

A Survey on Relation-Changing Modal Logics

Carlos Areces^[0000-0001-7845-8503]

Abstract In this chapter, we present results on 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 in the domain of a model. We define a generic framework to characterize this kind of operations. First, we investigate relation-changing modal logics as fragments of other, better investigated, logics, and in particular provide equivalence preserving translations into first order logic and hybrid logic. To investigate their expressive power, we define suitable notions of bisimulation for the logics introduced. We then turn to the complexity of different reasoning problems for these kind of logics. Finally, we discuss existing Hilbert-style axiomatizations, and the special techniques needed to establish completeness.

Key words: relation-changing modal logics, expressive power, complexity, axiomatization

1 Introduction

Modal logics were originally conceived as logics of necessary and possible truths. They are now viewed, more broadly, as logics that explore a wide range of modalities, or modes of truth: epistemic (“it is known that”), doxastic (“it is believed that”), deontic (“it ought to be the case that”), or temporal (“it has been the case that”), among others. From a model-theoretic perspective, the field evolved into a discipline that deals with languages interpreted on various kinds of relational structures or graphs. Nowadays, modal logics are actively used in areas as diverse as software verification, artificial intelligence, semantics and pragmatics of natural language,

Carlos Areces
FAMAF-UNC and CONICET, Córdoba, Argentina, e-mail: carlos.areces@unc.edu.ar

law, philosophy, etc. In this chapter we will assume that the reader has a background on modern modal logic and refer to [23, 22] for basic notions and results.

As mentioned, from an abstract point of view, modal logics can be seen as formal languages used to navigate and explore properties of a given relational structure. If we are interested, on the other hand, in describing how a given relational structure *evolves* (e.g., through time or through the application of certain operations) then classical modal languages seem, a priori, to fall short of the mark. Of course, it is possible to statically model the whole space of possible transformations as a graph, and use modal languages at that level, but this soon becomes unwieldy (see, e.g., [8, 32] for some results using this approach). It is also possible to represent model update conditions as *parts* of the model itself, and interact with them by means of the classical modal language. This is the approach taken by Gabbay in his study of reactive Kripke frames [36, 37]. Alternatively, it is possible to use standard relational models, and use modal languages with *dynamic modalities* encoding the desired changes. There exist several dynamic modal logics that fit in this last approach. A clear example are the dynamic operators introduced in dynamic epistemic logics (see, e.g., [74]). These operators are used to model changes in the epistemic state of an agent by removing edges from the graph that represents the information states the agent considers possible. A less obvious example is given by hybrid logics equipped with the down arrow operator \downarrow which is used to ‘rebind’ names to the current point of evaluation [24, 14].

In this survey we will focus on a family of logics that follow this approach, proposing dynamic operators that can modify the accessibility relation in a relational model. A prototypical example of this kind of language is sabotage logic, introduced by Van Benthem in [68]. The sabotage operator deletes individual edges in a graph and was introduced to model what Van Benthem called *the sabotage game*. This game is played on a graph by 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. Blocker, on the other hand, can delete one edge from the graph every time it is her turn. Runner wins if she manages to move from the origin to the final point, while Blocker wins otherwise. Van Benthem discusses in [68] how to transform the sabotage game into a modal logic. This original idea has been studied in several other works (e.g., [52, 61]) where the semantics of the (global) sabotage operator $\langle \text{gsb} \rangle$ is defined as:

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

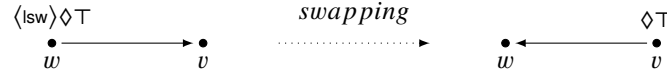
where $\mathcal{M}_{(u,v)}^-$ is identical to \mathcal{M} except that the edge (u, v) has been removed from the accessibility relation. Of course, moves from Runner can be represented by the classical \diamond operator.

It is clear that the $\langle \text{gsb} \rangle$ operator *changes* the model in which a formula is evaluated. As Van Benthem puts it in [68], $\langle \text{gsb} \rangle$ is an “external” modality that takes evaluation to another model, obtained from the current one by deleting some transition.

In this chapter, we investigate different relation-changing operators of this kind. For example, $\langle \text{lsb} \rangle$, for *local sabotage*, is a \diamond operator that destroys the *traversed*

arrow, i.e., it moves evaluation to an adjacent node in the graph, and deletes the edge used¹. In contrast, the *local bridge* operator $\langle \text{ibr} \rangle$ adds an arrow from the current point of evaluation to an inaccessible state of the graph and moves evaluation over there. A global version $\langle \text{gbr} \rangle$ would link two arbitrary, previously non-adjacent nodes in the graph, without changing the evaluation point. This operator can be seen as the opposite to $\langle \text{gsb} \rangle$.

We could consider more complex patterns. E.g., let $\langle \text{lsw} \rangle$ be the operator with the ability to swap the direction of a traversed arrow. The $\langle \text{lsw} \rangle$ 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. A picture will help understand the dynamics of $\langle \text{lsw} \rangle$. The formula $\langle \text{lsw} \rangle \diamond \top$ is true in a model with two related states:



Again, a global version $\langle \text{gsw} \rangle$ can be defined that swaps an arbitrary edge in the graph.

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 chapter 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 f_W defines a dynamic operator. For example, $\langle \text{gsb} \rangle$ would be defined by the function

$$f_W(w, R) = \{(w, R \setminus \{(u, v)\}) \mid (u, v) \in R\},$$

while $\langle \text{lsw} \rangle$ would be defined by

$$f_W(w, R) = \{(v, (R \setminus \{(w, v)\}) \cup \{(v, w)\}) \mid (w, v) \in R\}.$$

In the next sections we introduce the framework we just outlined, and investigate different properties of the resulting logics. We will discuss both specific results for concrete operators, and general results that can be proved when the defining functions satisfy certain constraints. Results by the author, presented in this chapter, were previously published in collaboration with colleagues in [6, 8, 7, 32, 9, 10, 11, 12]. Proper credit will be given when citing results by other authors.

¹ This seems to convey a *resource-sensitive* interpretation of the operator, in line, e.g., with linear logics [41, 66]

2 Basic Definitions

The syntax of the dynamic modal logics we study is a straightforward extension of the basic modal logic (see [23]):

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

$$\text{FORM} ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \diamond\varphi, \mid \langle f \rangle\varphi,$$

where $p \in \text{PROP}$, $\langle f \rangle \in \text{DYN}$ a set of dynamic operators, and $\varphi, \psi \in \text{FORM}$. Other operators are defined as usual. In particular, $\Box\varphi$ is defined as $\neg\diamond\neg\varphi$ and $[f]\varphi$ is defined as $\neg\langle f \rangle\neg\varphi$. Formulas of the basic modal language \mathcal{ML} are those that contains only the \diamond operator besides the Boolean operators. For $S \subseteq \text{DYN}$ 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 $S = \{\langle f \rangle\}$, we write $\mathcal{ML}(\langle f \rangle)$ instead of $\mathcal{ML}(\{\langle f \rangle\})$ ².

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 $V : \text{PROP} \mapsto 2^W$ 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.

The definition below formalizes the idea of relation updates previously introduced.

Definition 3 (Relation updates) Given a domain W , a *relation update for W* is a function $f_W : W \times 2^{W^2} \rightarrow 2^{W \times 2^{W^2}}$, 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 C be a class of models, a *family of model update functions* f is a class of relation updates, one for each domain of a model in C : $f = \{f_W \mid \langle W, R, V \rangle \in C\}$.

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

Clearly, the class of all models is closed under any family of relation updates. In the rest of the chapter we only discuss the class of all models.

Notice, in the definition above, that a relation update is defined relative to a domain. We specifically require that all models with the same domain have the same relation update. 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

² Here, we consider models with only one accessibility relation, with the dynamic operators acting over it. The extension to models with multiple accessibility relations, and their corresponding dynamic modalities, is straightforward.

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 relation updates for \mathcal{C} and $\langle f \rangle$ its associated dynamic operator. Let φ be a formula in $\mathcal{ML}(\langle f \rangle)$. We say that \mathcal{M}, w satisfies φ , and write $\mathcal{M}, w \models \varphi$, when

$$\begin{aligned} \mathcal{M}, w \models p & \quad \text{iff } w \in V(p) \\ \mathcal{M}, w \models \neg\varphi & \quad \text{iff } \mathcal{M}, w \not\models \varphi \\ \mathcal{M}, w \models \varphi \wedge \psi & \quad \text{iff } \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models \diamond\varphi & \quad \text{iff for some } v \in W \text{ s.t. } (w, v) \in R, \mathcal{M}, v \models \varphi \\ \mathcal{M}, w \models \langle f \rangle\varphi & \quad \text{iff for some } (v, R') \in f_W(w, R), \langle W, R', V \rangle, v \models \varphi. \end{aligned}$$

φ is satisfiable if for some \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 $\langle f \rangle$ potentially changes both the state of evaluation and the accessibility relation. On the other hand, the model domain remains the same, and hence all $\langle f \rangle$ operators in a formula are evaluated using the same relation update.

Consider the following relation updates. To simplify notation we use wv as a shorthand for $\{(w, v)\}$ or (w, v) ; context will disambiguate the intended use. Given a binary relation R define the following notation: $R_{wv}^- = R \setminus wv$, $R_{wv}^+ = R \cup wv$, and $R_{wv}^* = (R \setminus wv) \cup wv$.

Define now the following six relation updates, which give rise to natural dynamic modal operators:

$$\begin{aligned} f_W^{\text{lsb}}(w, R) &= \{(v, R_{wv}^-) \mid wv \in R\} & f_W^{\text{gsb}}(w, R) &= \{(w, R_{wv}^-) \mid wv \in R\} \\ f_W^{\text{lbr}}(w, R) &= \{(v, R_{wv}^+) \mid wv \notin R\} & f_W^{\text{gbr}}(w, R) &= \{(w, R_{wv}^+) \mid wv \notin R\} \\ f_W^{\text{lsw}}(w, R) &= \{(v, R_{wv}^*) \mid wv \in R\} & f_W^{\text{gsw}}(w, R) &= \{(w, R_{wv}^*) \mid wv \in R\}. \end{aligned}$$

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 $\langle \text{lsb} \rangle$, $\langle \text{gsb} \rangle$, $\langle \text{lbr} \rangle$, $\langle \text{gbr} \rangle$, $\langle \text{lsw} \rangle$ and $\langle \text{gsw} \rangle$ defined by the relation updates just introduced.

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 $\mathcal{ML}(\langle f \rangle)$ into second-order logic, where $\langle f \rangle$ is defined by a family of relation updates f . We then show that in some cases a translation into first-order logic is possible. We also

discuss how relation-changing modal logics can be seen as multi-modal logics over particular classes of models. Finally, we also discuss the relation between relation-changing logics and hybrid logics.

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 STr .

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

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

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

STr 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 [23].

Proposition 1 Let $\varphi \in \mathcal{ML}$ then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, g_w^x \models \text{STr}_x(\varphi)$, where g_w^x is an arbitrary first-order assignment that satisfies $g_w^x(x) = w$.

The translation STr can be extended to relation-changing modal operators when the family of relation updates 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 relation updates, can redefine the accessibility relation that should be used when translating a dynamic operator.

Definition 6 Let f be a family of relation updates, 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 ,

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

Given a family of relation updates f , and δ_f a formula that defines f , let $\text{STr}_{x,r}$ be as follows

$$\begin{aligned}\text{STr}_{x,r}(p) &= p(x) \\ \text{STr}_{x,r}(\neg\varphi) &= \neg\text{STr}_{x,r}(\varphi) \\ \text{STr}_{x,r}(\varphi \wedge \psi) &= \text{STr}_{x,r}(\varphi) \wedge \text{STr}_{x,r}(\psi) \\ \text{STr}_{x,r}(\diamond\varphi) &= \exists y.(r(x, y) \wedge \text{STr}_{y,r}(\varphi)) \\ \text{STr}_{x,r}(\langle f \rangle\varphi) &= \exists y.\exists s.(\delta_f[v_1/x, V_1/r, v_2/y, V_2/s] \wedge \text{STr}_{y,s}(\varphi)),\end{aligned}$$

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.

Proposition 2 *Let $\varphi \in \mathcal{ML}(\langle f \rangle)$ and let δ_f be a formula defining f . Then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, g_w^x \models \text{STr}_{x,r}(\varphi)$, where g_w^x is an arbitrary second-order assignment that satisfies $g_w^x(x) = w$.*

The function STr translates $\mathcal{ML}(\langle f \rangle)$ -formulas into second-order formulas. For example, a formula characterizing $\langle \text{lsb} \rangle$ is easy to define:

$$\delta_{\langle \text{lsb} \rangle} \doteq V_1(v_1, v_2) \wedge \neg V_2(v_1, v_2) \wedge \\ \forall z. \forall z'. ((v_1, v_2) \neq (z, z') \rightarrow (V_1(z, z') \leftrightarrow V_2(z, z'))).$$

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

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

Formulas for $\langle \text{lbr} \rangle$, $\langle \text{gbr} \rangle$, $\langle \text{lsw} \rangle$ and $\langle \text{gsw} \rangle$ can be defined in a similar way. 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.

3.1 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 \circ^+ with the following semantics: $\langle W, R, V \rangle, w \models \circ^+ \varphi$ 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, $\mathcal{ML}(\circ^+)$ is not compact and hence it cannot be translated into first-order logic.

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

The argument to establish Proposition 3 is similar to the one used for Propositional Dynamic Logic (see [26] for details). Consider the infinite set $\Gamma = \{\circ^+ \diamond p\} \cup \{\square^n \neg p \mid n \geq 0\}$. Every finite subset of Γ is satisfiable, but Γ is not.

We present now translations for $\langle \text{lsb} \rangle$, $\langle \text{lbr} \rangle$, $\langle \text{lsw} \rangle$, $\langle \text{gsb} \rangle$, $\langle \text{gbr} \rangle$, $\langle \text{gsw} \rangle$ to first-order logic, inspired by the translation of $\langle \text{gsb} \rangle$ presented in [61].

Let VAR be a set of first-order variables. We consider a set $S \subseteq \text{VAR} \times \text{VAR}$ 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:

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

where S is a finite set of pairs of variables. In particular $nm \in \emptyset$ is a notation for \perp 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 $\mathcal{ML}(\{\langle \text{lsb} \rangle, \langle \text{gsb} \rangle\})$ (which shares the case for the basic modality \diamond):

$$\begin{aligned}
\text{STr}_{x,S}(\diamond\varphi) &= \exists y.(r(x,y) \wedge xy \notin S \wedge \text{STr}_{y,S}(\varphi)) \\
\text{STr}_{x,S}(\langle \text{lsb} \rangle\varphi) &= \exists y.(r(x,y) \wedge xy \notin S \wedge \text{STr}_{y,S \cup xy}(\varphi)) \\
\text{STr}_{x,S}(\langle \text{gsb} \rangle\varphi) &= \exists y.\exists z.(r(y,z) \wedge yz \notin S \wedge \text{STr}_{x,S \cup yz}(\varphi)),
\end{aligned}$$

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 as part of a shorthand in the definition of the translation.

Proposition 4 *Given φ a formula of $\mathcal{ML}(\langle \text{lsb} \rangle)$ or $\mathcal{ML}(\langle \text{gsb} \rangle)$ and \mathcal{M}, w a pointed model, we have $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, g_w^x \models \text{STr}_{x,\emptyset}(\varphi)$, where g_w^x is an arbitrary first-order assignment that satisfies $g_w^x(x) = w$.*

For $\langle \text{lbr} \rangle$ and $\langle \text{gbr} \rangle$, the set S represents the edges added to the model. We define:

$$\begin{aligned}
\text{STr}_{x,S}(\diamond\varphi) &= \exists y.((r(x,y) \vee xy \in S) \wedge \text{STr}_{y,S}(\varphi)) \\
\text{STr}_{x,S}(\langle \text{lbr} \rangle\varphi) &= \exists y.(\neg(r(x,y) \vee xy \in S) \wedge \text{STr}_{y,S \cup xy}(\varphi)) \\
\text{STr}_{x,S}(\langle \text{gbr} \rangle\varphi) &= \exists y.\exists z.(\neg(r(y,z) \vee yz \in S) \wedge \text{STr}_{x,S \cup yz}(\varphi)).
\end{aligned}$$

For $\langle \text{lsw} \rangle$ and $\langle \text{gsw} \rangle$, S refers to the edges swapped in the model. We define:

$$\begin{aligned}
\text{STr}_{x,S}(\diamond\varphi) &= \exists y.(((r(x,y) \wedge xy \notin S) \vee xy \in S^{-1}) \wedge \text{STr}_{y,S}(\varphi)) \\
\text{STr}_{x,S}(\langle \text{lsw} \rangle\varphi) &= (r(x,x) \wedge \text{STr}_{x,S}(\varphi)) \\
&\quad \vee \exists y.(x \neq y \wedge r(x,y) \wedge xy \notin (S \cup S^{-1}) \wedge \text{STr}_{y,S \cup xy}(\varphi)) \\
&\quad \vee \bigvee_{yz \in S} (x = z \wedge \text{STr}_{y,S \setminus yz \cup zy}(\varphi)) \\
\text{STr}_{x,S}(\langle \text{gsw} \rangle\varphi) &= (\exists y.r(y,y) \wedge \text{STr}_{x,S}(\varphi)) \\
&\quad \vee \exists y.\exists z.(y \neq z \wedge r(y,z) \wedge yz \notin (S \cup S^{-1}) \wedge \text{STr}_{x,S \cup yz}(\varphi)) \\
&\quad \vee \bigvee_{yz \in S} \text{STr}_{x,S \setminus yz \cup zy}(\varphi).
\end{aligned}$$

In all cases, the translations preserve equivalence.

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

We refer to [9] for the proofs of the last two propositions.

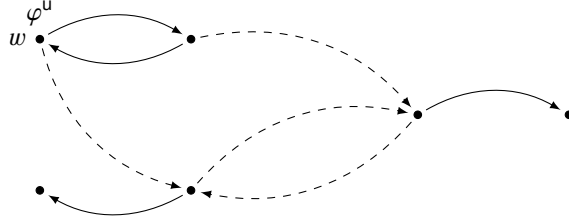
3.2 Model Unfolding

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 $\mathcal{ML}(\langle \text{!sw} \rangle)$:



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



This unfolded model has two accessibility relations: one represents the relation in each update model, the other is an “external” relation that links updated models. At the syntactic level, we can rewrite φ into a modal formula φ^u in the basic modal language \mathcal{ML} with two classic modalities, say \diamond_1 and \diamond_2 , with \diamond systematically replaced by \diamond_1 , and $\langle \text{!} \rangle$ by \diamond_2 . They will be interpreted using the relations in the unfolded model. Define relation updates as follows:

Definition 7 (Relation updates) Let W be a domain, R a relation on W , f_W a relation update 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 8 (Model unfolding) Let $\mathcal{M} = \langle W, R, V \rangle$ and let f be a family of relation updates. 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 $\varphi \in \mathcal{ML}(\langle f \rangle)$. Then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}_{f, \text{dmd}(\varphi)}, (w, R) \models \varphi^u$, where $\text{dmd}(\varphi)$ is the dynamic modal depth of φ , i.e., the maximum nesting of dynamic modalities in φ .*

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.

We refer to [32] for more details on this approach.

3.3 Hybrid Logics

In [10] we investigate relation-changing logics as fragments of hybrid logics [14]. We defined translations for the six concrete relation-changing operators into $\mathcal{H}(\mathbf{E}, \downarrow)$, the basic modal logic extended with nominals, the down arrow binder \downarrow , and the universal modality \mathbf{E} ; but, in some cases, the translations fall into the less expressive hybrid logic $\mathcal{H}(\cdot, \downarrow)$, i.e., with the satisfiability operator \cdot instead of \mathbf{E}^3 . We also showed that these relation-changing logics are strictly less expressive than the hybrid logics they are translated into. These translations let us take advantage of known results on decidable fragments of $\mathcal{H}(\mathbf{E}, \downarrow)$ and $\mathcal{H}(\cdot, \downarrow)$ to find decidable fragments of relation-changing logics. Moreover, these translations have been implemented into the hybrid logic prover HTab [46] so that it can now reason on relation-changing formulas. For brevity, here we only provide details for the sabotage operators $\langle \text{lsb} \rangle$ and $\langle \text{gsb} \rangle$.

³ The proposed translations take inspiration from the previously shown translations into first-order logic, and we use a similar notation. E.g., a pair (n, m) of nominals will be written simply as nm .

Our translations are parametrized over a set of pairs of nominals $S \subseteq \text{NOM} \times \text{NOM}$, for NOM a countable, non empty set of atomic symbols different from the set PROP of propositional symbols. For a given relation-changing formula φ , we write its translation as a hybrid formula $\text{HTr}(\varphi)_S$. When translating a formula, S will originally be empty and it will store pairs of nominals that we will use to simulate the edges affected by the relation-changing operators we encounter during the translation.

Intuitively, given that the hybrid operators cannot affect the accessibility relation, we have to simulate the updates by recording possible affected edges using nominals and \downarrow . As a result, in all the relation-changing logics we will consider, the relation-changing formula $\diamond\varphi$ cannot be simply translated into a hybrid formula $\diamond\text{HTr}(\varphi)_S$, even though we have \diamond at our disposal in the hybrid language, because in the source language \diamond is interpreted over the updated accessibility relation. Instead, diamond-formulas need to be translated in a way that takes into account the updated accessibility relation.

Let $S \subseteq \text{NOM} \times \text{NOM}$ and $n \in \text{NOM}$. We define the translation HTr_S from $\mathcal{ML}(\langle \text{lsb} \rangle, \langle \text{gsb} \rangle)$ to $\mathcal{H}(\text{E}, \downarrow)$ as follows:

$$\begin{aligned} \text{HTr}_S(p) &= p \\ \text{HTr}_S(\neg\varphi) &= \neg\text{HTr}_S(\varphi) \\ \text{HTr}_S(\varphi \wedge \psi) &= \text{HTr}_S(\varphi) \wedge \text{HTr}_S(\psi) \\ \text{HTr}_S(\diamond\varphi)_S &= \downarrow n. \diamond(\neg \text{belongs}(n, S) \wedge \text{HTr}_S(\varphi)) \\ \text{HTr}_S(\langle \text{lsb} \rangle\varphi)_S &= \downarrow n. \diamond(\neg \text{belongs}(n, S) \wedge \downarrow m. \text{HTr}_{S \cup nm}(\varphi)) \\ \text{HTr}_S(\langle \text{gsb} \rangle\varphi)_S &= \downarrow k. \text{E} \downarrow n. \diamond(\neg \text{belongs}(n, S) \wedge \downarrow m. k: \text{HTr}_{S \cup nm}(\varphi)) \end{aligned}$$

where n, m and k are nominals not used in S , and $\text{belongs}(n, S) = \bigvee_{xy \in S} (y \wedge n:x)$.

A few comments are in order. First, given some model $\mathcal{M} = \langle W, R, V \rangle$ and some set $S \subseteq \text{NOM} \times \text{NOM}$, the formula $\downarrow n. \diamond(\neg \text{belongs}(n, S))$ is true at some state $w \in W$ if there exists some state v such that $(w, v) \in R$ and there is no pair of nominals $(x, y) \in S$ such that $(V(x), V(y)) = (w, v)$. Then, observe that the cases for $\langle \text{gsb} \rangle$ and $\langle \text{lsb} \rangle$ modify the set of deleted pairs in the recursive call to the translation, in both cases by adding an edge named nm . In the $\langle \text{gsb} \rangle$ case, n names the evaluation state of the formula, while in the $\langle \text{lsb} \rangle$ case, n names some state anywhere in the model.

When considering the translated formula $\text{HTr}_S(\varphi)$ and its truth in some model $\mathcal{M} = \langle W, R, V \rangle$, one question that may arise is what should be the initial valuation of the nominals that appear in $\text{HTr}_S(\varphi)$. By definition of models for hybrid logics, nominals must be true at some state. This is not problematic: in $\text{HTr}_S(\varphi)$, all nominals are bounded by the \downarrow operator, so the truth value of $\text{HTr}_S(\varphi)$ does not depend on their initial valuation. This enables us to talk about equivalence preservation of the translation: the same model \mathcal{M} can be used for φ and $\text{HTr}_S(\varphi)$; the interpretation of nominals is irrelevant for the truth of $\text{HTr}_S(\varphi)$.

Proposition 7 For $\mathcal{M} = \langle W, R, V \rangle$ a model, $w \in W$, and $\varphi \in \mathcal{ML}(\langle \text{lsb} \rangle, \langle \text{gsb} \rangle)$ we have $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, w \models \text{HTr}_\emptyset(\varphi)$.

Similar translations can be defined for $\langle \text{lbr} \rangle$, $\langle \text{gbr} \rangle$, $\langle \text{lsw} \rangle$ and $\langle \text{gsw} \rangle$. Naturally the global version of the dynamic operators need the expressive power of the E operator

to access an arbitrary point in the model. And the same is true for $\langle \text{lbr} \rangle$ as the bridged node can be previously unreachable. We can now exploit these equivalence preserving translations to obtain decidability results.

Interesting decidable fragments of hybrid logics have been found over time. First, let us consider restricting the satisfiability problem to certain classes of models. The following logics are known to be decidable over the indicated classes:

- $\mathcal{H}(E, \downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames) [33, 62] (this includes $(\mathbb{N}, <)$),
- $\mathcal{H}(E, \downarrow)$ over models with a single, transitive tree relation [62],
- $\mathcal{H}(E, \downarrow)$ over models with a single, $S5$, or complete relation [62],
- $\mathcal{H}(\cdot, \downarrow)$ over models with a single relation of bounded finite width [65]; as a corollary, also over finite models.

Corollary 1 *The satisfiability problem for all six concrete relation-changing modal logics over linear, transitive trees, $S5$, or complete frames is decidable.*

Corollary 2 *The satisfiability problem for $\mathcal{ML}(\langle \text{gsb} \rangle)$ and $\mathcal{ML}(\langle \text{lsw} \rangle)$ over models of bounded width is decidable.*

Curiously, these results mean that relation-changing modal logics are decidable over certain classes of models, even if the updates resulting from evaluating the dynamic operators yield models that *do not* belong to such class. For instance, these two facts are simultaneously true: sabotage logic is decidable on the class of $S5$ models, and deleting edges in an $S5$ model can yield a non- $S5$ model.

Decidable fragments can also be defined by imposing restrictions of how modalities can be nested. See further details in [10].

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

- (atomic harmony) for all $p \in \text{PROP}$, $w \in V(p)$ 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 $\mathcal{ML}(\langle f \rangle)$ -bisimilar (notation, $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$) if there is an $\mathcal{ML}(\langle f \rangle)$ -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^{lsb} we get the following conditions:

- (f^{lsb} -**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^{lsb} -**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.

Proposition 8 (Invariance) *Let f be a family of relation updates, then $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$ implies $\mathcal{M}, w \equiv_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$.*

The proof of this result is fairly standard, and we only discuss the case of the dynamic operator. Let $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$, and Z such that $(w, R)Z(w', R')$. Suppose $\langle W, R, V \rangle, w \models \langle f \rangle \varphi$. 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 $\langle W', R', V' \rangle, w' \models \langle f \rangle \varphi$. For the other direction use (*f*-zag).

Proposition 8 proves that bisimilarity defines an equivalence relation that is at 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 [18]. A well known result establishes that ω -saturated models are a Hennessy-Milner class for many modal languages (see [23] for details). We define a notion of ω -saturation for relation-changing modal logics and show that the corresponding class of models is Hennessy-Milner. We again refer to [9] for proofs.

Definition 10 (*f*-saturation) Let $\mathcal{M} = \langle W, R, V \rangle$ be a model, $X \subseteq W \times 2^{W^2}$, f a family of relation updates and Σ a set of $\mathcal{ML}(\langle f \rangle)$ -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 $(s, S) \in \text{Img}(f_W) \cup \{(w, R) \mid w \in W\}$ 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 requires ω -saturation in each possible updated model, and also with respect to the set of possible relation updates in each state.

Proposition 9 *Let f be a family of relation updates, and let $\mathcal{M}, w, \mathcal{M}', w'$ be two *f*-saturated models. Then $\mathcal{M}, w \equiv_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$ implies $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$.*

The notion of model unfolding we introduced in Section 3 also gives rise to Hennessy-Milner classes.

Proposition 10 *Let \mathcal{M} and \mathcal{M}' be two models and let f be a family of relation updates. 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 $w' \in W'$, $\mathcal{M}, w \equiv_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$ implies $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$.*

Corollary 3 *Let \mathcal{M}, w and \mathcal{M}', w' be finite pointed models. Let f be a family of relation updates. Then $\mathcal{M}, w \equiv_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$ implies $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)} \mathcal{M}', w'$.*

Another classical notion investigated in modal logics, is n -bisimulations, i.e., bisimulations up to certain modal depth. It is easy to show that for relation-changing modal logics, n -bisimilarity for all n and modal equivalence coincide.

Definition 11 (n -bisimulations) Let $\mathcal{M}, \mathcal{M}'$ two models and w and w' states of \mathcal{M} and \mathcal{M}' .

We say that w and w' are n -bisimilar (notation $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)}^n \mathcal{M}', w'$), if there exists a sequence of binary relations $Z_n \subseteq \dots \subseteq Z_0$ with the following properties (for $i + 1 \leq n$):

1. $(w, R)Z_n(w', R')$
2. If $(v, S)Z_0(v', S')$ then v and v' agree on all propositional symbols.
3. If $(v, S)Z_{i+1}(v', S')$ and Svu then there is u' with $S'v'u'$ and $(u, S)Z_i(u', S')$.
4. If $(v, S)Z_{i+1}(v', S')$ and $S'v'u'$ then there is u with Svu and $(u, S)Z_i(u', S')$.
5. If $(v, S)Z_{i+1}(v', S')$ and $(u, T) \in f_W(u, S)$ then there is $(u', T') \in f_{W'}(v', S')$ such that $(u, T)Z_i(u', T')$.
6. If $(v, S)Z_{i+1}(v', S')$ and $(u', T') \in f_{W'}(v', S')$ then there is $(u, T) \in f_W(v, S)$ such that $(u, T)Z_i(u', T')$.

Intuitively, if $\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)}^n \mathcal{M}', w'$ then w and w' bisimulate up to modal depth n .

Proposition 11 *Let \mathcal{M} and \mathcal{M}' be two models and let f be a family of relation updates. Then for every w in \mathcal{M} and w' in \mathcal{M}' the following are equivalent:*

$$\mathcal{M}, w \Leftrightarrow_{\mathcal{ML}(\langle f \rangle)}^n \mathcal{M}', w' \text{ iff } \mathcal{M}, w \equiv_{\mathcal{ML}(\langle f \rangle)}^n \mathcal{M}', w',$$

where $\mathcal{M}, w \equiv_{\mathcal{ML}(\langle f \rangle)}^n \mathcal{M}', w'$ means that w and w' satisfy the same $\mathcal{ML}(\langle f \rangle)$ -formulas of modal depth at most n .

Adding relation-changing modal operators to the basic modal logic increases its expressive power. A basic result for \mathcal{ML} states 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 [23]. 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 12 $\mathcal{ML}(\langle f \rangle)$ does not have the tree model property, for $\langle f \rangle \in \{\langle lsb \rangle, \langle lbr \rangle, \langle lsw \rangle, \langle gsb \rangle, \langle gbr \rangle, \langle gsw \rangle\}$.

To establish Proposition 12 we show formulas that ensure that the accessibility relation is not a tree. For $\langle gsb \rangle$, the result has already been proved in [52], for $\langle lsw \rangle$ in [8] and for $\langle lsb \rangle$ and $\langle lbr \rangle$ in [6]. The remaining cases are established in [9]. Suppose the following formulas hold at some state w in a model:

1. $\diamond\diamond\top \wedge [lsb]\Box\perp$, then w is reflexive;
2. $\diamond\diamond\top \wedge [gsb]\Box\perp$, then w is reflexive;
3. $\Box\perp \wedge \langle lbr \rangle\Box\perp$, then w and some different state v are unconnected;
4. $\Box\perp \wedge \langle gbr \rangle\Box\perp$, then w and some different state v are unconnected;
5. $p \wedge (\bigwedge_{1 \leq i \leq 3} \Box^i \neg p) \wedge \langle lsw \rangle \diamond\diamond p$, then w has a reflexive successor;
6. $\Box\perp \wedge \langle gsw \rangle \diamond\top$, then w has an incoming edge.

We briefly discuss the case for $\mathcal{ML}(\langle gsb \rangle)$. $\varphi = \diamond\diamond\top \wedge [lsb]\Box\perp$ is true at w in a model only if w is reflexive. 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 $[lsb]\Box\perp$ 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.

As the six concrete 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.

In the rest of the section, we will decide whether the logics obtained by extended the basic modal logic with $\langle lsb \rangle$, $\langle lbr \rangle$, $\langle lsw \rangle$, $\langle gsb \rangle$, $\langle gbr \rangle$ and $\langle gsw \rangle$ 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 12 ($\mathcal{L} \leq \mathcal{L}'$) We say that \mathcal{L}' is at least as expressive as \mathcal{L} (notation $\mathcal{L} \leq \mathcal{L}'$) if there is a function MTr 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 $\mathcal{M} \models_{\mathcal{L}} \varphi$ iff $\mathcal{M} \models_{\mathcal{L}'} \text{MTr}(\varphi)$. \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 13 Consider any pair of different logics in the following list: $\mathcal{ML}(\langle lsb \rangle)$, $\mathcal{ML}(\langle lbr \rangle)$, $\mathcal{ML}(\langle lsw \rangle)$, $\mathcal{ML}(\langle gsb \rangle)$, $\mathcal{ML}(\langle gbr \rangle)$ and $\mathcal{ML}(\langle gsw \rangle)$; then their expressive power is incomparable, except perhaps for $\mathcal{ML}(\langle lsw \rangle)$ and $\mathcal{ML}(\langle gsw \rangle)$.





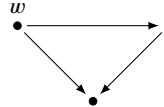

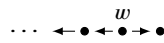
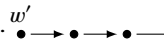
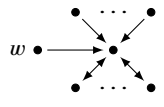
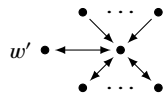


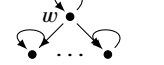

\mathcal{M}, w	\mathcal{M}', w'	Differentiated by	Bisimilar for
		$\langle \text{lbr} \rangle \langle \text{lbr} \rangle \top$ $\langle \text{gbr} \rangle \langle \text{gbr} \rangle \top$ $\langle \text{gsb} \rangle \top$ $\langle \text{gsw} \rangle \top$	$\mathcal{ML}(\langle \text{lsb} \rangle)$ $\mathcal{ML}(\langle \text{lsw} \rangle)$
		$\langle \text{lsb} \rangle \diamond \top$ $\langle \text{gsb} \rangle \diamond \top$	$\mathcal{ML}(\langle \text{lsw} \rangle)$ $\mathcal{ML}(\langle \text{lbr} \rangle)$ $\mathcal{ML}(\langle \text{gsw} \rangle)$ $\mathcal{ML}(\langle \text{gbr} \rangle)$
		$\langle \text{lsw} \rangle \diamond \diamond \diamond \square \perp$ $\diamond \langle \text{gsw} \rangle \diamond \diamond \diamond \square \perp$ $\langle \text{lbr} \rangle \langle \text{lbr} \rangle \top$ $\langle \text{gbr} \rangle^6 \langle \text{gbr} \rangle \top$	$\mathcal{ML}(\langle \text{gsb} \rangle)$ $\mathcal{ML}(\langle \text{lsb} \rangle)$
		$\langle \text{lsw} \rangle \diamond \square \perp$ $\langle \text{gsw} \rangle \square \perp$	$\mathcal{ML}(\langle \text{lbr} \rangle)$ $\mathcal{ML}(\langle \text{gbr} \rangle)$
		$\langle \text{lsb} \rangle \diamond \square \perp$	$\mathcal{ML}(\langle \text{gsb} \rangle)$
		$\langle \text{lbr} \rangle^3 \top$ $\langle \text{gbr} \rangle^3 \top$	$\mathcal{ML}(\langle \text{gsw} \rangle)$
		$\langle \text{lbr} \rangle \top$	$\mathcal{ML}(\langle \text{gbr} \rangle)$

Fig. 1 Bisimilar models and distinguishing formulas.

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 $\mathcal{ML}(\langle \text{gbr} \rangle) \not\leq \mathcal{ML}(\langle \text{lbr} \rangle)$ 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 $\mathcal{ML}(\langle \text{lbr} \rangle)$ -bisimilar and there is an $\mathcal{ML}(\langle \text{gbr} \rangle)$ -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 $\langle \text{gbr} \rangle \square \diamond \top$ is true at \mathcal{M}, w and false at \mathcal{M}, v . \mathcal{M}, w and \mathcal{M}, v are $\mathcal{ML}(\langle \text{lbr} \rangle)$ -bisimilar.

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 $\mathcal{ML}(\langle \text{lsw} \rangle)$ and $\mathcal{ML}(\langle \text{gsw} \rangle)$ remains open (in particular it is unknown if there is a translation from $\mathcal{ML}(\langle \text{lsw} \rangle)$ to $\mathcal{ML}(\langle \text{gsw} \rangle)$ or not), but we conjecture that their expressive power is also incomparable.

5 Complexity

In this section we investigate how hard it is to reason with relation-changing logics. When investigating complexity, it is difficult to provide general results. Instead we will show two approaches that we used to establish the complexity for the model checking problem and the satisfiability problem for the six concrete relation-changing logics we are using as running examples.

5.1 The Model Checking Problem

The model checking problem is defined as follows: given a pointed model \mathcal{M}, w and a formula φ , decide whether $\mathcal{M}, w \models \varphi$. We will argue that the complexity of this reasoning problem is PSPACE -complete for our six relation-changing logics. In all cases we use a similar argument: hardness proofs are done by encoding the satisfiability problem of Quantified Boolean Formulas (QBF) [58] 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. For the upper bound, we describe a model checking algorithm running in polynomial space.

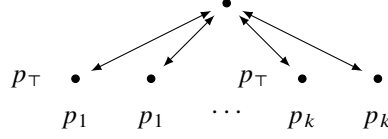
PSPACE -hardness for global sabotage was already proved in [53, 52], but we provide here a more direct proof. For $\langle \text{lsw} \rangle$, proof was given in [8], and for $\langle \text{lsb} \rangle$ and $\langle \text{lbr} \rangle$ proof was given in [6]. The complete result for the six concrete logics we are considering was first introduced in [32].

Proposition 14 *For $\langle f \rangle \in \{\langle \text{lsb} \rangle, \langle \text{lbr} \rangle, \langle \text{lsw} \rangle, \langle \text{gsb} \rangle, \langle \text{gbr} \rangle, \langle \text{gsw} \rangle\}$, model checking for $\mathcal{ML}(\langle f \rangle)$ is PSPACE -hard.*

To establish Proposition 14 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 $\langle \text{lsw} \rangle$. For the other operators a similar strategy establishes the result.

Let α be a QBF formula with variables $\{x_1, \dots, x_k\}$. 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 $\{p_\top, p_1, \dots, p_k\}$, where:

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



Let $(\cdot)'$ be the following linear translation from QBF to $\mathcal{ML}(\langle \text{lsw} \rangle)$:

$$\begin{aligned}
(\exists x_i. \alpha)' &= \langle \text{lsw} \rangle (p_i \wedge \diamond(\alpha)') \\
(x_i)' &= \neg \diamond(p_i \wedge p_\top) \\
(\neg \alpha)' &= \neg(\alpha)' \\
(\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'.
\end{aligned}$$

It remains to prove that α is satisfiable iff $\mathcal{M}_k, w \models (\alpha)'$. Let us write $v \models_{\text{qbf}} \alpha$ if valuation $v : \{x_1, \dots, x_k\} \rightarrow \{0, 1\}$ satisfies α . For a model \mathcal{M} with relation R we define $v_R : \{x_1, \dots, x_k\}$ as $v_R(x_i) = 1$ iff $(w, w_i^1) \notin R$. In the present case, this happens when 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 $\mathcal{M}, w \models \langle \text{lsw} \rangle (p_i \wedge \diamond(\gamma)')$ iff $\mathcal{M}, w \models (\exists x_i. \gamma)'$.

See [9] for details for the other logics. We now discuss a matching upper bound.

Proposition 15 *Model checking for $\mathcal{ML}(\{\langle \text{lsb} \rangle, \langle \text{lbr} \rangle, \langle \text{lsw} \rangle, \langle \text{gsb} \rangle, \langle \text{gbr} \rangle, \langle \text{gsw} \rangle\})$ is in $PSPACE$.*

To prove the upper bound, given a pointed model \mathcal{M}, w and a formula φ we present a recursive algorithm $\text{CHECK}(\mathcal{M}, w, \varphi)$ 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 $(\diamond\psi)$, $\text{CHECK}(\mathcal{M}, v, \psi)$ 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 $\langle f \rangle \psi$ involves building a certain number of pointed models \mathcal{M}', v' to compute $\text{CHECK}(\mathcal{M}', v', \psi)$. 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 $\langle \text{lsb} \rangle$, $\langle \text{lbr} \rangle$, $\langle \text{lsw} \rangle$, $\langle \text{gsb} \rangle$, $\langle \text{gbr} \rangle$ and $\langle \text{gsw} \rangle$ generate a polynomial amount of models so the counter would only use logarithmic space).

Corollary 4 *For $S \subseteq \{\langle \text{lsb} \rangle, \langle \text{lbr} \rangle, \langle \text{lsw} \rangle, \langle \text{gsb} \rangle, \langle \text{gbr} \rangle, \langle \text{gsw} \rangle\}$, model checking for any of the logics $\mathcal{ML}(S)$ is PSPACE-complete.*

5.2 The Satisfiability Problem

We now turn to the satisfiability problem. As we discussed in Section 4 relation-changing logics are, potentially, highly expressive. In [32] it is shown that the six concrete relation-changing operators are expressive enough to force infinite models.

Proposition 16 *$\mathcal{ML}(\langle f \rangle)$ does not have finite model property, for $\langle f \rangle \in \{\langle \text{lsb} \rangle, \langle \text{lbr} \rangle, \langle \text{lsw} \rangle, \langle \text{gsb} \rangle, \langle \text{gbr} \rangle, \langle \text{gsw} \rangle\}$.*

Further, in [12] we prove that the satisfiability problem of these six logics is undecidable, by an encoding of the satisfiability problem of memory logics [3, 57].

Memory logics are modal logics that can *store* the current state of evaluation into a memory and *check* whether the current state belongs to this memory. Its syntax and semantics are extensions of the syntax and semantics of the basic modal logic \mathcal{ML} . The memory is a subset of the domain of the model. $\mathcal{ML}(\mathbb{R}, \mathbb{K})$ is the memory logic that extends \mathcal{ML} with the operators \mathbb{R} and \mathbb{K} , which stand for “remember” and “known”, respectively. We briefly recap the syntax and semantics of these logics in the next paragraphs.

Definition 13 (Syntax and Semantics of $\mathcal{ML}(\mathbb{R}, \mathbb{K})$) Let PROP be a countable, infinite set of propositional symbols. The set FORM of formulas of $\mathcal{ML}(\mathbb{R}, \mathbb{K})$ over PROP is defined as:

$$\text{FORM} ::= p \mid \mathbb{K} \mid \neg\varphi \mid \varphi \wedge \psi \mid \diamond\varphi \mid \mathbb{R}\varphi,$$

where $p \in \text{PROP}$ and $\varphi, \psi \in \text{FORM}$.

A model $\mathcal{M} = \langle W, R, V, S \rangle$ is a relational model equipped with a set $S \subseteq W$ called the *memory*. Let w be a state in W . The inductive definition of satisfiability for the cases specific to memory logic is:

$$\begin{aligned} \langle W, R, V, S \rangle, w \models \textcircled{\mathbb{R}}\varphi &\text{ iff } \langle W, R, V, S \cup \{w\} \rangle, w \models \varphi \\ \langle W, R, V, S \rangle, w \models \textcircled{\mathbb{K}} &\text{ iff } w \in S. \end{aligned}$$

The remaining cases coincide with the semantics of \mathcal{ML} , and do not involve the memory. An $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$ -formula φ is *satisfiable* if there are a model $\mathcal{M} = \langle W, R, V, \emptyset \rangle$ and $w \in W$ such that $\mathcal{M}, w \models \varphi$. The empty initial memory ensures that no state of the model satisfies the unary predicate $\textcircled{\mathbb{K}}$ unless a formula $\textcircled{\mathbb{R}}\psi$ has previously been evaluated there.

Multimodal memory logic is shown to be undecidable in [13]. In [12], we strengthen this result showing that undecidability holds also in the monomodal case. We adapt the encoding of [56]: the $\mathbb{N} \times \mathbb{N}$ -tiling problem (i.e., the problem of tiling the plane given a set of tile types [27]) is encoded into $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$. In [54] it is shown that multimodal $\mathcal{ML}(\langle \text{gsb} \rangle)$ is undecidable via a reduction of the Post Correspondence Problem. The present proof extends this result to the monomodal case via a reduction of the satisfiability problem of the memory logic $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$. Here we only discuss the case of $\mathcal{ML}(\langle \text{gsb} \rangle)$.

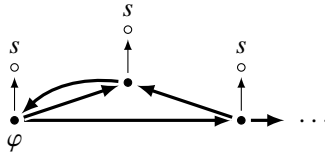
We will define a computable, satisfiability preserving translation $\tau_{\langle \text{gsb} \rangle}(\varphi)$ from formulas in $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$ to $\mathcal{ML}(\langle \text{gsb} \rangle)$. One piece of data needed to build $\tau_{\langle \text{gsb} \rangle}(\varphi)$ is the modal depth of the input formula ($\text{md}(\varphi)$). For a formula φ in $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$, $\text{md}(\varphi)$ is the maximum nesting of the \diamond and \square modalities in φ , disregarding the dynamic operators.

Define $\tau_{\langle \text{gsb} \rangle}(\varphi) = \text{Struct}_{\langle \text{gsb} \rangle}(\varphi) \wedge \text{MTr}_{\langle \text{gsb} \rangle}(\varphi)$. $\text{Struct}_{\langle \text{gsb} \rangle}(\varphi)$ forces a needed structure in the model, and $\text{MTr}_{\langle \text{gsb} \rangle}(\varphi)$ uses this structure to encode the memory operators. The notation $\square^i \varphi$ is defined as $\square^0 \varphi = \varphi$ and $\square^{n+1} \varphi = \square \square^n \varphi$.

$$\text{Struct}_{\langle \text{gsb} \rangle}(\varphi) = \neg s \wedge \bigwedge_{0 \leq i \leq \text{md}(\varphi)} \square^i (\neg s \rightarrow (\diamond s \wedge \langle \text{gsb} \rangle \neg \diamond s)).$$

$$\begin{aligned} \text{MTr}_{\langle \text{gsb} \rangle}(p) &= p \quad \text{for } p \in \text{PROP appearing in } \varphi \\ \text{MTr}_{\langle \text{gsb} \rangle}(\textcircled{\mathbb{K}}) &= \neg \diamond s \\ \text{MTr}_{\langle \text{gsb} \rangle}(\neg \psi) &= \neg \text{MTr}_{\langle \text{gsb} \rangle}(\psi) \\ \text{MTr}_{\langle \text{gsb} \rangle}(\psi \wedge \chi) &= \text{MTr}_{\langle \text{gsb} \rangle}(\psi) \wedge \text{MTr}_{\langle \text{gsb} \rangle}(\chi) \\ \text{MTr}_{\langle \text{gsb} \rangle}(\diamond \psi) &= \diamond (\neg s \wedge \text{MTr}_{\langle \text{gsb} \rangle}(\psi)) \\ \text{MTr}_{\langle \text{gsb} \rangle}(\textcircled{\mathbb{R}}\psi) &= \langle \text{gsb} \rangle (\neg \diamond s \wedge \text{MTr}_{\langle \text{gsb} \rangle}(\psi)). \end{aligned}$$

Up to the depth indicated by $\text{md}(\varphi)$, $\text{Struct}_{\langle \text{gsb} \rangle}(\varphi)$ adds to every state a transition to some state where s holds. It is as if each state of the input model had a flag that could be turned on to identify the state. The idea is illustrated in the following image.



Thus, remembering some state is simulated in $\text{MTr}_{\langle \text{gsb} \rangle}(\textcircled{\mathbb{R}})$ by deleting the edge between the state and its s -successor. For $\text{MTr}_{\langle \text{gsb} \rangle}(\textcircled{\mathbb{K}})$, we just check whether the current state has an s -successor.

There is one subtle difference between the $\textcircled{\mathbb{R}}$ operator and relation-changing operators like $\langle \text{gsb} \rangle$. While evaluating $\langle \text{gsb} \rangle \varphi$ always results in a change in the model, $\textcircled{\mathbb{R}} \varphi$ can leave the memory unchanged if the current state of evaluation is already memorized. We can ignore this difference by observing that any $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$ -formula can be rewritten into an equivalent formula where every occurrence of $\textcircled{\mathbb{R}}$ is “proper,” in the sense that it actually modifies the memory.

Definition 14 An $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$ -formula φ is in *proper normal form* (PNF) if every occurrence of the $\textcircled{\mathbb{R}}$ operator in φ occurs as $(\neg \textcircled{\mathbb{K}} \wedge \textcircled{\mathbb{R}} \psi) \vee (\textcircled{\mathbb{K}} \wedge \psi)$.

We assume that memory logic formulas are in PNF in the structural inductive proof of the next proposition.

Proposition 17 Let φ be an $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$ -formula in PNF that does not contain the propositional symbol s . Then, φ is satisfiable iff $\tau_{\langle \text{gsb} \rangle}(\varphi)$ is satisfiable.

Similar satisfiability translations of $\mathcal{ML}(\textcircled{\mathbb{R}}, \textcircled{\mathbb{K}})$ into each of the remaining concrete relation-changing logics we are considering can be defined, leading to corresponding undecidability results. Full details are provided in [12].

6 Axiomatizations

The last theme we want to address in this chapter is axiomatizations. Hilbert style axiomatizations for different dynamic logics can be found in the literature, but in general they all pose a challenge. Many dynamic logics *are not closed under uniform substitution*. I.e., it can be the case that φ is a tautology, but the formula obtained by uniformly replacing all instances of a propositional symbol p in φ by an arbitrary formula ψ might not be a tautology.

Consider, for example, the valid formula $(p \wedge \diamond \top) \rightarrow \langle \text{gsb} \rangle p$. It is easy to prove that if we uniformly replace p by $\langle \text{gsb} \rangle p$ the resulting formula $(\langle \text{gsb} \rangle p \wedge \diamond \top) \rightarrow \langle \text{gsb} \rangle \langle \text{gsb} \rangle p$ is not valid.

As it turns out, the additional expressivity provided by nominals and the satisfiability operator $:$ can be used to solve the problem of axiomatizing the behavior of $\langle \text{gsb} \rangle$. A very elegant solution to axiomatize the extension of the hybrid logic $\mathcal{H}(:)$ with the $\langle \text{gsb} \rangle$ operator is presented in [69] using ideas first introduced in [71].

Once $\langle \text{gsb} \rangle$ is added to the language of $\mathcal{H}(:)$ the following operator can be defined:

$$\langle \text{gsb} \rangle_{ab} \varphi := (a : \diamond b \wedge \langle \text{gsb} \rangle (a : \neg \diamond b \wedge \varphi)) \vee (a : \neg \diamond b \wedge \varphi)$$

where a, b are nominals. Let $\mathbf{ab} = a_1 b_1 \cdots a_n b_n$ be a finite sequence of pairs of nominals, we write $\langle \text{gsb} \rangle_{\mathbf{ab}} \varphi$ as a shorthand for $\langle \text{gsb} \rangle_{a_1 b_1} \cdots \langle \text{gsb} \rangle_{a_n b_n} \varphi$. $\langle \text{gsb} \rangle_{\mathbf{ab}} \varphi$ is just φ when $n = 0$.

Axioms schemes and rules for $\mathcal{H}(\cdot)$

All tautologies of classical propositional logic, plus Modus Ponens (CPL)

$$(K_{\Box}) \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$$

$$(Nec_{\Box}) \frac{\varphi}{\Box\varphi}$$

Axioms schemes for :

$$\begin{array}{ll} a:(\varphi \rightarrow \psi) \rightarrow (a:\varphi \rightarrow a:\psi) & a \wedge a:\varphi \rightarrow \varphi \\ a:\varphi \leftrightarrow \neg a:\neg\varphi & a \wedge \varphi \rightarrow a:\varphi \\ a:a & a:b \leftrightarrow b:a \\ a:b \wedge b:\varphi \rightarrow a:\varphi & b:a:\varphi \leftrightarrow a:\varphi \\ \diamond a:\varphi \rightarrow a:\varphi & \end{array}$$

Inference rules for $\mathcal{H}(\cdot)$

$$(Nec.) \frac{\varphi}{a:\varphi}$$

$$(Name) \frac{c \rightarrow \varphi}{c \rightarrow \varphi} \quad (c \text{ a nominal not in } \varphi)$$

$$(Paste) \frac{a:\diamond b \wedge b:\varphi \rightarrow \delta}{a:\diamond\varphi \rightarrow \delta} \quad (b \text{ a nominal not in } \varphi, \delta \text{ and } a \text{ are distinct from } b)$$

Normal axiom schema and inference rule for $[gsb]$

$$(K_{[gsb]}) [gsb](\varphi \rightarrow \psi) \rightarrow ([gsb]\varphi \rightarrow [gsb]\psi)$$

$$(Nec_{[gsb]}) \frac{\varphi}{[gsb]\varphi}$$

Recursion Axioms for $\langle gsb \rangle_{ab}$

$$\langle gsb \rangle_{ab} c \leftrightarrow c \quad (c \in \text{NOM})$$

$$\langle gsb \rangle_{ab} p \leftrightarrow p \quad (p \in \text{PROP})$$

$$\langle gsb \rangle_{ab} \neg\varphi \leftrightarrow \neg\langle gsb \rangle_{ab} \varphi$$

$$\langle gsb \rangle_{ab} (\varphi \wedge \psi) \leftrightarrow (\langle gsb \rangle_{ab} \varphi \wedge \langle gsb \rangle_{ab} \psi)$$

$$\langle gsb \rangle_{ab} c:\varphi \leftrightarrow c:\langle gsb \rangle_{ab} \varphi$$

$$\langle gsb \rangle_{ab} \diamond\varphi \leftrightarrow ((a \wedge \diamond(\neg b \wedge \langle gsb \rangle_{ab} \varphi)) \vee (\neg a \wedge \diamond\langle gsb \rangle_{ab} \varphi))$$

Inference rule for $\langle gsb \rangle$ and $\langle gsb \rangle_{ab}$

$$(B\text{-Mix}) \frac{c:\langle gsb \rangle_{ab} (a_{n+1}:\diamond b_{n+1} \wedge \langle gsb \rangle_{a_{n+1}b_{n+1}} \varphi) \rightarrow \theta}{c:\langle gsb \rangle_{ab} \langle gsb \rangle \varphi \rightarrow \theta}$$

where $n \geq 0$; nominals a_{n+1}, b_{n+1} are distinct from c and other nominals in $\langle gsb \rangle_{ab}$ and do not occur in neither φ nor θ .

Fig. 2 The Hilbert-style proof system for $\mathcal{H}(\cdot, \langle gsb \rangle)$ introduced in [69].

Informally, $\langle gsb \rangle_{ab} \varphi$ says that after cutting a (possibly nonexistent) edge between the states named a and b , φ will hold. The first disjunct describes the effects of actually cutting the edge when it exists, while the second takes care of the case when the edge does not exist. In turn, $\langle gsb \rangle_{ab} \varphi$ stands for a finite sequence of such cuts, before testing for φ .

Interestingly, it is now possible to define recursion axioms for $\langle gsb \rangle_{ab}$ similar to the ones used to axiomatize Public Announcement Logic (PAL) [59, 39]. Moreover, the inference rule (B-Mix) can then be introduced to describe the interaction between $\langle gsb \rangle$ and $\langle gsb \rangle_{ab}$. See Figure 2 for the complete axiomatic system.

Notice that the resulting system is not closed under uniform substitution. While in most cases substitutions are allowed by the axiom schemes involved, the base cases of the recursion axioms are sound only over atomic symbols on $\text{NOM} \cup \text{PROP}$.

The completeness proof then proceeds by means of a Henkin model (i.e., a model where all diamond modalities are witnessed using nominals, see [23, 25]).

Definition 15 (Named, pasted, B-mixed) A set of formulas Γ is

1. *named* if it contains a nominal,
2. *pasted* if $a:\diamond\varphi \in \Gamma$ implies that there is some nominal b such that the formula $a:\diamond b \wedge b:\varphi \in \Gamma$, and
3. *B-mixed* if $c:\langle\text{gsb}\rangle_{ab}\langle\text{gsb}\rangle\varphi \in \Gamma$ implies that $c:\langle\text{gsb}\rangle_{ab}(a_{n+1}:\diamond b_{n+1} \wedge \langle\text{gsb}\rangle_{a_{n+1}b_{n+1}}\varphi) \in \Gamma$ for some nominals a_{n+1} and b_{n+1} .

Then, a strengthened Lindenbaum Lemma can be established, proving that every consistent set of formulas in $\mathcal{H}(\cdot, \langle\text{gsb}\rangle)$ can be extended to a maximal consistent set of formulas which is named, pasted and B-mixed in a language extended with an additional set of nominals. All details can be found in [69].

In [31], an approach similar to the one presented above is used to axiomatize other relation-changing operators.

7 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 [29], 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 [63, 64], which is an extension of traditional doxastic logic (see [45]) 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 [17]). 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 agents. The epistemic models representing the knowledge of 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 iterated, revocable and higher-order revision can be formalized in this setting in a natural way (see, e.g., [15, 73, 72]).

Sabotage logic was proposed as a formalism for reasoning about formal learning theory [40]. Learning can be seen as a game with two players, Teacher and Learner, where Learner changes her information state through a step-by-step process. The process is successful if she eventually reaches an information state describing the real state of affairs. The information that Teacher provides can be interpreted as feedback about Learner's conjectures about the current state of affairs, allowing her to discard inconsistent hypotheses. From this game-theoretical perspective, the interaction between Teacher and Learner can be modeled using operators similar to those of sabotage logic.

In [16], 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., [28, 30]) and public announcements (see, e.g., [59, 49]), as well as logics of preference modification [70]. 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 [50, 51], 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 [59, 39], in which a statement eliminates access to all epistemic possibilities in which the statement does not hold. It is similar in spirit to the arrow pre-condition language proposed in [60], as well as other works about access elimination (see e.g., [28, 16, 68]). 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 [17, 74], 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 chapter, we presented 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., [20, 70, 19, 47]) 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 have shown in this chapter that the general framework we defined includes logics which are strictly more expressive than the basic modal logic.

8 Conclusions

In this chapter we introduced a framework to define relation-changing modal operators. It is based on the notion of relation updates 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 [68] and other variants investigated in, e.g., [6, 8, 61]. 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 [59, 38, 67, 74].

We introduced six different concrete dynamic modal operators with both local and global effects which can add, delete and swap 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 refer 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 [61, 8], it was shown that $\mathcal{ML}(\langle \text{!sw} \rangle)$ and $\mathcal{ML}(\langle \text{!gsb} \rangle)$ 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 convert models and formulas to the basic modal logic with two accessibility relations, over a particular class of models. Finally, we show translations into hybrid logics, and took advantage of previously known results to define model classes over which decidable logics obtain.

While in this chapter we considered only translations into first- and second-order logics, it would be interesting to consider other possibilities. For example, it would be rewarding to consider cases where the translation falls into decidable fragments of first-order logic, like the guarded fragment [2] or the two variable fragment [42]. We could also consider well behaved fragments of second-order logic, like μFO , i.e., the fixed-point extension of first-order logic [43].

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 under Bisimulation” result for relation-changing modal logics, and the existence of Hennessy-Milner model classes. We used this definition of bisimulations 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 relation updates to obtain general expressivity results. We have established that the six concrete logics introduced are pairwise incomparable in terms of expressive power, except for $\mathcal{ML}(\langle \text{isw} \rangle)$ and $\mathcal{ML}(\langle \text{gsw} \rangle)$. 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, while their satisfiability problem is undecidable. Proofs for the complexity of model checking 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. Proofs of the undecidability of the satisfiability problem are based on the encoding of the undecidable satisfiability problem for the memory logic $\mathcal{ML}(\mathbb{R}, \mathbb{K})$. All six concrete logics investigated in this chapter have an undecidable satisfiability problem. It would be interesting to find ways leading to decidability. One option could be to carefully design dynamic modal operators like the ones investigated in [4]. Another option would be to restrict the application of operators, for example, by means of protocols. These kind of logics have been investigated in, e.g., [48, 21].

Finally we briefly discussed axiomatizations. As a consequence of their high expressive power, in most cases it is not possible to define reduction axioms to basic modal logic, such as it is done for several dynamic epistemic logics. We discussed how [71] provides an elegant Hilbert-style axiomatization for an extension of the hybrid modal logic $\mathcal{H}(\cdot)$ with the global sabotage operator $\langle \text{gsb} \rangle$. This approach can be extended to other relation-changing operators, as was done in [31].

This chapter 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 [14]). Investigating succinctness question for these languages, as is done for different dynamic epistemic logics in, e.g., [55, 34], 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 [35, 32, 5, 4]. 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.

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. H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
3. C. Areces. Hybrid Logics: The Old and the New. In *Proceedings of LogKCA-07*, pages 15–29, 2007.
4. C. Areces, H. van Ditmarsch, R. Fervari, B. Maubert, and F. Schwarzentruber. Copy and remove as dynamic operators. *Journal of Applied Non-Classical Logics*, 31(3-4):181–220, 2021.
5. 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, *Proceedings of the Workshop on Logic, Language, Information and Computation (WoLLIC 2014)*, volume 8652 of *LNCS*, pages 51–65. Springer, 2014.
6. C. Areces, R. Fervari, and G. Hoffmann. Moving arrows and four model checking results. In L. Ong and R. de Queiroz, editors, *Proceedings of the Workshop on Logic, Language, Information and Computation (WoLLIC 2012)*, volume 7456 of *LNCS*, pages 142–153. Springer Berlin Heidelberg, 2012.
7. C. Areces, R. Fervari, and G. Hoffmann. Tableaux for relation-changing modal logics. In *Proceedings of Frontiers of Combining Systems (FROCCOS 2013)*. Springer, 2013.
8. C. Areces, R. Fervari, and G. Hoffmann. Swap logic. *Logic Journal of the IGPL*, 22(2):309–332, 2014.
9. C. Areces, R. Fervari, and G. Hoffmann. Relation-changing modal operators. *Logic Journal of the IGPL*, 23(4):601–627, 2015.
10. C. Areces, R. Fervari, G. Hoffmann, and M. Martel. Relation-changing logics as fragments of hybrid logics. *Electronic Proceedings in Theoretical Computer Science*, 226:16–29, 2016. Proceedings of the 7th International Symposium on Games, Automata, Logics and Formal Verification.
11. C. Areces, R. Fervari, G. Hoffmann, and M. Martel. Undecidability of relation-changing modal logics. In *Dynamic Logic. New Trends and Applications*, pages 1–16. Springer, 2017.
12. C. Areces, R. Fervari, G. Hoffmann, and M. Martel. Satisfiability for relation-changing logics. *Journal of Logic and Computation*, 28(7):1443–1470, 2018.
13. C. Areces, D. Figueira, S. Figueira, and S. Mera. The Expressive Power of Memory Logics. *The Review of Symbolic Logic*, 4(2):290–318, 2011.
14. C. Areces and B. ten Cate. Hybrid logics. In Blackburn et al. [26], pages 821–868.
15. G. Aucher. A combined system for update logic and belief revision. Master’s thesis, ILLC, Universiteit van Amsterdam, 2003.
16. 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.
17. 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.
18. J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Mathematical Institute, University of Amsterdam, 1977.
19. J. van Benthem. Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155, 2007.
20. J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.
21. J. van Benthem, J. Gerbrandy, T. Hoshi, and E. Pacuit. Merging frameworks for interaction. *Journal of Philosophical Logic*, 38(5):491–526, 2009.
22. P. Blackburn and J. van Benthem. Modal logic: A semantic perspective. In Blackburn et al. [26], pages 1–84.

23. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 2001.
24. P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4:251–272, 1995.
25. P. Blackburn and B. ten Cate. Pure extensions, proof rules, and hybrid axiomatics. *Studia Logica*, 84(2):277–322, 2006.
26. P. Blackburn, F. Wolter, and J. van Benthem, editors. *Handbook of Modal Logics*. Elsevier, 2006.
27. P. van Emde Boas. The convenience of tilings. In *Complexity, Logic, and Recursion Theory*, pages 331–363. Marcel Dekker Inc, 1997.
28. G. Renardel de Lavalette. Changing modalities. *Journal of Logic and Computation*, 14(2):253–278, 2004.
29. M. de Rijke. Meeting some neighbours. *Logic and information flow*, pages 170–195, 1994.
30. H. van Ditmarsch, W. van der Hoek, and B. Kooi. Dynamic epistemic logic with assignment. In *Proceedings of AAMAS 2005*, pages 141–148, 2005.
31. P. Du and Q. Chen. Axiomatization of hybrid logic of link variations. In N. Gierasimczuk and F. Velázquez-Quesada, editors, *Dynamic Logic. New Trends and Applications*, pages 35–51. Springer Nature Switzerland, 2024.
32. R. Fervari. *Relation-Changing Modal Logics*. PhD thesis, Facultad de Matemática Astronomía y Física, Universidad Nacional de Córdoba, Argentina, 2014. Supervised by C. Areces.
33. M. Franceschet, M. de Rijke, and B. Schlingloff. Hybrid logics on linear structures: Expressivity and complexity. In *TIME-ICTL 2003, Cairns, Queensland, Australia*, pages 166–173, 2003.
34. 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.
35. 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.
36. D. Gabbay. Introducing reactive Kripke semantics and arc accessibility. In A. Avron, N. Dershowitz, and A. Rabinovich, editors, *Pillars of Computer Science, Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*, volume 4800 of *Lecture Notes in Computer Science*, pages 292–341. Springer, 2008.
37. D. Gabbay. *Reactive Kripke Semantics*. Cognitive Technologies. Springer, 2013.
38. J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999. ILLC Dissertation series DS-1999-01.
39. J. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169, 1997.
40. N. Gierasimczuk, L. Kurzen, and F. Velázquez-Quesada. Learning and teaching as a game: A sabotage approach. In He et al. [44], pages 119–132.
41. J-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.
42. Erich Grädel and Martin Otto. On logics with two variables. *Theoretical Computer Science*, 224(1):73–113, 1999.
43. Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.
44. X. He, J. Horty, and E. Pacuit, editors. *LORI 2009*, volume 5834 of *LNCS*. Springer, 2009.
45. J. Hintikka. *Knowledge and belief*. Cornell University Press Ithaca, N.Y., 1962.
46. G. Hoffmann and C. Areces. Htab: A terminating tableaux system for hybrid logic. *Electronic Notes in Theoretical Computer Science*, 231:3–19, March 2009.
47. 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.
48. T. Hoshi. *Logics of public announcements with constrained protocols*. PhD thesis, Philosophy Department, Stanford University, 2007.
49. B. Kooi. Expressivity and completeness for public update logic via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2):231–253, 2007.

50. B. Kooi and B. Renne. Arrow update logic. *Review of Symbolic Logic*, 4(4):536–559, 2011.
51. B. Kooi and B. Renne. Generalized arrow update logic. In K. Apt, editor, *Proceedings of TARK 2011*, pages 205–211. ACM, 2011.
52. 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.
53. 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.
54. Ch. Löding and P. Rohde. Model checking and satisfiability for sabotage modal logic. In P. Pandya and J. Radhakrishnan, editors, *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science: 23rd Conference, Mumbai, India, December 15-17, 2003. Proceedings*, pages 302–313, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
55. 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.
56. M. Marx. Narcissists, stepmothers and spies. In I. Horrocks and S. Tessaris, editors, *Proceedings of the 2002 International Workshop on Description Logics (DL2002), Toulouse, France, April 19-21, 2002*, volume 53 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2002.
57. S. Mera. *Modal Memory Logics*. PhD thesis, Universidad de Buenos Aires & Université Henri Poincaré, 2009. Supervised by C. Areces.
58. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
59. J. Plaza. Logics of public communications. In M. Emrich, M. Pfeifer, M. Hadzikadic, and Z. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.
60. B. Renne, J. Sack, and A. Yap. Dynamic epistemic temporal logic. In He et al. [44], pages 263–277.
61. P. Rohde. *On games and logics over dynamically changing structures*. PhD thesis, RWTH Aachen, 2006.
62. T. Schneider. *The Complexity of Hybrid Logics over Restricted Frame Classes*. PhD thesis, University of Jena, 2007.
63. K. Segerberg. Belief revision from the point of view of doxastic logic. *Logic Journal of the IGPL*, 3(4):535–553, 1995.
64. K. Segerberg. Irrevocable belief revision in dynamic doxastic logic. *Notre Dame Journal of Formal Logic*, 39(3):287–306, 1998.
65. B. ten Cate and M. Franceschet. On the complexity of hybrid logics with binders. In L. Ong, editor, *Computer Science Logic*, volume 3634 of *Lecture Notes in Computer Science*, pages 339–354, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
66. A. Troelstra. *Lectures on Linear Logic*. CSLI Publications, 1992.
67. J. van Benthem. Logics for information update. In *Proceedings of TARK 2001*, pages 51–67. Morgan Kaufmann, 2001.
68. J. van Benthem. An essay on sabotage and obstruction. In *Mechanizing Mathematical Reasoning*, pages 268–276, 2005.
69. J. van Benthem, L. Li, C. Shi, and H. Yin. Hybrid sabotage modal logic. *Journal of Logic and Computation*, 33(6):1216–1242, 03 2022.
70. J. van Benthem and F. Liu. Dynamic logic of preference upgrade. *Journal of Applied Nonclassical Logics*, 17(2):157–182, 2007.
71. J. van Benthem, K. Mierzewski, and F. Zaffora Blando. The modal logic of stepwise removal. *The Review of Symbolic Logic*, 15(1):36–63, 2022.
72. H. van Ditmarsch. Prolegomena to dynamic logic for belief revision. *Synthese*, 147(2):229–275, 2005.
73. H. van Ditmarsch, W. van der Hoek, and B. Kooi. Public announcements and belief expansion. *Advances in Modal Logic* 5, pages 62–73, 2004.
74. H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Kluwer, 2007.