \text{ such that } \mathcal{M}^-_{(u,v)}, w \models \varphi,$$

where $\mathcal{M}^-_{(u,v)}$ is identical to $\mathcal{M}$ except that the edge (u,v) has been removed from the accessibility relation.

It is clear that the $\blacklozenge_{\mathsf{gsb}}$ operator *changes* the model in which a formula is evaluated. As Van Benthem puts it, $\blacklozenge_{\mathsf{gsb}}$ is an "external" modality that takes evaluation to another model, obtained from the current one by deleting some transition. It has been proved that solving the sabotage game is PSPACE-hard, while the model checking problem of the associated modal logic is PSPACE-complete and the satisfiability problem is undecidable. The logic fails to have both the finite model property and the tree model property [28,34].

In this article, we investigate various model changing operators. For example, $\blacklozenge_{\mathsf{sb}}$, for *local sabotage*, is a $\Diamond$ operator that destroys the traversed arrow. In contrast, the *bridge* operator $\blacklozenge_{\mathsf{br}}$ adds an arrow to an inaccessible state of the model and moves evaluation over there. We also consider $\blacklozenge_{\mathsf{sw}}$ which has the ability to swap the direction of a traversed arrow. The $\blacklozenge_{\mathsf{sw}}$ operator is a $\Diamond$ operator — to be true at a state w it requires the existence of an accessible state v where evaluation continues — but it changes the accessibility relation during evaluation: the pair (w,v) is deleted, and the pair (v,w) added to the accessibility relation (see [3] for details). A picture will help understand the dynamics of $\blacklozenge_{\mathsf{sw}}$. The formula $\blacklozenge_{\mathsf{sw}}\Diamond\top$ is true in a model with two related states:



As we can see in the picture, evaluation starts at state w with the arrow pointing from w to v, but after evaluating the $\blacklozenge_{\mathsf{sw}}$ operator, it continues at state v with the arrow now pointing from v to w.

More generally, let $\mathcal{M} = \langle W, R, V \rangle$ be a relational model and let f_W be a function that takes an element w of W and the current accessibility relation R over W and returns a set of pairs (v, S), where $v \in W$ is the new state of evaluation and S is the new accessibility relation to be used. In this article we focus on binary accessibility relations and, hence, $f_W : W \times 2^{W^2} \mapsto 2^{W \times 2^{W^2}}$, but of course the idea generalizes to modalities of arbitrary arity. In each model $\mathcal{M}$, each different function f_W defines a dynamic operator. For example, $\blacklozenge_{\mathsf{sw}}$ would be defined by the function $f_W(w, R) = \{(v, R\backslash\{(w, v)\} \cup \{(v, w)\}) \mid (w, v) \in R\}$.

Clearly, the modalities defined in this form do not cover all possible dynamic modal operators. For instance, dynamic modal operators investigated in different dynamic epistemic logics [47] change the set of states in the model, or the valuation function (see [4] for a more general, but complex framework). But, as we discussed, the present framework covers Van Benthem's original sabotage operator and other variants investigated in, e.g., [34].

In the next sections we, first, formally introduce the framework we just outlined; then, we investigate the logics obtained as fragments of classical logics, and we introduce the tools needed to investigate their expressive power. We also investigate the complexity of their model checking problem. We present both specific results for concrete operators, and general results that can be proved when the defining functions satisfy certain constraints. This article collects and extends results previously published in [2,3,16].

1.1 Related Work

There exists previous work that investigates operators which change a model during the evaluation of a formula, applied in different contexts. In the field of Belief Revision, the theories of belief change that have been developed are usually not presented as logics, in the proper sense, but rather as (more or less formal) axiomatic theories. The AGM approach [1], for example, is presented by means of a number of postulates in natural language that characterize the mathematical structures under study. In [13], the authors suggest representing belief change within the logical framework of a dynamic modal logic. This idea led to the development of Dynamic Doxastic Logic [36,37], which is an extension of traditional doxastic logic (see [23]) with dynamic operators representing various kinds of transformations of the agent's doxastic state. The main goal of basic dynamic doxastic logic is to describe an agent that has opinions about the external world and that can change these opinions in the light of new information.

Model changing operators have also been used in the field of Dynamic Epistemic Logics. One of the most used dynamic epistemic language is Action Model Logic (see [9]). This logic uses entities called *action models* as part of its syntax, which themselves use formulas of action model logic to define pre- and post-conditions. In this way, action models can be used to specify changes in the epistemic state of a group of agents. The epistemic models representing the knowledge of certain agents are updated according to the information represented by action models. In epistemic logic the knowledge of an agent is represented by the accessibility relation of the epistemic model. Epistemic updates correspond to the shrinking or expansion of each agent's accessibility to possible states of the world represented in the epistemic model.

Belief revision was investigated also in an epistemic setting, combining the works mentioned in the previous paragraphs. Belief revision and epistemic logics are two different approaches to information change, and the main idea is to take advantage of dynamic epistemic operators such as public announcements and action models to represent belief revision operations. Some complex forms of belief revision such as, for instance, iterated, revocable and higher-order revision can be formalized in this setting in a natural way (see, e.g., [7,45,44]).

In [8], some relation-changing operators are investigated as data structure modifiers. They can also be used to reason about changes in a graph. Two logics are introduced: one only involves global modifications (of some state label, or of some edge label) anywhere in the graph; the second allows for modifications that are local to states. The global version generalizes logics of public assignments (see, e.g., [12,46]) and public announcements (see, e.g., [32,25]), as well as logics of preference modification [42]. By means of reduction axioms they show that this logic is as expressive as the underlying logic without global modifiers. They also show that adding local modifiers dramatically increases the power of the logic turning the satisfiability problem undecidable.

In [26,27], arrow update logic is introduced as a theory of epistemic access elimination, that can be used to reason about multi-agent belief change. Arrow update logic generalizes the public announcement logic introduced in [20], in which a statement eliminates access to all epistemic possibilities in which the statement does not hold. It is inspired by the arrow pre-condition language proposed in [33], as well as other works about access elimination (see e.g., [12,8,40]). Arrow update logic is an extension of the basic epistemic logic with updates to eliminate edges according to certain conditions on their nodes. While the belief-changing updates of arrow logic can be transformed into equivalent updates with action models [9,47], arrow updates are sometimes exponentially more succinct than action models. The main difference between arrow updates and the sabotage operator à la Van Benthem is that arrow updates remove edges according to a pre and a post-condition, and sabotage removes arbitrary edges in the model.

The different lines of work we mentioned in this section are examples of the use of relation-changing modal logics. In this article, we investigate a general framework which encompasses a wide family of relation-changing modal logics. One of the main differences of the new logics investigated, with respect to the ones previously mentioned, is their high expressive power. For example, many dynamic epistemic logics (see, e.g., [43,42,41,24]) have *reduction axioms* into basic modal logic (i.e., each formula can be rewritten to an equivalent formula in the basic modal logic). Instead, we will show that the general framework we define includes logics which are strictly more expressive that the basic modal logic (actually, it has been previously shown that in some cases, their satisfiability problem is undecidable, see [28,3,16]).

2 Basic Definitions

The syntax of the dynamic modal logics we study is a straightforward extension of the basic modal logic (see [10]):

Definition 1 (Syntax). *Let* PROP *be a countable, infinite set of propositional symbols. The set* FORM *of formulas over* PROP *is defined as:*

$$\mathsf{FORM} ::= \bot \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond\varphi, \mid \blacklozenge_i\varphi,$$

where $p \in \mathsf{PROP}$, $\blacklozenge_i \in \mathsf{DYN}$ *a set of dynamic operators, and* $\varphi, \psi \in \mathsf{FORM}$. *Other operators are defined as usual. In particular,* $\Box\varphi$ *is defined as* $\neg\Diamond\neg\varphi$ *and* $\boxdot_i\varphi$ *is defined as* $\neg\blacklozenge_i\neg\varphi$.

Formulas of the basic modal language $\mathcal{ML}$ *are those that contains only the* $\Diamond$ *operator besides the Boolean operators. For* $S \subseteq \mathsf{DYN}$ *a set of dynamic operators, we call* $\mathcal{ML}(S)$ *the extension of* $\mathcal{ML}$ *allowing also the operators in* S*. If* S *is a singleton set* $S = \{\blacklozenge\}$*, we write* $\mathcal{ML}(\blacklozenge)$ *instead of* $\mathcal{ML}(\{\blacklozenge\})$.

In the rest of the article we refer to the modal depth (md) and the dynamic modal depth (dmd) of a formula. These two measures are defined as follows:

$$\begin{aligned} \mathsf{md}(p) &= 0 & \mathsf{dmd}(p) &= 0 \\ \mathsf{md}(\neg\varphi) &= \mathsf{md}(\varphi) & \mathsf{dmd}(\neg\varphi) &= \mathsf{dmd}(\varphi) \\ \mathsf{md}(\varphi \wedge \psi) &= \max\{\mathsf{md}(\varphi), \mathsf{md}(\psi)\} & \mathsf{dmd}(\varphi \wedge \psi) &= \max\{\mathsf{dmd}(\varphi), \mathsf{dmd}(\psi)\} \\ \mathsf{md}(\Diamond) &= 1 + \mathsf{md}(\varphi) & \mathsf{dmd}(\Diamond\varphi) &= \mathsf{dmd}(\varphi) \\ \mathsf{md}(\blacklozenge_f) &= 1 + \mathsf{md}(\varphi) & \mathsf{dmd}(\blacklozenge_f\varphi) &= 1 + \mathsf{dmd}(\varphi). \end{aligned}$$

Semantically, formulas of $\mathcal{ML}(S)$ are evaluated in standard relational models, and the meaning of all the operators of the basic modal logic is unchanged.

Definition 2 (Models). *A model* $\mathcal{M}$ *is a triple* $\mathcal{M} = \langle W, R, V\rangle$*, where* W *is the* domain*, a non-empty set whose elements are called points or states;* $R \subseteq W \times W$ *is the* accessibility relation*; and* $V : \mathsf{PROP} \mapsto 2^W$ *is the* valuation*. For* $\mathcal{M}$ *a model, we usually write* $|\mathcal{M}|$ *for its domain.*

Let w *be a state in* $\mathcal{M}$*, the pair* $(\mathcal{M}, w)$ *is called a pointed model; we usually drop parentheses and call* $\mathcal{M}, w$ *a pointed model.*

In this article, we restrict ourselves to models with only one accessibility relation (i.e., the underlying modal language has only one modal operator). A generalization to models with multiple accessibility relations is possible, but leads to further choices concerning the definition of the dynamic operators (e.g., which relation is affected by a given dynamic operator).

Definition 3 (Model update functions). *Given a domain* W*, a* model update function for W *is a function* $f_W : W \times 2^{W^2} \to 2^{W \times 2^{W^2}}$*, that takes a state in* W *and a binary relation over* W *and returns a set of possible updates to the state of evaluation and accessibility relation.*

Let $\mathcal{C}$ *be a class of models, a* family of model update functions f *is a class of model update functions, one for each domain of a model in* $\mathcal{C}$*:*

$$f = \{f_W \mid \langle W, R, V\rangle \in \mathcal{C}\}.$$

$\mathcal{C}$ *is* closed under a family of model update functions f *if whenever* $\mathcal{M} = \langle W, R, V\rangle \in \mathcal{C}$*, then* $\{\langle W, R', V\rangle \mid f_W \in f, w \in W, (v, R') \in f_W(w, R)\} \subseteq \mathcal{C}$.

Clearly, the class of all pointed models is closed under any family of model update functions. In the rest of the article we only discuss the class of all models.

Notice, in the definition above, that a model update function is defined relative to a domain. We specifically require that all models with the same domain have the same model update function. This constraint limits the number of operators that can be captured in the framework, but at the same time leads to operators with a more uniform behavior. We will discuss this issue further after we introduce the formal semantics of the relation-changing operators below.

We now introduce the semantics for the general case.

Definition 4 (Semantics). *Let $\mathcal{C}$ be a class of models, $\mathcal{M} = \langle W, R, V \rangle$ be a model in $\mathcal{C}$, $w \in W$ a state, f a family of model update functions for $\mathcal{C}$ and $\blacklozenge_f$ its associated dynamic operator. Let φ be a formula in $\mathcal{ML}(\blacklozenge_f)$. We say that $\mathcal{M}, w$* satisfies *φ, and write $\mathcal{M}, w \models \varphi$, when*

$$\begin{array}{ll}
\mathcal{M}, w \models p & \textit{iff } w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \textit{iff } \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \textit{iff } \mathcal{M}, w \models \varphi \textit{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Diamond\varphi & \textit{iff for some } v \in W \textit{ s.t. } (w, v) \in R,\ \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models \blacklozenge_f\varphi & \textit{iff for some } (v, R') \in f_W(w, R),\ \langle W, R', V \rangle, v \models \varphi.
\end{array}$$

The definition extends to languages with many modal dynamic operators in the obvious manner. φ is satisfiable if for some pointed model $\mathcal{M}, w$ we have $\mathcal{M}, w \models \varphi$. We write $\mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$ when both models satisfy the same $\mathcal{L}$-formulas, i.e., for all $\varphi \in \mathcal{L}$, $\mathcal{M}, w \models \varphi$ if and only if $\mathcal{N}, v \models \varphi$. We drop the $_{\mathcal{L}}$ subindex when no confusion arises.

Notice, in the semantic definition, how the relation-changing modal operator $\blacklozenge_f$ potentially changes both the state of evaluation and the accessibility relation. On the other hand, the model domain remains the same, and hence all $\blacklozenge_f$ operators in a formula are evaluated using the same model update function.

Consider the following model update functions. To simplify notation we use wv as a shorthand for $\{(w, v)\}$ or (w, v); context will always disambiguate the intended use. Given a binary relation R define the following notation:

$$\begin{array}{l}
R^-_{wv} = R \backslash wv \\
R^+_{wv} = R \cup wv \\
R^*_{wv} = (R \backslash vw) \cup wv.
\end{array}$$

Define now the following six model update functions, which give rise to natural dynamic modal operators: Van Benthem's sabotage operator $\blacklozenge_{\mathsf{gsb}}$, and a local version $\blacklozenge_{\mathsf{sb}}$ that deletes an existing edge between the current state of evaluation and a successor state; a "bridge" operator $\blacklozenge_{\mathsf{gbr}}$ that adds an edge between two previously unconnected states, and a local version $\blacklozenge_{\mathsf{br}}$ that links the current state of evaluation and an inaccessible state; and the global and local versions ($\blacklozenge_{\mathsf{gsw}}$ and $\blacklozenge_{\mathsf{sw}}$, respectively) of the swap operator we discussed above. Let W be a domain and R a binary relation over W,

$$
\begin{aligned}
f_W^{\mathsf{sb}}(w,R) &= \{(v, R^-_{wv}) \mid wv \in R\} & \qquad f_W^{\mathsf{gsb}}(w,R) &= \{(w, R^-_{uv}) \mid uv \in R\}\\
f_W^{\mathsf{br}}(w,R) &= \{(v, R^+_{wv}) \mid wv \notin R\} & \qquad f_W^{\mathsf{gbr}}(w,R) &= \{(w, R^+_{uv}) \mid uv \notin R\}\\
f_W^{\mathsf{sw}}(w,R) &= \{(v, R^*_{vw}) \mid wv \in R\} & \qquad f_W^{\mathsf{gsw}}(w,R) &= \{(w, R^*_{vu}) \mid uv \in R\}.
\end{aligned}
$$

In the next sections we investigate dynamic logics that can be defined in the framework we introduced, with particular focus on the six concrete operators $\blacklozenge_{\mathsf{sb}}$, $\blacklozenge_{\mathsf{gsb}}$, $\blacklozenge_{\mathsf{br}}$, $\blacklozenge_{\mathsf{gbr}}$, $\blacklozenge_{\mathsf{sw}}$ and $\blacklozenge_{\mathsf{gsw}}$ associated to the six families of model update functions we just introduced. As mentioned, $\blacklozenge_{\mathsf{gsb}}$ in Van Benthem's sabotage operator. The other operators also have natural properties. For example, local sabotage and local swap are logically stronger than the diamond operator when restricted to non-dynamic predicates, as the formulas $\blacklozenge_{\mathsf{sb}}p \to \Diamond p$ and $\blacklozenge_{\mathsf{sw}}p \to \Diamond p$ are valid. The operators are very expressive and, as we discuss in Section 4, they can force non-tree models. For example, the formula $\boxdot_{\mathsf{sb}}\Box\bot$ means that any local sabotage leads to a dead-end, hence the formula $\Diamond\Diamond\top \wedge \boxdot_{\mathsf{gsb}}\Box\bot$ can only be true at a reflexive state, a property that cannot be expressed in the basic modal language.

3 Translations

In this section we discuss relation-changing modal logics as fragments of better known logics. We start by defining a generic translation from any logic $\mathcal{ML}(\blacklozenge_f)$ into second-order logic, where $\blacklozenge_f$ is defined by a family of model update functions f. We then show that in some cases a translation into first-order logic is possible. Finally, we discuss how relation-changing modal logics can be seen as multi-modal logics over particular classes of models.

3.1 The Standard Translation

It is a well known result that the basic modal logic $\mathcal{ML}$ can be translated into first-order logic using, for example, the following (standard) translation ST.

Definition 5. *The* correspondence language *for the basic modal language $\mathcal{ML}$ is a relational language with a unary relation symbol p for each propositional symbol p and a binary relation symbol r for the modality $\Diamond$.*

Let ST be the following function that translates formulas from $\mathcal{ML}$ into its correspondence language:

$$
\begin{aligned}
\mathsf{ST}_x(p) &= p(x)\\
\mathsf{ST}_x(\neg\varphi) &= \neg\mathsf{ST}_x(\varphi)\\
\mathsf{ST}_x(\varphi \wedge \psi) &= \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi)\\
\mathsf{ST}_x(\Diamond\varphi) &= \exists y.(r(x,y) \wedge \mathsf{ST}_y(\varphi)),
\end{aligned}
$$

where y is a variable which has not been used yet in the translation.

ST mimics the conditions for the satisfiability of a formula in a model and the resulting first-order formula is equivalent to the original modal formula [10].

Proposition 1. *Let* $\varphi \in \mathcal{ML}$ *then* $\mathcal{M}, w \models \varphi$ *if and only if* $\mathcal{M}, g^x_w \models \mathsf{ST}_x(\varphi)$, *where* g *is an arbitrary first-order assignment and* g^x_w *is identical to* g *except perhaps in that* $g^x_w(x) = w$.

Notice, in the theorem above, that $\mathcal{M} = \langle W, R, V\rangle$ can be used to interpret $\mathsf{ST}_x(\varphi)$, with the accessibility relation R interpreting the relation symbol r and $V(p)$ interpreting p. The translation ST can be extended to relation-changing modal operators when the family of model update functions can be defined in the language and we allow second-order quantification. The intuition is that the second-order quantifier, with the help of the formula defining the family of model update functions, can redefine the accessibility relation that should be used when translating a dynamic operator.

Definition 6. *Let* f *be a family of model update functions, and let* $\delta_f(v_1, V_1, v_2, V_2)$ *be a formula over the appropriate correspondence language with only the first-order variables* v_1, v_2 *and the second-order binary variables* V_1,V_2 *free. We say that* δ_f defines f *if in every model* $\mathcal{M} = \langle W, R, V\rangle$, *for every* $w \in W$, *and for every second-order assignment* g,

$$\textit{for all } v,\ S,\ (v, S) \in f_W(w, R) \textit{ iff } \mathcal{M}, (((g^{v_1}_w)^{V_1}_R)^{v_2}_v)^{V_2}_S \models \delta_f.$$

Given a family of model update functions f, *and* δ_f *a formula that defines* f, *define* $\mathsf{ST}_{x,r}$ *as follows*

$$\begin{aligned}
\mathsf{ST}_{x,r}(p) &= p(x)\\
\mathsf{ST}_{x,r}(\neg\varphi) &= \neg\mathsf{ST}_{x,r}(\varphi)\\
\mathsf{ST}_{x,r}(\varphi \wedge \psi) &= \mathsf{ST}_{x,r}(\varphi) \wedge \mathsf{ST}_{x,r}(\psi)\\
\mathsf{ST}_{x,r}(\Diamond\varphi) &= \exists y.(r(x, y) \wedge \mathsf{ST}_{y,r}(\varphi))\\
\mathsf{ST}_{x,r}(\blacklozenge_f\varphi) &= \exists y.\exists s.(\delta_f[{}^{v_1}/_{x}, {}^{V_1}/_{r}, {}^{v_2}/_{y}, {}^{V_2}/_{s}] \wedge \mathsf{ST}_{y,s}(\varphi)),
\end{aligned}$$

where $\theta[{}^{x}/_{y}]$ *is the formula obtained by replacing all free occurrences of* x *by* y *in* θ, *and* y, s *are variables which have not been used yet in the translation.*

It is not difficult to prove by induction that these translation functions preserve the meaning of formulas in a model.

Proposition 2. *Let* $\varphi \in \mathcal{ML}(\blacklozenge_f)$ *and let* δ_f *be a formula defining* f. *Then* $\mathcal{M}, w \models \varphi$ *if and only if* $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,r}(\varphi)$, *where* g *is an arbitrary second-order assignment and* g^x_w *is identical to* g *except perhaps in that* $g^x_w(x) = w$.

The function ST translates $\mathcal{ML}(\blacklozenge_f)$-formulas into second-order formulas. It has been proved in [3,16] that $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ is a proper fragment of first-order logic, but the translation we give in this section provides a more general framework.

It is easy to define the formula $\varphi_\Diamond$ which characterizes the operator $\Diamond$:

$$\delta_\Diamond \doteq V_1(v_1, v_2) \wedge \forall z.\forall z'.(V_1(z, z') \leftrightarrow V_2(z, z')).$$

The formula above clearly establishes that the current state has a successor, and that the accessibility relation does not change. A formula characterizing $\Diamond_{\mathsf{sb}}$ is:

$$\delta_{\Diamond_{\mathsf{sb}}} \doteq V_1(v_1,v_2)\wedge\neg V_2(v_1,v_2)\wedge\forall z.\forall z'.((v_1,v_2)\neq(z,z')\rightarrow(V_1(z,z')\leftrightarrow V_2(z,z'))).$$

For $\Diamond_{\mathsf{gsb}}$, we need to specify that the update is in any part of the model, and the evaluation state does not change:

$$\delta_{\Diamond_{\mathsf{gsb}}} \doteq (v_1=v_2)\wedge\exists z.\exists z'.(V_1(z,z')\wedge\neg V_2(z,z')\wedge \forall w.\forall w'.((z,z')\neq(w,w')\rightarrow(V_1(w,w')\leftrightarrow V_2(w,w')))).$$

The formulas for the bridge operator are similar.

$$\delta_{\Diamond_{\mathsf{br}}} \doteq \neg V_1(v_1,v_2)\wedge V_2(v_1,v_2)\wedge\forall z.\forall z'.((v_1,v_2)\neq(z,z')\rightarrow(V_1(z,z')\leftrightarrow V_2(z,z'))).$$
$$\delta_{\Diamond_{\mathsf{gbr}}} \doteq (v_1=v_2)\wedge\exists z.\exists z'.(\neg V_1(z,z')\wedge V_2(z,z')\wedge \forall w.\forall w'.((z,z')\neq(w,w')\rightarrow(V_1(w,w')\leftrightarrow V_2(w,w')))).$$

The formulas for $\Diamond_{\mathsf{sw}}$ and $\Diamond_{\mathsf{gsw}}$ are only slightly more involved. In all cases, the resulting function translates formulas into second-order logic. More interestingly, for these particular six concrete relation-changing operators it is possible to define translations into first-order logic as we show in the next section.

3.2 Explicit Translations to First-Order Logic

Let us first observe that, in general, not all relation-changing modal operators can be translated into first-order logic. For instance consider the operator $\bigcirc^+$ with the following semantics: $\langle W,R,V\rangle,w\models\bigcirc^+\varphi$ iff $\langle W,R^+,V\rangle,w\models\varphi$. Its intuitive semantics is that φ is evaluated after replacing the current accessibility relation by its transitive closure. As stated by the following proposition, $\mathcal{ML}(\bigcirc^+)$ is not compact and hence it cannot be translated into first-order logic.

Proposition 3. *$\mathcal{ML}(\bigcirc^+)$ is not compact.*

Proof. The argument is similar to the one used for Propositional Dynamic Logic (see [11] for details). Consider the infinite set $\Gamma=\{\bigcirc^+\Diamond p\}\cup\{\Box^n\neg p\mid n\geq 0\}$. Every finite subset of Γ is satisfiable, but Γ is not.

We present now translations from $\mathcal{ML}(\Diamond)$, $\Diamond\in\{\Diamond_{\mathsf{sb}},\Diamond_{\mathsf{br}},\Diamond_{\mathsf{sw}},\Diamond_{\mathsf{gsb}},\Diamond_{\mathsf{gbr}},\Diamond_{\mathsf{gsw}}\}$ to first-order logic, inspired by the translation of $\mathcal{ML}(\Diamond_{\mathsf{gsb}})$ presented in [34].

Let VAR be a totally ordered set of first-order variables. We consider a set $S\subseteq\mathsf{VAR}\times\mathsf{VAR}$ to be interpreted as the set of modified edges in the model, according to the logic we are translating. We write xy for (x,y), and use the following notation:

$nm=xy$	is defined as	$n=x\wedge m=y$
$nm\neq xy$	is defined as	$n\neq x\ \vee\ m\neq y$
$nm\in S$	is defined as	$\bigvee_{xy\in S} nm=xy$, and
$nm\notin S$	is defined as	$\bigwedge_{xy\in S} nm\neq xy$,

where S is a finite set of pairs of variables. In particular $nm \in \emptyset$ is a notation for $\bot$ and $nm \notin \emptyset$ is a notation for $\top$. For S a set of pairs of variables, define $S^{-1} = \{mn \mid nm \in S\}$.

We present the non-trivial cases of the translation for $\mathcal{ML}(\{\blacklozenge_{\mathsf{sb}}, \blacklozenge_{\mathsf{gsb}}\})$ (which shares the case for the basic modality $\Diamond$):

$$\begin{aligned}
\mathsf{ST}_{x,S}(\Diamond\varphi) &= \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S}(\varphi)) \\
\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{sb}}\varphi) &= \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S\cup xy}(\varphi)) \\
\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{gsb}}\varphi) &= \exists y.\exists z.(r(y,z) \wedge yz \notin S \wedge \mathsf{ST}_{x,S\cup yz}(\varphi)),
\end{aligned}$$

where y and z are variables which have not been used yet in the translation.

Notice that S is not a relational symbol but a set of pairs of variables, that refer to deleted edges in the model. It does not appear in the final formula and is used only during the translation.

Proposition 4. *Given φ a formula of $\mathcal{ML}(\blacklozenge_{\mathsf{sb}})$ or $\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$ and $\mathcal{M}, w$ a pointed model, we have $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,\emptyset}(\varphi)$, where g is an arbitrary first-order assignment and g^x_w is identical to g except perhaps in that $g(x) = w$.*

Proof. Let $\mathcal{M} = \langle W, R, V\rangle$ be a model, $X \subseteq W \times W$, $S \subseteq \mathsf{VAR}\times\mathsf{VAR}$ and $g : \mathsf{VAR} \mapsto W$ be an assignment. We say that the tuple $\langle g, S, X\rangle$ is *adequate* if the mapping $(x, y) \mapsto (g(x), g(y))$ is bijective from S to X. We write $\mathcal{M}^-_X$ for $\langle W, R\backslash X, V\rangle$.

By structural induction on the formula, we show that for all φ, $\mathcal{M} = \langle W, R, V\rangle$, $w \in W$, $X \subseteq W\times W$, $S \subseteq \mathsf{VAR}\times\mathsf{VAR}$ and an assignment $g : \mathsf{VAR} \mapsto W$ such that $\langle g, S, X\rangle$ is adequate:

$$\mathcal{M}^-_X, w \models \varphi \text{ iff } \mathcal{M}, g^x_w \models \mathsf{ST}_{x,S}(\varphi).$$

First, the case for p is trivial, as are $\neg\varphi$, and $\varphi \wedge \psi$ by inductive hypothesis.

Consider the case $\Diamond\psi$. We need to show that for all $\mathcal{M}$, w, and all adequate $\langle g, S, X\rangle$, $\mathcal{M}^-_X, w \models \Diamond\psi$ iff $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,S}(\Diamond\psi)$. Assume the left part of the equivalence, i.e., $\mathcal{M}^-_X, w \models \Diamond\psi$. That is, there exists v such that $(w, v) \in (R\backslash X)$ and $\mathcal{M}^-_X, v \models \psi$. Then by inductive hypothesis, $\mathcal{M}, g^y_v \models \mathsf{ST}_{y,S}(\psi)$, with adequate $\langle g, S, X\rangle$.

We can assume without loss of generality that there exists some $x \in \mathsf{VAR}$ such that $g(x) = w$. This is because either it exists, or we can use some arbitrary x that does not appear in $\mathsf{ST}_{y,S}(\Diamond\psi)$ and modify g such that $g(x) = w$. In any case, since $(w, v) \in (R\backslash X)$, this implies $\mathcal{M}, g^y_v \models r(x, y) \wedge xy \notin S$. This is equivalent to $\mathcal{M}, (g^x_w)^y_v \models \mathsf{ST}_{x,S}(\Diamond\psi)$, which is what we wanted.

For the right-to-left direction, assume $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,S}(\Diamond\psi)$ with some assignment g, and some set of pairs S and X such that $\langle g, S, X\rangle$ is adequate. We have $\mathcal{M}, g^x_w \models \exists y.(r(x, y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S}(\psi))$. This implies $\mathcal{M}, (g^x_w)^y_v \models r(x, y)$, $\mathcal{M}, (g^x_w)^y_v \models xy \notin S$ and $\mathcal{M}, (g^x_w)^y_v \models \mathsf{ST}_{y,S}(\psi)$. The first two formulas imply that $(w, v) \in (R\backslash X)$, and by inductive hypothesis, the third one implies that $M^-_X, v \models \psi$, thus $M^-_X, w \models \Diamond\psi$.

Now consider the case where $\varphi = \blacklozenge_{\mathsf{sb}}\psi$. We need to show that for all $\mathcal{M}$, w, and adequate $\langle g, S, X\rangle$, $\mathcal{M}^-_X, w \models \blacklozenge_{\mathsf{sb}}\psi$ iff $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{sb}}\psi)$.

Assume $\mathcal{M}^-_X, w \models \blacklozenge_{\mathsf{sb}}\psi$. There exists v such that $(w, v) \in (R\backslash X)$ and $\mathcal{M}^-_{X\cup wv}, v \models \psi$. Then by inductive hypothesis, $\mathcal{M}, g^y_v \models \mathsf{ST}_{y,T}(\psi)$, with adequate $\langle g, T, X\cup wv\rangle$. Again, we can assume there exists $x \in \mathsf{VAR}$ such that $g(x) = w$ without loss of generality. Let us call S the set such that $T = S\cup xy$.

Since $\langle g, S\cup xy, X\cup wv\rangle$ is adequate and g maps xy to wv, then $\langle g, S, X\rangle$ is also adequate. Thus we have, given that $(w, v) \in (R\backslash X)$, $\mathcal{M}, g^y_v \models r(x, y) \wedge xy \notin S$. With the previous formulas this entails $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{sb}}\psi)$.

For the right-to-left direction, assume $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{sb}}\psi)$ and let X a set such that $\langle g, S, X\rangle$ is adequate. We have $\mathcal{M}, g^x_w \models \exists y.(r(x, y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S\cup xy}(\psi))$. Since y is a new variable, we can assume without loss of generality that $g(y) = v$, thus: $\mathcal{M}, g^x_w \models r(x, y)$, $\mathcal{M}, g^x_w \models xy \notin S$, and $\mathcal{M}, g^x_w \models \mathsf{ST}_{y,S\cup xy}(\psi))$. The first two formulas imply that $(w, v) \in (R\backslash X)$. From the last formula, and by inductive hypothesis, we have: $\mathcal{M}^-_{X\cup g^x_w(x)g^x_w(y)}, g(y) \models \psi$, i.e., $\mathcal{M}^-_{X\cup wv}, v \models \psi$. This implies that $\mathcal{M}^-_X, w \models \blacklozenge_{\mathsf{sb}}\psi$.

The case for $\blacklozenge_{\mathsf{gsb}}\psi$ can be proved similarly. □

The translations for the bridge and swap operations also use an intermediate set of variable pairs to represent modifications in the model. For $\mathcal{ML}(\{\blacklozenge_{\mathsf{br}}, \blacklozenge_{\mathsf{gbr}}\})$, the set S represents the edges added to the model. We define:

$$\begin{aligned}
\mathsf{ST}_{x,S}(\Diamond\varphi) &= \exists y.((r(x, y) \vee xy \in S) \wedge \mathsf{ST}_{y,S}(\varphi))\\
\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{br}}\varphi) &= \exists y.(\neg(r(x, y) \vee xy \in S) \wedge \mathsf{ST}_{y,S\cup xy}(\varphi))\\
\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{gbr}}\varphi) &= \exists y.\exists z.(\neg(r(y, z) \vee yz \in S) \wedge \mathsf{ST}_{x,S\cup yz}(\varphi)).
\end{aligned}$$

For $\mathcal{ML}(\{\blacklozenge_{\mathsf{sw}}, \blacklozenge_{\mathsf{gsw}}\})$, S refers to the edges swapped in the model. We define:

$$\begin{aligned}
\mathsf{ST}_{x,S}(\Diamond\varphi) &= \exists y.(((r(x, y) \wedge xy \notin S) \vee xy \in S^{-1}) \wedge \mathsf{ST}_{y,S}(\varphi))\\
\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{sw}}\varphi) &= \quad (r(x, x) \wedge \mathsf{ST}_{x,S}(\varphi))\\
&\quad \vee\ \exists y.(x \neq y \wedge r(x, y) \wedge xy \notin (S\cup S^{-1}) \wedge \mathsf{ST}_{y,S\cup xy}(\varphi))\\
&\quad \vee\ \textstyle\bigvee_{yz\in S}(x = z \wedge \mathsf{ST}_{y,S\backslash yz\cup zy}(\varphi))\\
\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{gsw}}\varphi) &= \quad (\exists y.r(y, y) \wedge \mathsf{ST}_{x,S}(\varphi))\\
&\quad \vee\ \exists y.\exists z.(y \neq z \wedge r(y, z) \wedge yz \notin (S\cup S^{-1}) \wedge \mathsf{ST}_{x,S\cup yz}(\varphi))\\
&\quad \vee\ \textstyle\bigvee_{yz\in S} \mathsf{ST}_{x,S\backslash yz\cup zy}(\varphi).
\end{aligned}$$

We show that these translations also preserve equivalence.

Proposition 5. *Given $\mathcal{M}, w$ some pointed model and φ a formula of $\mathcal{ML}(\blacklozenge_{\mathsf{br}})$, $\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$, $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ or $\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$, we have $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, g^x_w \models \mathsf{ST}_{x,\emptyset}(\varphi)$, where g is an arbitrary first-order assignment and g^x_w is identical to g except perhaps in that $g(x) = w$.*

Proof. We use an inductive hypothesis similar to the one in the previous proof. For $\mathcal{ML}(\blacklozenge_{\mathsf{br}})$ and $\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$, the difference is that the standard translation maintains S as a set of variables referring to *added* edges in the model.

For $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ and $\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$, the translation is more complicated. It maintains the following invariant: the set S is irreflexive, and when an edge xy belongs to S, it means that xy is removed from the relation of the original model, and yx is added. That is, xy is no longer in the model because it has been swapped.

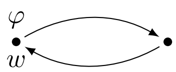
$\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{sw}}\varphi)$ is the disjunction of three cases. Either the swap operation occurs on a reflexive edge, in which case the set S is not modified (since swapping a reflexive edge leaves it unchanged). Or, swapping occurs at some edge xy, such that neither xy is in S, nor its inverse yx. In this case swapping is remembered simply by adding xy to the set S. We require yx not to be in S to avoid naming twice the same edge in the translation. This makes re-swapping (the next case) doable. Finally, swapping occurs by traversing some edge zy present in S^{-1}. That is, we swap again a swapped edge. In this case we remove yz from S, add zy, and continue the translation standing at the state stored in y.

$\mathsf{ST}_{x,S}(\blacklozenge_{\mathsf{gsw}}\varphi)$ is a generalization of the previous case for swapping occurring anywhere in the model. □

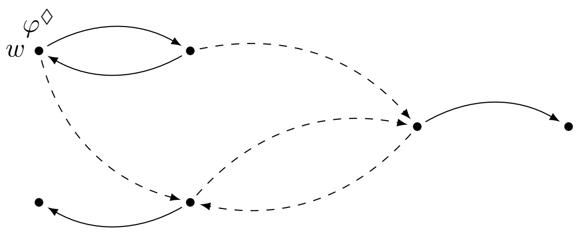
3.3 Unfolding

We now show a different kind of translation. We no longer translate only formulas, but also the models in which they are to be evaluated. Moreover, we do not translate to first- or second-order logic, but to the basic modal logic, albeit with two modalities. The key idea is the use of *model unfolding*.

Suppose we start with a model $\mathcal{M}$ and a formula φ of a dynamic modal logic like $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$:



Starting from $\mathcal{M}$, we can explicitly build the variants obtained by successive applications of the dynamic operator $\blacklozenge_{\mathsf{sw}}$:



This unfolded model has two accessibility relations: one represents the relation in each update model, the another is an "external" relation that links updated model. At the syntactic level, we can rewrite φ into a modal formula $\varphi^{\Diamond}$ in the basic modal language $\mathcal{ML}$ with two classic modalities interpreted using the relations in the unfolded model. In general, we can always define the following translation:

Definition 7 (Bi-modal translation). *Consider the language $\mathcal{ML}(\blacklozenge_f)$ for a family of model update functions f. Define the following translation from this* *language to the basic modal language with two modalities:*

$$\begin{aligned}(p)^\Diamond &= p\\ (\varphi \wedge \psi)^\Diamond &= \varphi^\Diamond \wedge \psi^\Diamond\\ (\neg\varphi)^\Diamond &= \neg\varphi^\Diamond\\ (\Diamond\varphi)^\Diamond &= \Diamond_1\varphi^\Diamond\\ (\blacklozenge_f\varphi)^\Diamond &= \Diamond_2\varphi^\Diamond.\end{aligned}$$

Define relation updates as follows:

Definition 8 (Relation updates). *Let W be a domain, R a relation on W, f_W a model update function for W, and n a natural number. Define $R_{f_W,n}$, the set of all possible* relation updates *obtained applying n times the function f_W on the relation R as:*

$$\begin{aligned}R_{f_W,0} &= \{R\}\\ R_{f_W,n+1} &= R_{f_W,n} \cup \{T \mid (v,T) \in f_W(w,S), S \in R_{f_W,n}, w \in W\}.\end{aligned}$$

Define the set of all relation variants obtained applying f_W on the relation R as

$$R_{f_W} = \bigcup_{n<\omega} R_{f_W,n}.$$

We can now define the unfolding of a model.

Definition 9 (Model unfolding). *Let $\mathcal{M} = \langle W, R, V\rangle$ and let f be a family of model update functions. Define $\mathcal{M}_{f,n} = \langle W', \{R'_1, R'_2\}, V'\rangle$ the* n-bounded unfolding *of $\mathcal{M}$ as follows. For $n = 0$, let*

$$\begin{aligned}W' &= W \times \{R\}\\ R'_1 &= \{((s,R),(t,R)) \mid (s,t) \in R\}\\ R'_2 &= \emptyset\\ V'(p) &= \{(s,R) \mid s \in V(p)\}.\end{aligned}$$

While for $n+1$, let

$$\begin{aligned}W' &= W \times R_{f_W,n+1}\\ R'_1 &= \{((s,S),(t,S)) \mid (s,t) \in S, S \in R_{f_W,n+1}\}\\ R'_2 &= \{((s,S),(t,T)) \mid (t,T) \in f(s,S), S \in R_{f_W,n}\}\\ V'(p) &= \{(s,S) \mid s \in V(p), S \in R_{f_W}\}.\end{aligned}$$

The unbounded unfolding *of a model is $\mathcal{M}_f = \langle W', \{R'_1, R'_2\}, V'\rangle$, where*

$$\begin{aligned}W' &= W \times R_{f_W}\\ R'_1 &= \{((s,S),(t,S)) \mid (s,t) \in S, S \in R_f\}\\ R'_2 &= \{((s,S),(t,T)) \mid (t,T) \in f_W(s,S), S \in R_{f_W}\}\\ V'(p) &= \{(s,S) \mid s \in V(p), S \in R_{f_W}\}.\end{aligned}$$

We can prove the equivalence between satisfiability of a dynamic formula in a model, and satisfiability of its bi-modal translation in the bounded unfolding:

Proposition 6. *Let* $\mathcal{M} = \langle W, R, V \rangle$*, and* $\varphi \in \mathcal{ML}(\blacklozenge_f)$*. Then* $\mathcal{M}, w \models \varphi$ *iff* $\mathcal{M}_{f,\mathsf{dmd}(\varphi)}, (w, R) \models \varphi^{\Diamond}$*.*

Proof. By induction on the complexity of φ. $\square$

This translation reflects the idea that relation-changing operators can be seen as external modalities, that move evaluation of a formula to a different model.

4 Bisimulations and Expressivity

In modal model theory, the notion of bisimulation is a crucial tool. Typically, a bisimulation is a binary relation linking elements of the domains that have the same atomic information, and preserving the relational structure of the model. Because we need to keep track of the changes on the accessibility relation that the dynamic operators can introduce, we will define bisimulations as relations that link pairs of a state together with the current accessibility relation.

Definition 10 (Bisimulations). *Let* $\mathcal{M} = \langle W, R, V \rangle$*,* $\mathcal{M}' = \langle W', R', V' \rangle$ *be two models, and* f *a family of model update functions. A non empty relation* $Z \subseteq (W \times 2^{W^2}) \times (W' \times 2^{W'^2})$ *is an* $\mathcal{ML}(\blacklozenge_f)$*-bisimulation if it satisfies the following conditions. If* $(w, S)Z(w', S')$ *then*

(atomic harmony) *for all* $p \in \mathsf{PROP}$*,* $w \in V(p)$ *iff* $w' \in V'(p)$*;*
(zig) *if* $(w, v) \in S$*, there is* $v' \in W'$ *s.t.* $(w', v') \in S'$ *and* $(v, S)Z(v', S')$*;*
(zag) *if* $(w', v') \in S'$*, there is* $v \in W$ *s.t.* $(w, v) \in S$ *and* $(v, S)Z(v', S')$*;*
(f-zig) *if* $(v, T) \in f_W(w, S)$*, there is* $(v', T') \in f_{W'}(w', S')$ *s.t.* $(v, T)Z(v', T')$*;*
(f-zag) *if* $(v', T') \in f_{W'}(w', S')$*, there is* $(v, T) \in f_W(w, S)$ *s.t.* $(v, T)Z(v', T')$*.*

Given two pointed models $\mathcal{M}, w$ *and* $\mathcal{M}', w'$ *they are* $\mathcal{ML}(\blacklozenge_f)$*-bisimilar (notation,* $\mathcal{M}, w \leftrightarroweq_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$*) if there is an* $\mathcal{ML}(\blacklozenge_f)$*-bisimulation* Z *such that* $(w, R)Z(w', R')$ *where* R *and* R' *are respectively the relations of* $\mathcal{M}$ *and* $\mathcal{M}'$*.*

For instance, according to the above definition, besides (atomic harmony), (zig) and (zag), instantiating f with f^{sb} we get the following conditions:

(f^{sb}-zig) If $(w, v) \in S$, there is $v' \in W'$ s.t. $(w', v') \in S'$ and $(v, S^-_{wv})Z(v', S'^-_{w'v'})$;
(f^{sb}-zag) If $(w', v') \in S'$, there is $v \in W$ s.t. $(w, v) \in S$ and $(v, S^-_{wv})Z(v', S'^-_{w'v'})$.

In the same way, we can instantiate f with any of the concrete model update functions mentioned in Section 2.

Theorem 1 (Invariance.). *Let* f *be a family of model update functions, then* $\mathcal{M}, w \leftrightarroweq_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$ *implies* $\mathcal{M}, w \equiv_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$*.*

Proof. We prove the theorem by structural induction. The base case holds by (atomic harmony), and the $\wedge$ and $\neg$ cases are trivial.

Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$, and Z such that $(w, R)Z(w', R')$. [$\Diamond\varphi$ **case:**] Suppose $\mathcal{M}, w \models \Diamond\varphi$. Then there is v in W s.t. $(w, v) \in R$ and $\mathcal{M}, v \models \varphi$. Since Z is a bisimulation, by (zig) we have $v' \in W'$ s.t. $(w', v') \in R'$ and $(v, R)Z(v', R')$. By inductive hypothesis, $\mathcal{M}', v' \models \varphi$ and by definition $\mathcal{M}', w' \models \Diamond\varphi$. For the other direction use (zag).

[$\blacklozenge_f\varphi$ **case:**] Suppose $\langle W, R, V\rangle, w \models \blacklozenge_f\varphi$. Then there is $(v, S) \in f_W(w, R)$ s.t. $\langle W, S, V\rangle, v \models \varphi$. Because Z is a bisimulation, by (f-zig) we have $(v', S') \in f_{W'}(w', R')$ s.t. $(v, S)Z(v', S')$. By inductive hypothesis, $\langle W', S', V'\rangle, v' \models \varphi$ and by definition $\langle W', R', V'\rangle, w' \models \blacklozenge_f\varphi$. For the other direction use (f-zag). □

Clearly the result holds when we extend $\mathcal{ML}$ with any set of relation-changing modal operators. It suffices to require that the bisimulation comply with the various (f-zig) and (f-zag) conditions corresponding to all operators.

The Invariance Theorem proves that bisimilarity defines an equivalence relation that is as least as fine as the one defined by modal equivalence. Over certain classes of models the two notions actually coincide. These classes are usually called Hennessy-Milner classes and the theorem stating the equivalence is called a Hennessy-Milner Theorem [38].

A well known result establishes that ω-saturated models are a Hennessy-Milner class for many modal languages (see [10] for details). We will define a suitable notion of ω-saturation for relation-changing modal logics and prove a Hennessy-Milner Theorem with respect to the corresponding class of models.

Definition 11 (f-saturation). *Let $\mathcal{M} = \langle W, R, V\rangle$ be a model, $X \subseteq W \times 2^{W^2}$, f a family of model update functions and Σ a set of $\mathcal{ML}(\blacklozenge_f)$-formulas. Σ is* satisfiable over X in $\mathcal{M}$ *if there is some $(u, S) \in X$ such that $\langle W, S, V\rangle, u \models \varphi$, for all $\varphi \in \Sigma$ (we will not mention $\mathcal{M}$ when it is obvious from context). Σ is* finitely satisfiable over X in $\mathcal{M}$ *if each finite subset of Σ is satisfiable over X.*

We say that $\mathcal{M} = \langle W, R, V\rangle$ is f-saturated *if for all Σ, and for all pairs $(s, S) \in \mathrm{Img}(f_W) \cup \{(w, R) \mid w \in W\}$ whenever Σ is finitely satisfiable over $X = \{(t, T) \mid (t, T) \in f_W(s, S)\}$ then it is satisfiable over X; and for all $w \in W$ whenever Σ is finitely satisfiable over $X = \{(t, S) \mid (w, t) \in S\}$ then it is satisfiable over X.*

The definition of f-saturation is a variation of the standard definition of ω-saturation and intuitively, requires ω-saturation in each possible updated model, and also with respect to the set of possible model updates in each state.

Proposition 7. *Let f be a family of model update functions, and let $\mathcal{M}, w$, $\mathcal{M}', w'$ be two f-saturated models. Then $\mathcal{M}, w \equiv_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$ implies $\mathcal{M}, w$ $\underline{\leftrightarrow}_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$.*

Proof. We prove that when two f-saturated pointed models satisfy the same formulas, they are bisimilar.

Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be given, and let f be a family of model update functions. Define the relation $\leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)}$ over $\mathrm{Img}(f_W) \cup \{(w, R)\} \times \mathrm{Img}(f_{W'}) \cup \{(w', R')\}$ such that $(v, S) \leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)} (v', S')$ holds if and only if $\langle W, S, V\rangle, w \equiv_{\mathcal{ML}(\blacklozenge_f)} \langle W', S', V'\rangle, w'$. Notice that $(w, R) \leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)} (w', R')$ by hypothesis. We show that $\leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)}$ is an $\mathcal{ML}(\blacklozenge_f)$-bisimulation.

We only prove the f-zig condition. Assume $(s, S) \leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)} (s', S')$ and let $(t, T) \in f_W(s, S)$, we should prove that there is $(t', T') \in f_W(s', S')$ such that $(t, T) \leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)} (t', T')$.

Let $\Sigma = \{\varphi \mid \langle W, T, V\rangle, t \models \varphi\}$, for every finite $\Delta \subseteq \Sigma$ we have $\langle W, S, V\rangle, s \models \blacklozenge_f \bigwedge \Delta$. By $\leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)}$, $\langle W', S', V'\rangle, s' \models \blacklozenge_f \bigwedge \Delta$, then there is some $(t'_\Delta, T'_\Delta) \in f(s', S')$ such that $\langle W', T'_\Delta, V'\rangle, t'_\Delta \models \bigwedge \Delta$, i.e., Σ is finitely satisfiable over $X' = \{(t'_\Delta, T'_\Delta) \mid \Delta \in \Sigma\}$. Then, by f-saturation, Σ is satisfiable over X'. Take $(t', T') \in X'$ as the pair that satisfies Σ we have $(t, T) \leftrightsquigarrow_{\mathcal{ML}(\blacklozenge_f)} (t', T')$. □

The notion of model unfolding we introduced in Section 3.3 also gives rise to Hennessy-Milner classes.

Proposition 8. *Let $\mathcal{M}$ and $\mathcal{M}'$ be two models and let f be a family of model update functions. Assume that the unfolded models $\mathcal{M}_f$ and $\mathcal{M}'_f$ are image-finite (i.e., each state has a finite number of immediate successors). Then for every $w \in W$ and $w' \in W'$, $\mathcal{M}, w \equiv_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$ implies $\mathcal{M}, w \leftrightarroweq_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$.*

Proof. We prove that the relation $\equiv_{\mathcal{ML}(\blacklozenge_f)}$ is a bisimulation. The (atomic harmony) condition is immediate.

For (f-zig), assume that $\langle W, R, V\rangle, w \equiv_{\mathcal{ML}(\blacklozenge_f)} \langle W', R', V'\rangle, w'$ and there exists some $(v, S) \in f_W(w, R)$. To create a contradiction, let us assume that there is no $(v', S') \in f_{W'}(w', R')$ such that $\langle W, S, V\rangle, v \equiv_{\mathcal{ML}(\blacklozenge_f)} \langle W', S', V'\rangle, v'$. First note that $f_{W'}(w', R')$ is non-empty, otherwise $\blacklozenge_f \top$ would hold at $\langle W, R, V\rangle, w$ but not at $\langle W', R', V'\rangle, w'$, which would contradict our assumption. Furthermore, by assumption, $f_{W'}(w', R')$ is image-finite (since $\mathcal{M}'_f$ is image-finite). Let us call $\{(u_1, T_1), \ldots, (u_n, T_n)\} = f_{W'}(w', R')$. By assumption, for every i there exists a formula φ_i such that $\langle W, S, V\rangle, v \models \varphi_i$ and $\langle W', T'_i, V'\rangle, u_i \not\models \varphi_i$.

Then $\langle W, R, V\rangle, w \models \blacklozenge_f(\varphi_1 \wedge \ldots \wedge \varphi_n)$ and $\langle W', R', V'\rangle, w' \not\models \blacklozenge_f(\varphi_1 \wedge \ldots \wedge \varphi_n)$ which is a contradiction.

The (f-zag), (zig) and (zag) conditions can be shown in similar ways. □

Note that the condition requiring image-finiteness of the unfolded models can easily fail. For instance, for f^{br}, the unfolding of a model that is image-finite and infinite is image-infinite. This happens because we have an infinite supply of inaccessible states to which we can build new edges. On the other hand, a model update function on a finite domain is always image-finite. This implies that finite models guarantee image-finiteness of their corresponding unfolded models.

Proposition 9. *Let $\mathcal{M}, w$ and $\mathcal{M}', w'$ be finite pointed models. Let f be a family of model update functions. Then $\mathcal{M}, w \equiv_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$ implies $\mathcal{M}, w \leftrightarroweq_{\mathcal{ML}(\blacklozenge_f)} \mathcal{M}', w'$.*

4.1 Expressivity and Ehrenfeucht-Fraïssé Games

Adding relation-changing modal operators to the basic modal logic increases its expressive power. A basic result for $\mathcal{ML}$ [10] shows that it has the *tree model property*: every satisfiable formula of $\mathcal{ML}$ can be satisfied at the root of a model where the accessibility relation defines a tree (i.e., there is a root, the relation is irreflexive, all elements different from the root can be reached from the root via the transitive closure of the accessibility relation, and no element has two different immediate predecessors). We will show that $\mathcal{ML}$ extended with any of the six concrete relation-changing modal operators introduced in Section 2 lacks the tree model property.

Proposition 10. *$\mathcal{ML}(\langle\cdot\rangle)$ does not have the tree model property, for $\langle\cdot\rangle \in \{\langle\cdot\rangle_{\mathsf{sb}}, \langle\cdot\rangle_{\mathsf{br}}, \langle\cdot\rangle_{\mathsf{sw}}, \langle\cdot\rangle_{\mathsf{gsb}}, \langle\cdot\rangle_{\mathsf{gbr}}, \langle\cdot\rangle_{\mathsf{gsw}}\}$.*

Proof. We show formulas that ensure that the accessibility relation is not a tree. For $\langle\cdot\rangle_{\mathsf{gsb}}$, the result has already been proved in [28], for $\langle\cdot\rangle_{\mathsf{sw}}$ in [3] and for $\langle\cdot\rangle_{\mathsf{sb}}$ and $\langle\cdot\rangle_{\mathsf{br}}$ in [2]. Suppose the following formulas hold at some state w in a model:

1. $\Diamond\Diamond\top \ \wedge \ [\cdot]_{\mathsf{sb}}\Box\bot$, then w is reflexive;
2. $\Diamond\Diamond\top \ \wedge \ [\cdot]_{\mathsf{gsb}}\Box\bot$, then w is reflexive;
3. $\Box\bot \wedge \langle\cdot\rangle_{\mathsf{br}}\Box\bot$, then w and some different state v are unconnected;
4. $\Box\bot \wedge \langle\cdot\rangle_{\mathsf{gbr}}\Box\bot$, then w and some different state v are unconnected;
5. $p \wedge (\bigwedge_{1\leq i\leq 3}\Box^i\neg p) \wedge \langle\cdot\rangle_{\mathsf{sw}}\Diamond\Diamond p$, then w has a reflexive successor;
6. $\Box\bot \wedge \langle\cdot\rangle_{\mathsf{gsw}}\Diamond\top$, then w has an incoming edge.

In each case, the formula cannot be satisfied at the root of a tree.

1) The formula $\varphi = \Diamond\Diamond\top \wedge [\cdot]_{\mathsf{sb}}\Box\bot$ is true at a state w in a model, only if w is reflexive. Suppose we evaluate φ at some state w of an arbitrary model. The static part of the formula $\Diamond\Diamond\top$ ensures it is possible to take two steps in the accessibility relation. The dynamic part of the formula $[\cdot]_{\mathsf{sb}}\Box\bot$ tells us that after moving through any edge in the accessibility relation and eliminating it, we are at a dead end. This can only happen if the state w is reflexive and does not have any other outgoing links.

2) Similar to the previous case.

3) The formula $\varphi = \Box\bot \wedge \langle\cdot\rangle_{\mathsf{br}}\Box\bot$ is only satisfiable in models that have at least two unconnected states. The static part of the formula ($\Box\bot$) establishes that the evaluation state has no successors. The dynamic part ($\langle\cdot\rangle_{\mathsf{br}}\Box\bot$) tells us that after we create a new arrow from the evaluation state to an inaccessible state, we are at a dead end. In both cases, the $\Box\bot$ part guarantees that the corresponding evaluation state has no successors, then it follows that they are not connected.

4) Similar to the previous case.

5) The formula $\varphi = p \wedge (\bigwedge_{1\leq i\leq 3}\Box^i\neg p) \wedge \langle\cdot\rangle_{\mathsf{sw}}\Diamond\Diamond p$ is true at a state w in a model, only if w has a reflexive successor. Suppose we evaluate φ at some state w of an arbitrary model. The static part of the formula $p \wedge (\bigwedge_{1\leq i\leq 3}\Box^i\neg p)$ makes sure

that p is true at w and that no p state is reachable within three steps from w (in particular, w cannot be reflexive). Because $\blacklozenge_{\mathsf{sw}}\Diamond\Diamond p$ is true at w, there should be an R-successor v where $\Diamond\Diamond p$ holds once the accessibility relation has been updated to R^*_{vw}. That is, v has to reach a p-state in exactly two R^*_{vw}-steps. But the only p-state sufficiently close is w, which is reachable in one step. As w is not reflexive, v has to be reflexive so that we can linger at v for one loop and reach p in the correct number of steps.

6) The formula $\varphi = \Box\bot \wedge \blacklozenge_{\mathsf{gsw}}\Diamond\top$ is true at a state with an incoming edge. Indeed, $\Box\bot$ tells us that we are at a dead end, but $\blacklozenge_{\mathsf{gsw}}\Diamond\top$ establishes that after swapping around some edge in the model, we are not longer in a dead end. This happens if in the original model, the evaluation state has an incoming edge. $\square$

As the six logics we introduced are conservative extensions of $\mathcal{ML}$, we have shown that each is strictly more expressive than $\mathcal{ML}$. Now, a natural question is whether these dynamic logics are different from each other.

We have presented bisimulations in a relational perspective. However, it is sometimes difficult to argue that two models are bisimilar, as the bisimulation may be too unwieldy to define and verify. An alternative, equivalent, notion of indistinguishability can be defined in terms of Ehrenfeucht-Fraïssé Games [14,35].

Definition 12 (Ehrenfeucht-Fraïssé Games). *Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be two models, $w \in W$, $w' \in W'$ and let f be a family of model update functions. Let $S \subseteq W^2$ and $S' \subseteq W'^2$, an* Ehrenfeucht-Fraïssé game *$EF_{\blacklozenge_f}(\mathcal{M}, \mathcal{M}', (w, S), (w', S'))$ is defined as follows. There are two players called* Spoiler *and* Duplicator. *The game stops and Duplicator immediately loses if w and w' do not satisfy the same propositional symbols. Otherwise, the game starts, with the players moving alternatively. Spoiler always makes the first move in a turn of the game, starting by choosing in which model he will make a move. The game continues in one of the following ways:*

1. *Spoiler chooses v such that $(w, v) \in S$. If there is no such v, the game stops and Duplicator wins. Otherwise, Duplicator has to choose v' such that $(w', v') \in S'$, with v and v' satisfying the same propositional symbols. If there is no such v', Spoiler wins. Otherwise the game continues with the configuration $EF_{\blacklozenge_f}(\mathcal{M}, \mathcal{M}', (v, S), (v', S'))$. If Spoiler starts by choosing an element in $\mathcal{M}'$, the same process has to be followed by exchanging the models where each player has to choose.*
2. *Spoiler chooses (v, T) such that $(v, T) \in f_W(w, S)$. If there is no such (v, T), the game stops and Duplicator wins. Otherwise, Duplicator has to choose (v', T') such that $(v', T') \in f_W(w', S')$, with v and v' satisfying the same propositional symbols. If there is no such (v', T'), Spoiler wins. Otherwise the game continues with the configuration $EF_{\blacklozenge_f}(\mathcal{M}, \mathcal{M}', (v, T), (v', T'))$. If Spoiler starts by choosing an element in $\mathcal{M}'$, the same process has to be followed by exchanging the models where each player has to choose.*

Duplicator wins on infinite runs.

The winning conditions for $EF_{\diamondsuit_f}(\mathcal{M}, \mathcal{M}', (w, S), (w', S'))$ establish that before the game begins, Duplicator immediately loses if w and w' do not coincide in the propositional symbols. In subsequent rounds, if Duplicator responds with a successor that differs in the atomic propositions with respect to the point chosen by Spoiler, Duplicator loses. If one player cannot move, the other wins, and Duplicator wins on infinite runs. Given these conditions, observe that exactly one of Spoiler or Duplicator wins each game. Given two models $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$, $w \in W$ and $w' \in W'$, we write $\mathcal{M}, w \equiv^{EF}_{\diamondsuit_f} \mathcal{M}', w'$ when Duplicator has a winning strategy for $EF_{\diamondsuit_f}(\mathcal{M}, \mathcal{M}', (w, R), (w', R'))$.

There is an obvious resemblance between the conditions for bisimulations and the rules of the games.

Proposition 11. *Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be two models, $w \in W$, $w' \in W'$ and let f be a family of model update functions. Then $\mathcal{M}, w \equiv^{EF}_{\diamondsuit_f} \mathcal{M}', w'$ iff $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\diamondsuit_f)} \mathcal{M}', w'$.*

Proof. Let Z be a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. Duplicator can answer correctly every move of Spoiler by choosing the appropriate pair in Z. The (zig), (zag), (f-zig) and (f-zag) conditions ensure that she always has a correct move available. In the other direction, a winning strategy defines a correct move for Duplicator, as a response to each possible move of Spoiler. A relation between $\mathcal{M}$ and $\mathcal{M}'$ can be defined in terms of this information, which will satisfy the conditions for bisimulation. □

Proposition 11 establishes that bisimulations and Ehrenfeucht-Fraïssé Games define the same notion of indistinguishability for the family of languages we are investigating. In this way, Ehrenfeucht-Fraïssé Games give us an operational way to check if two models are bisimilar, which is a fundamental tool for the comparison of the expressive power of relation-changing modal logics. In the rest of the section, we will decide whether the logics obtained by extended the basic modal logic with $\diamondsuit_{\mathsf{sb}}$, $\diamondsuit_{\mathsf{br}}$, $\diamondsuit_{\mathsf{sw}}$, $\diamondsuit_{\mathsf{gsb}}$, $\diamondsuit_{\mathsf{gbr}}$ and $\diamondsuit_{\mathsf{gsw}}$ are all pairwise distinct in terms of expressive power.

We use the following standard definition of when a logic is at least as expressive as another.

Definition 13 **($\mathcal{L} \leq \mathcal{L}'$).** *We say that $\mathcal{L}'$ is* at least as expressive as *$\mathcal{L}$ (notation $\mathcal{L} \leq \mathcal{L}'$) if there is a function Tr between formulas of $\mathcal{L}$ and $\mathcal{L}'$ such that for every model $\mathcal{M}$ and every formula φ of $\mathcal{L}$ we have that*

$$\mathcal{M} \models_{\mathcal{L}} \varphi \text{ iff } \mathcal{M} \models_{\mathcal{L}'} \mathsf{Tr}(\varphi).$$

$\mathcal{M}$ is seen as a model of $\mathcal{L}$ on the left and as a model of $\mathcal{L}'$ on the right, and we use in each case the appropriate semantic relation $\models_{\mathcal{L}}$ or $\models_{\mathcal{L}'}$ as required.

We say that $\mathcal{L}$ and $\mathcal{L}'$ are incomparable *if $\mathcal{L} \not\leq \mathcal{L}'$ and $\mathcal{L}' \not\leq \mathcal{L}$.*

According to this definition, to prove that $\mathcal{L} \not\leq \mathcal{L}'$, it suffices to exhibit two models which are bisimilar for $\mathcal{L}'$ and distinguishable by $\mathcal{L}$. Formally, we need models $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$ and states $w \in W, w' \in W'$ such that (w, R) and (w', R') belong to an $\mathcal{L}'$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$, together with an $\mathcal{L}$ formula φ such that $\mathcal{M}, w$ and $\mathcal{M}', w'$ disagree on it.

Proposition 12. *The expressive power of all pairs of different logics among* $\mathcal{ML}(\Diamond_{\mathsf{sb}})$, $\mathcal{ML}(\Diamond_{\mathsf{br}})$, $\mathcal{ML}(\Diamond_{\mathsf{sw}})$, $\mathcal{ML}(\Diamond_{\mathsf{gsb}})$, $\mathcal{ML}(\Diamond_{\mathsf{gbr}})$ *and* $\mathcal{ML}(\Diamond_{\mathsf{gsw}})$ *are incomparable, except perhaps for the pair of* $\mathcal{ML}(\Diamond_{\mathsf{sw}})$ *and* $\mathcal{ML}(\Diamond_{\mathsf{gsw}})$.

Proof. Most of the results are summed up in Figure 1. For every pair of pointed models $\mathcal{M}, w$ and $\mathcal{M}', w'$ shown in the figure, and for all formulas φ of the column "Differentiated by", we have that $\mathcal{M}, w \not\models \varphi$ and $\mathcal{M}', w' \models \varphi$. For all corresponding logics $\mathcal{L}$ of the column "Bisimilar for", we have that $\mathcal{M}, w$ is $\mathcal{L}$ bisimilar to $\mathcal{M}', w'$. For each pair of different logics mentioned in the Proposition, it is possible to find in the table a pair of bisimilar models and differentiating formulas proving $\mathcal{L} \not\leq \mathcal{L}'$.

We treat the case of $\mathcal{ML}(\Diamond_{\mathsf{gbr}}) \not\leq \mathcal{ML}(\Diamond_{\mathsf{br}})$ separately as the model involved is complex. For this case, we give the description of an infinite model $\mathcal{M}$ with two states w and v such that $\mathcal{M}, w$ and $\mathcal{M}, v$ are $\mathcal{ML}(\Diamond_{\mathsf{br}})$-bisimilar and there is an $\mathcal{ML}(\Diamond_{\mathsf{gbr}})$-formula φ such that $\mathcal{M}, w \models \varphi$ and $\mathcal{M}, v \not\models \varphi$.

Let a *piece* be a part of a model with a finite and non-zero number of states and some relation between them. Let a *collection* be the disjoint union of an infinite number of copies of the same piece. Let $\mathcal{M}$ be the disjoint union of all collections obtained from all pieces involved. Let w be the root of the following piece: $\bullet \rightarrow \bullet$, and v the root of the following piece: $\bullet \leftarrow \bullet \rightarrow \bullet$. First, notice that $\Diamond_{\mathsf{gbr}}\Box\Diamond\top$ is true at $\mathcal{M}, w$ and false at $\mathcal{M}, v$. Now we show that $\mathcal{M}, w$ and $\mathcal{M}, v$ are $\mathcal{ML}(\Diamond_{\mathsf{br}})$-bisimilar using Ehrenfeucht-Fraïssé games. We present a winning strategy for Duplicator. $\mathcal{M}, w$ and $\mathcal{M}, v$ are bisimilar for the basic modal logic. We show that after any $\Diamond_{\mathsf{br}}$ move in any model, it is possible to do a $\Diamond_{\mathsf{br}}$ move in the other model that leads to a modally bisimilar part of the model. Assume Spoiler does a bridge to some part of the model. As the model consist of pieces of finite size, evaluation moves to a finite connected component. Duplicator only needs to do a bridge to a finite connected component of the same shape.

We now discuss all other cases, and provide some further details on the models and formulas shown in Figure 1. That the pairs of models shown disagree on the given formulas can be easily verified. That the models are bisimilar for the given logics is also easily verified in most cases, except for the following (we will use Ehrenfeucht-Fraïssé games to check bisimilarity).

Row 1. The models are bisimilar for $\mathcal{ML}(\Diamond_{\mathsf{sb}})$ and $\mathcal{ML}(\Diamond_{\mathsf{sw}})$ since the evaluation states have no successors.

Row 2. The models are bisimilar for $\mathcal{ML}(\Diamond_{\mathsf{br}})$ and $\mathcal{ML}(\Diamond_{\mathsf{gbr}})$ since they are bisimilar for the basic modal logic and their relations is complete. They are also bisimilar for $\mathcal{ML}(\Diamond_{\mathsf{sw}})$ and $\mathcal{ML}(\Diamond_{\mathsf{gsb}})$ since swapping a reflexive arrow has no effect (and can be done in both models), and swapping non-reflexive arrows on the rightmost model cannot create dead-ends.

Row 3. The models are bisimilar for $\mathcal{ML}(\Diamond_{\mathsf{gsb}})$ and $\mathcal{ML}(\Diamond_{\mathsf{sb}})$ because they are bisimilar for $\mathcal{ML}$, they are acyclic and (for $\mathcal{ML}(\Diamond_{\mathsf{gsb}})$) they contain the

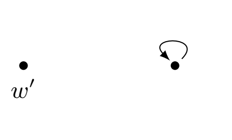
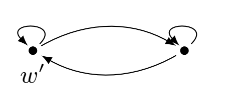
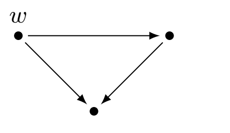
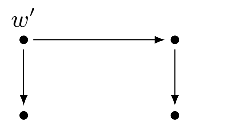
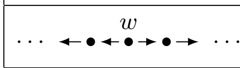
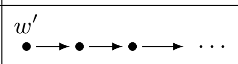
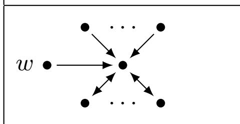
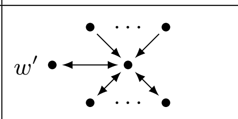
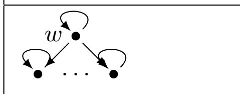
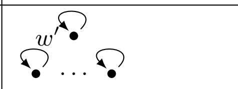
$\mathcal{M}, w$	$\mathcal{M}', w'$	Differentiated by	Bisimilar for
		$\blacklozenge_{\mathsf{br}}\blacklozenge_{\mathsf{br}}\top$ $\blacklozenge_{\mathsf{gbr}}\blacklozenge_{\mathsf{gbr}}\top$ $\blacklozenge_{\mathsf{gsb}}\top$ $\blacklozenge_{\mathsf{gsw}}\top$	$\mathcal{ML}(\blacklozenge_{\mathsf{sb}})$ $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$
		$\blacklozenge_{\mathsf{sb}}\Diamond\top$ $\blacklozenge_{\mathsf{gsb}}\Diamond\top$	$\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ $\mathcal{ML}(\blacklozenge_{\mathsf{br}})$ $\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$ $\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$
		$\blacklozenge_{\mathsf{sw}}\Diamond\Diamond\Diamond\Box\bot$ $\Diamond\blacklozenge_{\mathsf{gsw}}\Diamond\Diamond\Diamond\Box\bot$ $\blacklozenge_{\mathsf{br}}\blacklozenge_{\mathsf{br}}\top$ $\blacklozenge^{6}_{\mathsf{gbr}}\blacklozenge_{\mathsf{gbr}}\top$	$\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$ $\mathcal{ML}(\blacklozenge_{\mathsf{sb}})$
		$\blacklozenge_{\mathsf{sw}}\Diamond\Box\bot$ $\blacklozenge_{\mathsf{gsw}}\Box\bot$	$\mathcal{ML}(\blacklozenge_{\mathsf{br}})$ $\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$
		$\blacklozenge_{\mathsf{sb}}\Diamond\Box\bot$	$\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$
		$\blacklozenge^{3}_{\mathsf{br}}\top$ $\blacklozenge^{3}_{\mathsf{gbr}}\top$	$\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$
		$\blacklozenge_{\mathsf{br}}\top$	$\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$

Fig. 1. Bisimilar models and distinguishing formulas.

same number of edges. Then each time that Spoiler moves to some successor, or deletes an edge, Duplicator can mimic the move.

Row 4. The models are $\mathcal{ML}(\blacklozenge_{\mathsf{br}})$-bisimilar since they are infinite, hence for every link that Spoiler adds, Duplicator answers by adding a new link to a modally bisimilar component.

Row 5. The models are identical except for the point of evaluation. The graph is a star that has infinitely many ingoing branches, and infinitely many ingoing-outgoing branches. w is a point located at the end of an ingoing branch, and w' is at the end of an ingoing-outgoing branch. We have to check that there is a $\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$-bisimulation between the models. If Spoiler moves to the center of the star, Duplicator can do the same and both situations become indistinguishable. If Spoiler deletes one of the ingoing edges that has w or w' as origin, then Duplicator does the same on the other graph, and any further edge deletion can also be imitated. If Spoiler deletes the outgoing edge that goes from the center of the graph towards w', then Duplicator can delete any outgoing edge without changing the graph, given that there are infinitely many edges of both kinds.

Row 6. The models are bisimilar for $\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$ since the connected component from the evaluation states cannot be changed by swapping.

Row 7. The models are bisimilar for $\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$ since adding arrows globally is always possible, but does not change that both models are bisimilar to a single reflexive state. □

Notice that, in particular, we have shown that the local versions of sabotage and bridge, cannot be simulated by a combination of their global versions and the classical diamond. The case for $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ and $\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$ remains open (in particular it is unknown if there is a translation from $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ to $\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$ or not), but we conjecture that their expressive power is also incomparable.

5 Model Checking

In this section we establish complexity results for the model checking task in the various dynamic modal logics we presented. The model checking problem consists in, given a pointed model $\mathcal{M}, w$ and a formula φ, deciding whether $\mathcal{M}, w \models \varphi$. All the results are established using a similar argument: hardness proofs are done by encoding the satisfiability problem of Quantified Boolean Formulas (QBF) [31] as the model checking problem of each logic. For each logic involved, we simulate variable assignment of QBF as a model modification done by deleting, adding or swapping edges during model checking.

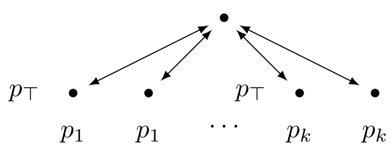
PSpace-hardness for global sabotage was already proved in [29,28], but we provide here a more direct proof.

Proposition 13. *For* $\blacklozenge \in \{\blacklozenge_{\mathsf{sb}}, \blacklozenge_{\mathsf{br}}, \blacklozenge_{\mathsf{sw}}, \blacklozenge_{\mathsf{gsb}}, \blacklozenge_{\mathsf{gbr}}, \blacklozenge_{\mathsf{gsw}}\}$*, model checking for any of* $\mathcal{ML}(\blacklozenge)$ *is* PSpace*-hard.*

Proof. We reduce the PSpace-hard satisfiability problem of QBF to the model checking problem of each of these logics. We give a complete proof for $\blacklozenge_{\mathsf{sw}}$. For the other operators a similar strategy establishes the result.

Let α be a QBF formula with variables $\{x_1, \ldots, x_k\}$. Without loss of generality we can assume that α has no free variables and no variable is quantified twice. One can build in polynomial time the relational structure $\mathcal{M}_k = \langle W, R, V\rangle$ over a signature with one relational symbol and propositions $\{p_\top, p_1, \ldots, p_k\}$, where:

$$\begin{aligned} W &= \{w\} \cup \{w_i^1, w_i^0 \mid 1 \leq i \leq k\} \\ V(p_i) &= \{w_i^1, w_i^0\} \\ V(p_\top) &= \{w_i^1 \mid 1 \leq i \leq k\} \\ R &= \{(w, w_i^1), (w, w_i^0), \\ &\qquad (w_i^1, w), (w_i^0, w) \mid 1 \leq i \leq k\} \end{aligned}$$



Let $(\ )'$ be the following linear translation from QBF to $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$:

$$\begin{aligned} (\exists x_i.\alpha)' &= \blacklozenge_{\mathsf{sw}}(p_i \wedge \Diamond(\alpha)') \\ (x_i)' &= \neg\Diamond(p_i \wedge p_\top) \\ (\neg\alpha)' &= \neg(\alpha)' \\ (\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'. \end{aligned}$$

It remains to see that α is satisfiable iff $\mathcal{M}_k, w \models (\alpha)'$ holds. Let us write $v \models_{\text{qbf}} \alpha$ if valuation $v : \{x_1, \ldots, x_k\} \to \{0,1\}$ satisfies α. For a model $\mathcal{M}$ with relation R we define $v_R : \{x_1, \ldots, x_k\}$ as "$v_R(x_i) = 1$ iff $(w, w_i^1) \notin R$", in the present case, iff the link between w and w_i^1 has been swapped.

Let β be any subformula of α. We show by induction on β that $\mathcal{M}, w \models (\beta)'$ iff $v_R \models_{\text{qbf}} \beta$. The first observation is that R satisfies i) if x_i is free in β, then $(w, w_i^1) \notin R$ or $(w, w_i^0) \notin R$ but not both, and ii) if x_i is not free in β then $(w, w_i^1) \in R$ and $(w, w_i^0) \in R$. From here it will follow that $\mathcal{M}_k, w \models (\alpha)'$ iff $v \models_{\text{qbf}} \alpha$ for any v since α has no free variables, iff α is satisfiable.

For the base case, $v_R \models_{\text{qbf}} x_i$ iff $(w, w_i^1) \notin R$ which implies (by definition of $\mathcal{M}_k$) $\mathcal{M}, w \models (x_i)'$. For the other direction, suppose $\mathcal{M}, w \not\models (x_i)'$. Hence $\mathcal{M}, w \models \Diamond(p_i \wedge p_\top)$ which implies $(w, w_i^1) \in R$ and $v_R \not\models_{\text{qbf}} x_i$.

The Boolean cases follow directly from the inductive hypothesis.

Consider the case $\beta = \exists x_i.\gamma$. Since no variable is bound twice in α we know $(w, w_{x_i}^1) \in R$ and $(w, w_i^0) \in R$. We have $v_R \models_{\text{qbf}} \beta$ iff $(v_R[x_i \mapsto 0] \models_{\text{qbf}} \gamma$ or $v_R[x_i \mapsto 1] \models_{\text{qbf}} \gamma)$ iff $(v_{R^*_{w_i^0 w}} \models_{\text{qbf}} \gamma$ or $v_{R^*_{w_i^1 w}} \models_{\text{qbf}} \gamma)$. By inductive hypothesis, this is the case if and only if $(\mathcal{M}^*_{w_i^0 w}, w_i^0 \models \Diamond(\gamma)'$ or $\mathcal{M}^*_{w_i^1 w}, w_i^1 \models \Diamond(\gamma)')$ iff $\mathcal{M}, w \models \blacklozenge_{\mathsf{sw}}(p_i \wedge \Diamond(\gamma)')$ iff $\mathcal{M}, w \models (\exists x_i.\gamma)'$.

This shows that the model checking problem of $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$ is PSpace-hard.

For $\mathcal{ML}(\blacklozenge_{\mathsf{sb}})$ and $\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$, we use the same model construction, but with different translations of QBF formulas.

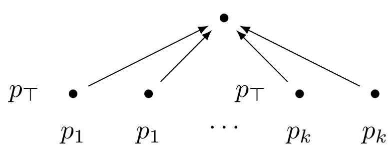
For $\mathcal{ML}(\blacklozenge_{\mathsf{sb}})$, use:
$$\begin{aligned} (\exists x_i.\alpha)' &= \blacklozenge_{\mathsf{sb}}(p_i \wedge \Diamond(\alpha)') \\ (x_i)' &= \neg\Diamond(p_i \wedge p_\top) \\ (\neg\alpha)' &= \neg(\alpha)' \\ (\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'. \end{aligned}$$

For $\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$, use:
$$\begin{aligned} (\exists x_i.\alpha)' &= \blacklozenge_{\mathsf{gsb}}\neg(\Diamond(p_i \wedge p_\top) \wedge \Diamond(p_i \wedge \neg p_\top)) \\ (x_i)' &= \neg\Diamond(p_i \wedge p_\top) \\ (\neg\alpha)' &= \neg(\alpha)' \\ (\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'. \end{aligned}$$

In both cases, showing that a QBF formula α is satisfiable if, and only if, $\mathcal{M}_k, w \models (\alpha)'$ holds can be done similarly to the case of $\mathcal{ML}(\blacklozenge_{\mathsf{sw}})$.

To prove PSpace-hardness for $\mathcal{ML}(\blacklozenge_{\mathsf{br}})$, $\mathcal{ML}(\blacklozenge_{\mathsf{gbr}})$ and $\mathcal{ML}(\blacklozenge_{\mathsf{gsw}})$, build the following model:

$$\begin{aligned} W &= \{w\} \cup \{w_i^1, w_i^0 \mid 1 \le i \le k\} \\ V(p_i) &= \{w_i^1, w_i^0\} \\ V(p_\top) &= \{w_i^1 \mid 1 \le i \le k\} \\ R &= \{(w_i^1, w), (w_i^0, w) \mid 1 \le i \le k\} \end{aligned}$$



For $\mathcal{ML}(\blacklozenge_{\mathsf{br}})$, use:
$$\begin{aligned} (\exists x_i.\alpha)' &= \blacklozenge_{\mathsf{br}}(p_i \wedge \Diamond(\alpha)') \\ (x_i)' &= \Diamond(p_i \wedge p_\top) \\ (\neg\alpha)' &= \neg(\alpha)' \\ (\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'. \end{aligned}$$

For $\mathcal{ML}(\diamondsuit_{\mathsf{gbr}})$, use:
$$\begin{aligned}(\exists x_i.\alpha)' &= \diamondsuit_{\mathsf{gbr}}(\Diamond p_i \ \wedge \ (\alpha)')\\ (x_i)' &= \Diamond(p_i \wedge p_\top)\\ (\neg\alpha)' &= \neg(\alpha)'\\ (\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'.\end{aligned}$$

For $\mathcal{ML}(\diamondsuit_{\mathsf{gsw}})$, use:
$$\begin{aligned}(\exists x_i.\alpha)' &= \diamondsuit_{\mathsf{gsw}}(\Diamond p_i \ \wedge \ (\alpha)')\\ (x_i)' &= \Diamond(p_i \wedge p_\top)\\ (\neg\alpha)' &= \neg(\alpha)'\\ (\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'.\end{aligned}$$

This covers all the cases. □

Proposition 14. *Model checking for* $\mathcal{ML}(\{\diamondsuit_{\mathsf{sb}}, \diamondsuit_{\mathsf{br}}, \diamondsuit_{\mathsf{sw}}, \diamondsuit_{\mathsf{gsb}}, \diamondsuit_{\mathsf{gbr}}, \diamondsuit_{\mathsf{gsw}}\})$ *is in* PSPACE.

Proof. Given a pointed model $\mathcal{M}, w$ and a formula φ, we present a recursive algorithm $\mathsf{CHECK}(\mathcal{M}, w, \varphi)$ that checks whether $\mathcal{M}, w \models \varphi$ and uses a polynomial amount of space in function of its input. Its implementation follows the evaluation of the truth of a formula in a model as described in Definition 4.

For the atomic, negation and conjunctive cases, the algorithm does not use extra memory except for its recursive calls. For the classical diamond case ($\Diamond\psi$), $\mathsf{CHECK}(\mathcal{M}, v, \psi)$ is ran for all v successors of w in $\mathcal{M}$, which uses a (logarithmic space) counter on (at most) all states of the model. For all these cases, CHECK does not need to copy its input $\mathcal{M}$.

On the other hand, the dynamic diamond case $\diamondsuit_f\psi$ involves building a certain number of pointed models $\mathcal{M}', v'$ to compute $\mathsf{CHECK}(\mathcal{M}', v', \psi)$. Each new model uses a polynomial amount of space but this memory is reclaimed when the corresponding computation is over.

As there is at most a linear nesting of dynamic operators in the input formula φ, CHECK will not maintain more than a linear number of models in memory, each one of size polynomial with respect to the input model. Cycling over all models requires a counter that uses at most polynomial space (actually, the model update functions of $\diamondsuit_{\mathsf{sb}}$, $\diamondsuit_{\mathsf{br}}$, $\diamondsuit_{\mathsf{sw}}$, $\diamondsuit_{\mathsf{gsb}}$, $\diamondsuit_{\mathsf{gbr}}$ and $\diamondsuit_{\mathsf{gsw}}$ generate a polynomial amount of models so the counter would only use logarithmic space). □

Theorem 2. *For* $S \subseteq \{\diamondsuit_{\mathsf{sb}}, \diamondsuit_{\mathsf{br}}, \diamondsuit_{\mathsf{sw}}, \diamondsuit_{\mathsf{gsb}}, \diamondsuit_{\mathsf{gbr}}, \diamondsuit_{\mathsf{gsw}}\}$*, model checking for any of the logics* $\mathcal{ML}(S)$ *is* PSPACE*-complete.*

5.1 Formula Complexity and Program Complexity

We established the complexity of the *combined* model checking task, measured in function of the size of an input model and an input formula. It is also possible to consider the task of model checking against a fixed model, measuring its complexity in function of the size of an input formula (this is known as formula complexity). One can also fix a formula and measure the complexity of model checking in function of the length of an input model (this is known as program or data complexity). Both notions were introduced in [48], and it has been shown in [28] that formula and program complexity for model checking in $\mathcal{ML}(\diamondsuit_{\mathsf{gsb}})$

are, respectively, linear and polynomial. We are going to show that these results generalize to many more dynamic operators.

To investigate formula complexity and program complexity, we use model unfoldings. Observe that if a dynamic operator can create only a polynomial number of relation updates in one step, then the number of possible relation updates after n steps is also polynomial:

Proposition 15. *Let f_W be a model update function. Suppose that there exists a $c > 0$ such that for all $s \in W$ and $S \subseteq W \times W$, $\#f(s, S) \in \mathcal{O}((\#W)^c)$ (i.e., f is polynomially bounded). Then there exists $d > 0$ such that $\#R_{f,k} \in \mathcal{O}((\#W)^{dk})$.*

This leads to the general result of formula complexity and program complexity for modal logics equipped with polynomially bounded dynamic operators:

Theorem 3. *Consider $\mathcal{ML}(\blacklozenge_f)$ with f a family of polynomially bounded model update functions.*

1. *The model checking problem for $\mathcal{ML}(\blacklozenge_f)$ with a fixed finite model can be solved in linear time with respect to the size of the formula.*
2. *The model checking problem for $\mathcal{ML}(\blacklozenge_f)$ with a fixed formula can be solved in polynomial time with respect to the size of the finite model.*

Proof. Both parts rely on the result that model checking for $\mathcal{ML}$ is P-complete and can be solved in time $\mathcal{O}(\#\varphi \cdot \#\mathcal{M})$, where $\#\varphi$ is the size of the given $\mathcal{ML}$-formula φ and $\#\mathcal{M}$ is the size of the given model $\mathcal{M}$ [15].

1. Fix a model $\mathcal{M} = \langle W, R, V \rangle$ with a state $w \in W$. Observe that if $\mathcal{M}$ is finite, $\mathcal{M}_f$ is also finite. For some input formula φ, build $\varphi^\Diamond$ in time linear with respect to $\#\varphi$. Then check that $\mathcal{M}_f, (w, R) \models \varphi^\Diamond$ in time linear with respect to $\#\varphi^\Diamond = \#\varphi$.
2. Fix a formula φ, and let $k = \mathsf{dmd}(\varphi)$. For some input model $\mathcal{M} = \langle W, R, V \rangle$ and state $w \in W$, build $\mathcal{M}_{f,k}$ in time polynomial with respect to $\#\mathcal{M}$. Then check that $\mathcal{M}_{f,k}(w, R) \models \varphi^\Diamond$ in time linear with respect to $\#\mathcal{M}_{f,k}$, i.e., in time polynomial with respect to $\#\mathcal{M}$. □

Observe that the modalities $\blacklozenge_{\mathsf{sb}}$, $\blacklozenge_{\mathsf{br}}$, $\blacklozenge_{\mathsf{sw}}$, $\blacklozenge_{\mathsf{gsb}}$, $\blacklozenge_{\mathsf{gbr}}$ and $\blacklozenge_{\mathsf{gsw}}$ all satisfy the conditions of Proposition 15, hence Theorem 3 applies to the corresponding logics. Moreover, the result extends to the basic modal logic equipped with any combination of dynamic operators if each dynamic operator satisfies the conditions of the Proposition.

An example of a dynamic modal operator that does not satisfy the conditions of the theorem is the following modality, which can blow up the number of possible relations to 2^{W^2} in just one step:

$$f_W : (s, S) \mapsto \{(t, T) \mid T \subseteq W \times W, t \in W\}.$$

Evaluating this operator requires considering all models with domain W, some fixed valuation V and all possible binary relations on W.

6 Conclusions

In this article we investigated a framework to define relation-changing modal operators. It is based on the notion of model update functions that take a state in the model and the current accessibility relation, and return the new state of evaluation and the new accessibility relation to be used. We showed that the framework can accommodate a variety of operators, like Van Benthem's sabotage logic [40] and other variants investigated in [2,3,34]. On the other hand, some well known dynamic operators are not covered by the framework. The most important example are those investigated in Dynamic Epistemic Logics [32,19,39,47].

We introduced six different dynamic modal operators with both local and global effects which can add, delete and modify edges in the accessibility relation. The goal was to investigate the degrees of liberty that the operators offered, and how much overlap there was between the logics they define, and the models they can describe. Many of the results we establish refers to these six logics, but we believe that the *techniques* used in the different proofs are sufficiently general to handle other logics that can be accommodated in the framework.

We investigated the logics obtained by adding relation-changing modal operators to the basic modal logic, as fragments of classical logics. First, we introduced a translation from relation-changing modal formulas to second-order formulas, which covers all the operators that can be defined in our framework under the assumption that the family of update model functions can be characterized by a second-order formula. In [34,3], it was shown that $\mathcal{ML}(\Diamond_{\mathsf{sw}})$ and $\mathcal{ML}(\Diamond_{\mathsf{gsb}})$ are proper fragments of first-order logic. We proved the same result for the six concrete relation-changing operators we introduced. The existence of translations into first-order logic opens the way to the use of first-order theorem provers for automated deduction. We also showed a translation that uses model unfolding to converts models and formulas to the basic modal logic with two accessibility relations, over a particular class of models.

We also investigated the expressive power of these languages using bisimulations. We defined bisimulations in a general way, and then instantiated this definition for different, concrete logics. We showed an Invariance Theorem for relation-changing modal logics, and Hennessy-Milner Theorems under different conditions on a model class. We used this definition of bisimulations together with Ehrenfeucht-Fraïssé games to compare the expressive power of different relation-changing modal logics. A direction for future research would be to use properties of the family of model update functions to obtain general expressivity results. In this article, we proved that the six concrete logics introduced are pairwise incomparable in terms of expressive power, except for $\mathcal{ML}(\Diamond_{\mathsf{sw}})$ and $\mathcal{ML}(\Diamond_{\mathsf{gsw}})$. We conjecture that their expressive power is also incomparable.

Finally, we showed that the complexity of model checking is PSpace-complete for the six logics considered. The proofs are fairly uniform, and are based in the encoding of the PSpace-complete QBF satisfiability problem. In each case, a suitable representation for a propositional assignment, and the concrete translation used needs to be defined. Once this is done the proof is straightforward. We first established the complexity of the *combined* model checking task, measured in function of the length of an input model and an input formula. We then considered the task of model checking against a fixed model, measuring its complexity in function of the size of an input formula. Finally, it is also possible to fix a formula and measure the complexity of model checking in function of the length of an input model. In [28] it was shown that the formula complexity and the program complexity of $\mathcal{ML}(\blacklozenge_{\mathsf{gsb}})$ are respectively linear and polynomial. We generalized this result to our general framework using model update functions and we proved, using model unfolding, that the same bounds hold for logics defined by a model update function satisfying certain properties.

An important question, left for future work, is the study of axiomatizations for this family of logics. As a consequence of their high expressive power, it is not possible to define reduction axioms to basic modal logic, such as it is done for several dynamic epistemic logics. The definition of sound and complete axiomatic systems seems non-trivial. For instance, consider the classical *uniform substitution*, one of the main axioms of the basic modal logic. This axiom says that if a formula φ is a theorem, then φ' obtained by substituting all appearances of a propositional variable p by an arbitrary formula ψ is also a theorem. Consider the validity $\blacklozenge_{\mathsf{sw}}p \to \Diamond p$. The uniform substitution of p by $\blacklozenge_{\mathsf{sw}}p$ leads to a non valid formula.

This article shows that some of the dynamic operators that can be captured in the presented framework are strictly more expressive than the basic modal language while they are no more expressive than first-order logic. It would be interesting to investigate in which cases they are sufficiently expressive to be a conservative reduction class for first-order logic (see, e.g., the case of hybrid logics discussed in [5]). Investigating succinctness question for these languages, as is done for different dynamic epistemic logics in, e.g., [30,18], would also be worthwhile.

Another interesting line of research is to exploit the expressive power of relation-changing modal logics to encode dynamic epistemic logics. For example, in Action Model Logic the dynamic operators are defined using complex *action models* which define how the model should be altered. The action models themselves can use formulas of action model logic to define pre- and post-conditions, and as a result the syntax and semantics of the logic is involved. It would be interesting to represent these epistemic logics using model-changing operators (preliminary results have been presented in [17,16,6]). This would result in simpler syntax and semantics which, in turn, could lead so a better understanding of their expressive power, complexity and model and completeness theory.

Acknowledgments: This work was partially supported by grants ANPCyT-PICT-2008-306, ANPCyT-PICT-2010-688, ANPCyT-PICT-2013-2011, STIC-AmSud "MISMT", the FP7-PEOPLE-2011-IRSES Project "Mobility between Europe and Argentina applying Logics to Systems" (MEALS) and the Laboratoire International Associé "INFINIS." We thank the two anonymous reviewers for their useful comments for improvement.

References

1. C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.
2. C. Areces, R. Fervari, and G. Hoffmann. Moving arrows and four model checking results. In L. Ong and R. de Queiroz, editors, *WoLLIC 2012*, volume 7456 of *LNCS*, pages 142–153. Springer Berlin Heidelberg, 2012.
3. C. Areces, R. Fervari, and G. Hoffmann. Swap logic. *Logic Journal of the IGPL*, 22(2):309–332, 2014.
4. C. Areces and D. Gorín. Coinductive models and normal forms for modal logics (or how we learned to stop worrying and love coinduction). *Journal of Applied Logic*, 8(4):305–318, 2010.
5. C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*, pages 821–868. Elsevier, 2006.
6. C. Areces, H. van Ditmarsch, R. Fervari, and F. Schwarzentruber. Logics with copy and remove. In U. Kohlenbach, P. Barceló, and R. de Queiroz, editors, *WoLLIC 2014*, volume 8652 of *LNCS*, pages 51–65. Springer, 2014.
7. G. Aucher. A combined system for update logic and belief revision. Master's thesis, ILLC, Universiteit van Amsterdam, 2003.
8. G. Aucher, P. Balbiani, L. Fariñas Del Cerro, and A. Herzig. Global and local graph modifiers. *Electronic Notes in Theoretical Computer Science (ENTCS), Special issue Proceedings of M4M5*, 231:293–307, 2009.
9. A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge and private suspicions. In I. Gilboa, editor, *Proceedings of TARK 1998*, pages 43–56. Morgan Kaufmann, 1998.
10. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 2001.
11. P. Blackburn, F. Wolter, and J. van Benthem, editors. *Handbook of Modal Logics*. Elsevier, 2006.
12. G. Renardel de Lavalette. Changing modalities. *Journal of Logic and Computation*, 14(2):253–278, 2004.
13. M. de Rijke. Meeting some neighbours. *Logic and information flow*, pages 170–195, 1994.
14. H. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer-Verlag, 1984.
15. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about knowledge*. The MIT Press, 2003.
16. R. Fervari. *Relation-Changing Modal Logics*. PhD thesis, Facultad de Matemática Astronomía y Física, Universidad Nacional de Córdoba, Argentina, 2014.
17. T. French, J. Hales, and E. Tay. A composable language for action models. In R. Goré, B. Kooi, and A. Kurucz, editors, *Advances in Modal Logic 10*, pages 197–216. College Publications, 2014.
18. T. French, W. van der Hoek, P. Iliev, and B. Kooi. Succinctness of epistemic languages. In T. Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 881–886, Menlo Park, 2011. AAAI Press.
19. J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999. ILLC Dissertation series DS-1999-01.

20. J. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169, 1997.
21. N. Gierasimczuk, L. Kurzen, and F. Velázquez-Quesada. Learning and teaching as a game: A sabotage approach. In He et al. [22], pages 119–132.
22. X. He, J. Horty, and E. Pacuit, editors. *LORI 2009*, volume 5834 of *LNCS*. Springer, 2009.
23. J. Hintikka. *Knowledge and belief.* Cornell University Press Ithaca, N.Y., 1962.
24. W. Holliday. Trust and the dynamics of testimony. In D. Grossi, L. Kurzen, and F. Velázquez-Quesada, editors, *Logic and Interactive RAtionality. Seminar's yearbook 2009*, pages 118–142. Universiteit van Amsterdam, The Netherlands, 2010.
25. B. Kooi. Expressivity and completeness for public update logic via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2):231–253, 2007.
26. B. Kooi and B. Renne. Arrow update logic. *Review of Symbolic Logic*, 4(4):536–559, 2011.
27. B. Kooi and B. Renne. Generalized arrow update logic. In K. Apt, editor, *Proceedings of TARK 2011*, pages 205–211. ACM, 2011.
28. C. Löding and P. Rohde. Model checking and satisfiability for sabotage modal logic. In P. Pandya and J. Radhakrishnan, editors, *Proceedings of FSTTCS 2003*, volume 2914 of *LNCS*, pages 302–313. Springer, 2003.
29. C. Löding and P. Rohde. Solving the sabotage game is PSPACE-hard. In *Mathematical Foundations of Computer Science 2003*, volume 2747 of *LNCS*, pages 531–540. Springer, Berlin, 2003.
30. C. Lutz. Complexity and succinctness of public announcement logic. In *Proceedings AAMAS-06: Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Japan, 2006.
31. C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
32. J. Plaza. Logics of public communications. In M. Emrich, M. Pfeifer, M. Hadzikadic, and Z. Ras, editors, *Procceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.
33. B. Renne, J. Sack, and A. Yap. Dynamic epistemic temporal logic. In He et al. [22], pages 263–277.
34. P. Rohde. *On games and logics over dynamically changing structures.* PhD thesis, RWTH Aachen, 2006.
35. D. Sangiorgi. On the origins of bisimulation and coinduction. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 31(4), 2009.
36. K. Segerberg. Belief revision from the point of view of doxastic logic. *Logic Journal of the IGPL*, 3(4):535–553, 1995.
37. K. Segerberg. Irrevocable belief revision in dynamic doxastic logic. *Notre Dame Journal of Formal Logic*, 39(3):287–306, 1998.
38. J. van Benthem. *Modal Correspondence Theory.* PhD thesis, Mathematical Institute, University of Amsterdam, 1977. Doctoral Dissertation.
39. J. van Benthem. Logics for information update. In *Proceedings of TARK 2001*, pages 51–67. Morgan Kaufmann, 2001.
40. J. van Benthem. An essay on sabotage and obstruction. In *Mechanizing Mathematical Reasoning*, pages 268–276, 2005.
41. J. van Benthem. Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155, 2007.
42. J. van Benthem and F. Liu. Dynamic logic of preference upgrade. *Journal of Applied Nonclassical Logics*, 17(2):157–182, 2007.
43. J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

44. H. van Ditmarsch. Prolegomena to dynamic logic for belief revision. *Synthese*, 147(2):229–275, 2005.
45. H. van Ditmarsch, W. van der Hoek, and B. Kooi. Public announcements and belief expansion. *Advances in Modal Logic 5*, pages 62–73, 2004.
46. H. van Ditmarsch, W. van der Hoek, and B. Kooi. Dynamic epistemic logic with assignment. In *Proceedings of AAMAS 2005*, pages 141–148, 2005.
47. H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Kluwer, 2007.
48. M. Vardi. The complexity of relational query languages (extended abstract). In H. Lewis, B. Simons, W. Burkhard, and L. Landweber, editors, *Proceedings of STOC 1982*, pages 137–146. ACM, 1982.