Undecidability of Relation-Changing Modal Logics

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

[1]FaMAF, Universidad Nacional de Córdoba, Argentina
[2]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
{areces, fervari, hoffmann}@famaf.unc.edu.ar
[3]Fachbereich Mathematik und Informatik, Universität Bremen, Germany
martel@informatik.uni-bremen.de

Abstract. Relation-changing modal logics are extensions of the basic modal logic that allow to change the accessibility relation of a model during the evaluation of a formula. In particular, they are equipped with dynamic modalities that are able to delete, add and swap edges in the model, both locally and globally. We investigate the satisfiability problem of these logics. We define satisfiability-preserving translations from an undecidable memory logic to relation-changing modal logics. This way we show that their satisfiability problems are undecidable.

Keywords: modal logics, dynamic logics, satisfiability, undecidability.

1 Introduction

Modal logics [12,14] 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, law, philosophy, etc.

From an abstract point of view, modal logics can be seen as formal languages to navigate and explore properties of a given relational structure. But if we want to describe and reason about *dynamic aspects* of a given situation, e.g., how the relations between a set of elements *evolve* through time or through the application of certain operations, the use of modal logics (or actually, any kind of logic with classical semantics) becomes less clear. We can always resort to modeling the whole space of possible evolutions as a graph, but this soon becomes unwieldy. It would be more elegant to use truly dynamic modal logics with operators that can mimic the changes the structure will undergo.

There exist several dynamic modal operators that fit in this approach. A clear example are the dynamic operators introduced in dynamic epistemic logics (see, e.g. [22]). Less obvious examples are given by hybrid logics [8,13] equipped with the down arrow operator $\downarrow$ which is used to 'rebind' names for states to the current point of evaluation, and memory logics [19], a kind of restricted form of hybrid logics that come equipped with a memory and operators to store and retrieve states from it. Finally, a classical example which can arguably be taken as the origin of the studies of logics in this approach is Sabotage Logic introduced by van Benthem in [21], which provides an operator that deletes individual edges in the model.

Generalizing this last logic, we study operators that do various kinds of change to the accessibility relation of a model: deleting, adding, and swapping edges, both locally (near the state of evaluation) and globally (anywhere). We call these operators *relation-changing*. In [2], the operators are introduced, and it is shown that the model checking problem is PSPACE-complete for the basic modal logic enriched with any of these operators. In this article, we consider the satisfiability problem of these logics. Previous results on this topic are the undecidability of (multimodal) global sabotage logic, via encoding of the Post Correspondence Problem [16] the undecidability of local swap logic with a single relation, by reduction from memory logic [4]; and non-terminating tableau methods for all six logics [3]. Here we present undecidability proofs for all six logics using reductions from memory logic.

The undecidability results can be surprising, considering for instance that dynamic epistemic logics are decidable [17,22,11]. However, other very expressive dynamic operators are undecidable, such as the hybrid logic with the $\downarrow$ operator [8]. As we mentioned before, $\downarrow$ binds states of the model to some particular names. We will show in this article that relation-changing operators can take advantage of adding, deleting or swapping around edges, to perform some sort of binding in the model, turning them undecidable.

Contributions.

- We sketch the undecidability proof for the memory logic $\mathsf{ML}(ⓡ, ⓚ)$, by adapting the undecidability argument introduced in [18] for the description logic $\mathcal{ALC}\mathsf{self}$.
- We introduce undecidability proofs for the satisfiability problem of six relation-changing modal logics via satisfiability of memory logic. In this way, we complete the picture of the computational aspects of the family of languages defined in this framework.
- Our proofs improve previous ones for local swap [4] and global sabotage [16], by exploiting undecidability of memory logics. This allows for shorter proofs and avoid redundant encodings of the tiling problem.

The article is organized as follows. In Section 2 we introduce the syntax and semantics of relation-changing modal logics. In Section 3 we introduce the memory logic $\mathsf{ML}(ⓡ, ⓚ)$ and a sketch of the proof of its undecidability. We dedicate Section 4 to the translations from memory to global and local relation-changing modal logics. Finally we draw our conclusions in Section 5.

2 Relation-Changing Modal Logics

In this section, we formally introduce *relation-changing modal logics*. For more details and motivations, we direct the reader to [15].

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

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

where $p \in \mathsf{PROP}$*,* $\blacklozenge \in \{\langle\mathsf{sb}\rangle, \langle\mathsf{br}\rangle, \langle\mathsf{sw}\rangle, \langle\mathsf{gsb}\rangle, \langle\mathsf{gbr}\rangle, \langle\mathsf{gsw}\rangle\}$ *and* $\varphi, \psi \in \mathsf{FORM}$*. Other operators are defined as usual. In particular,* $\blacksquare\varphi$ *is defined as* $\neg\blacklozenge\neg\varphi$*.*

Let ML *(the basic modal logic) be the logic without the* $\{\langle\mathsf{sb}\rangle, \langle\mathsf{br}\rangle, \langle\mathsf{sw}\rangle, \langle\mathsf{gsb}\rangle, \langle\mathsf{gbr}\rangle, \langle\mathsf{gsw}\rangle\}$ *operators, and* $\mathsf{ML}(\blacklozenge)$ *the extension of* ML *allowing also* $\blacklozenge$*, for* $\blacklozenge \in \{\langle\mathsf{sb}\rangle, \langle\mathsf{br}\rangle, \langle\mathsf{sw}\rangle, \langle\mathsf{gsb}\rangle, \langle\mathsf{gbr}\rangle, \langle\mathsf{gsw}\rangle\}$*.*

Semantically, formulas are evaluated in standard relational models, and the meaning of the operators of the basic modal logic remains unchanged (see [12] for details). When we evaluate formulas containing relation-changing operators, we will need to keep track of the edges that have been modified. To that end, let us define precisely the models that we will use.

Definition 2 (Models and model updates). *A model* M *is a triple* $\mathsf{M} = \langle W, R, V\rangle$*, where* W *is a non-empty set whose elements are called points or states,* $R \subseteq W \times W$ *is the accessibility relation, and* $V : \mathsf{PROP} \to \mathcal{P}(W)$ *is a valuation. We define the following notation:*

(sabotaging) $\mathsf{M}^-_S = \langle W, R^-_S, V\rangle$*, with* $R^-_S = R \backslash S$*,* $S \subseteq R$*.*
(bridging) $\mathsf{M}^+_S = \langle W, R^+_S, V\rangle$*, with* $R^+_S = R \cup S$*,* $S \subseteq (W \times W)\backslash R$*.*
(swapping) $\mathsf{M}^*_S = \langle W, R^*_S, V\rangle$*, with* $R^*_S = (R\backslash S^{-1}) \cup S$*,* $S \subseteq R^{-1}$*.*

Intuitively, M^-_S is obtained from M by deleting the edges in S, and similarly M^+_S adds the edges in S to the accessibility relation, and M^*_S adds the edges in S as inverses of edges previously in the accessibility relation.

Let w be a state in M, the pair (M, w) is called a pointed model (we will usually drop parentheses). In the rest of this article, we will use wv as a shorthand for $\{(w, v)\}$ or (w, v); context will always disambiguate the intended use.

Definition 3 (Semantics). *Given a pointed model* M, w *and a formula* φ*, we say that* M, w *satisfies* φ*, and write* $\mathsf{M}, w \models \varphi$*, when*

$\mathsf{M}, w \models p$ *iff* $w \in V(p)$
$\mathsf{M}, w \models \neg\varphi$ *iff* $\mathsf{M}, w \not\models \varphi$
$\mathsf{M}, w \models \varphi \wedge \psi$ *iff* $\mathsf{M}, w \models \varphi$ *and* $\mathsf{M}, w \models \psi$
$\mathsf{M}, w \models \Diamond\varphi$ *iff for some* $v \in W$ *s.t.* $(w, v) \in R, \mathsf{M}, v \models \varphi$
$\mathsf{M}, w \models \langle\mathsf{sb}\rangle\varphi$ *iff for some* $v \in W$ *s.t.* $(w, v) \in R, \mathsf{M}^-_{wv}, v \models \varphi$
$\mathsf{M}, w \models \langle\mathsf{br}\rangle\varphi$ *iff for some* $v \in W$ *s.t.* $(w, v) \notin R, \mathsf{M}^+_{wv}, v \models \varphi$
$\mathsf{M}, w \models \langle\mathsf{sw}\rangle\varphi$ *iff for some* $v \in W$ *s.t.* $(w, v) \in R, \mathsf{M}^*_{vw}, v \models \varphi$
$\mathsf{M}, w \models \langle\mathsf{gsb}\rangle\varphi$ *iff for some* $v, u \in W$*, s.t.* $(v, u) \in R, \mathsf{M}^-_{vu}, w \models \varphi$
$\mathsf{M}, w \models \langle\mathsf{gbr}\rangle\varphi$ *iff for some* $v, u \in W$*, s.t.* $(v, u) \notin R, \mathsf{M}^+_{vu}, w \models \varphi$
$\mathsf{M}, w \models \langle\mathsf{gsw}\rangle\varphi$ *iff for some* $v, u \in W$*, s.t.* $(v, u) \in R, \mathsf{M}^*_{uv}, w \models \varphi$*.*

We say φ is satisfiable if for some pointed model M, w, we have $\mathsf{M}, w \models \varphi$.

Notice that $\langle \mathsf{br} \rangle$ and $\langle \mathsf{gbr} \rangle$ always add new edges in the model, and fail in case no new edge can be created. Other versions in which such edge is not necessarily new could be considered, but in that case the operators would behave sometimes as a $\Diamond$ or as a "do nothing", respectively. However, we conjecture that similar results could be proved for those and other versions of the operations.

Relation-changing operators can modify the accessibility relation and check for such changes in the model, and therefore can be used to mark and check for marked states, simulating some sort of binding. Adequately, marking and checking states are the basic dynamic operations *remember* and *known* that can be performed by *memory logics*, a formalism that we present in the next section.

3 Undecidability of Monomodal Memory Logic

Memory logics [1,19] are modal logics that can *store* the current state of evaluation into a memory and *check* whether the current state belongs to this memory. The memory is a subset of the domain of the model. We call $\mathsf{ML}(Ⓡ, Ⓚ)$ the memory logic that extends ML with the operators Ⓡ and Ⓚ, which stand for "remember" and "known", respectively.

Definition 4 (Syntax). *Let PROP be a countable, infinite set of propositional symbols. The set FORM of formulas over PROP is defined as:*

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

where $p \in \mathsf{PROP}$ and $\varphi, \psi \in \mathsf{FORM}$. Other operators are defined as usual.

Definition 5 (Semantics). *A model $\mathsf{M} = \langle W, R, V, S \rangle$ 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:*

$$\begin{array}{lll} \langle W, R, V, S \rangle, w \models Ⓡ\varphi & \textit{iff} & \langle W, R, V, S \cup \{w\} \rangle, w \models \varphi \\ \langle W, R, V, S \rangle, w \models Ⓚ & \textit{iff} & w \in S. \end{array}$$

The remaining cases coincide with the semantics of ML, and do not involve the memory.

An $\mathsf{ML}(Ⓡ, Ⓚ)$-formula φ is satisfiable *if there are a model $\mathsf{M} = \langle W, R, V, \emptyset \rangle$ and $w \in W$ such that $\mathsf{M}, w \models \varphi$. The empty initial memory ensures that no state of the model satisfies the unary predicate Ⓚ unless a formula $Ⓡ\psi$ has previously been evaluated there.*

Multimodal memory logic is shown to be undecidable in [7]. We strengthen this result, showing that undecidability holds also in the monomodal case.

Theorem 1. *The satisfiability problem of $\mathsf{ML}(Ⓡ, Ⓚ)$ is undecidable.*

Proof. The problem of concept consistency in the description logic $\mathcal{ALC}\mathsf{self}$ is undecidable [18]. Let us name $\mathsf{Tiling}(t)$ the concept defined in [18] that encodes an instance t of the (undecidable) problem of tiling the plane. A reduction of Tiling to the satisfiability problem of $\mathsf{ML}(ⓡ, ⓚ)$ can be done by replacing the $\mathcal{ALC}\mathsf{self}$ operator $\forall R$ by $\Box$, $\exists R$ by $\Diamond$, $\mathbf{I}$ by ⓡ and **me** by ⓚ.

We previously suggested that relation-changing operators could, each one in its own way, simulate remember and known operators. However, there is one important difference between the ⓡ operator and relation-changing operators like $\langle \mathsf{sb} \rangle$. While $\langle \mathsf{sb} \rangle\varphi$ always results in a change in the model, ⓡφ can leave the memory unchanged if the current state of evaluation is already memorized. We ignore this difference by observing that any $\mathsf{ML}(ⓡ, ⓚ)$-formula can be rewritten into an equivalent formula where every occurrence of ⓡ is "proper", in the sense that it actually modifies the memory.

Definition 6 (PNF). *An* $\mathsf{ML}(ⓡ, ⓚ)$*-formula is in* proper normal form *(PNF) if every occurrence of a sub-formula* ⓡψ *occurs within the following sub-formula:*

$$(\neg ⓚ \wedge ⓡ\psi) \vee (ⓚ \wedge \psi)$$

Finally, we define the notion of *modal depth* of an $\mathsf{ML}(ⓡ, ⓚ)$-formula.

Definition 7. *Given* φ *in* $\mathsf{ML}(ⓡ, ⓚ)$*, we define the* modal depth *of* φ *(notation* $\mathsf{md}\varphi$*) as*

$$\begin{aligned} \mathsf{md}(ⓚ) &= 0 \\ \mathsf{md}(p) &= 0 \textit{ for } p \in \mathsf{PROP} \\ \mathsf{md}(ⓡ\varphi) &= \mathsf{md}(\varphi) \\ \mathsf{md}(\neg\varphi) &= \mathsf{md}(\varphi) \\ \mathsf{md}(\varphi \wedge \psi) &= max\{\mathsf{md}(\varphi), \mathsf{md}(\psi)\} \\ \mathsf{md}(\Diamond\varphi) &= 1 + \mathsf{md}(\varphi). \end{aligned}$$

In the next section we prove that the satisfiability problem of relation-changing modal logics is undecidable via reductions from monomodal memory logic. We assume that memory logic formulas are always in PNF. This is important for structural inductive proofs.

4 Undecidability of Relation-Changing Logics

In this section, we present satisfiability-preserving translations from $\mathsf{ML}(ⓡ, ⓚ)$ to relation-changing modal logics. Combining these translations with the undecidability result of Theorem 1, we can claim:

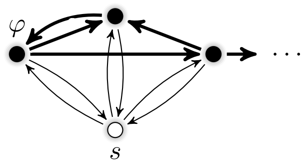
Theorem 2. *The satisfiability problem of* $\mathsf{ML}(\blacklozenge)$ *is undecidable, for* $\blacklozenge \in \{\langle \mathsf{sb} \rangle,$ $\langle \mathsf{br} \rangle, \langle \mathsf{sw} \rangle, \langle \mathsf{gsb} \rangle, \langle \mathsf{gbr} \rangle, \langle \mathsf{gsw} \rangle\}$.

The main idea of these translations is to simulate the behavior of $\mathsf{ML}(ⓡ, ⓚ)$ without having an external memory in the model. We simulate the ability to store states in a memory by changing the accessibility relation of a model. Checking for membership in the memory is simulated by checking for changes in the accessibility relation.

Every translation $\tau_{\blacklozenge}$ from $\mathsf{ML}(\textcircled{r}, \textcircled{k})$-formulas to $\mathsf{ML}(\blacklozenge)$-formulas proceeds in two steps. For a given target logic, the translation includes a fixed part called $Struct_{\blacklozenge}$, that enforces constraints on the structure of the model. The second part, called $\mathsf{Tr}_{\blacklozenge}$, is defined inductively on $\mathsf{ML}(\textcircled{r}, \textcircled{k})$-formulas, and uses the structure provided by $Struct_{\blacklozenge}$ to simulate the $\textcircled{r}$ and $\textcircled{k}$ operators.

Sabotage Logic

Local Sabotage. In the translation to local sabotage logic, the $Struct_{\langle \mathsf{sb} \rangle}$ subformula should ensure that every state of the model can be memorized using the expressivity of $\langle \mathsf{sb} \rangle$. This operator changes the point of evaluation after deleting an edge. To compensate for this, the $Struct_{\langle \mathsf{sb} \rangle}$ formula guarantees that every state has an edge that is deleted when the state is memorized, and a second edge back to the original state to ensure that evaluation can continue at the correct state. We use a spy point s to ensure this structure. The idea is illustrated in the following image.



We need to ensure that every satisfiable formula of $\mathsf{ML}(\textcircled{r}, \textcircled{k})$ is translated into a satisfiable formula (and vice-versa, if the translated formula is satisfiable, then the original formula is satisfiable, too). The image above shows an intended model for the translated formula $\tau_{\langle \mathsf{sb} \rangle}(\varphi)$. Intuitively, bold edges and arrows correspond to the model of φ. The complete translation is given in Definition 8. Here we discuss in detail how it works.

$Struct_{\langle \mathsf{sb} \rangle}$ adds a spy state with symmetric edges between itself and all other states. In particular, (1) in Definition 8 ensures that the evaluation state satisfies s and that it is irreflexive, and (2) guarantees that its immediate successors reach a state where s holds. Formulas (3) and (4) ensure that this state is the original s state. They work together as follows: (3) makes $\Box\Diamond s$ true in any s-state reachable in two steps, and by deleting the traversed edges we avoid a cycle of size two between this s-state and an immediate successor of the evaluation state, distinguishing the original s-state from any other s-state reachable in two steps. (4) then traverses one edge, deletes the next one, and reaches a state where s implies $\Diamond\Box\neg s$. This contradicts (3), unless we have arrived in the original s state. Formulas (5), (6) and (7) mimic (2), (3) and (4), but for edges which are removed twice. Observe that (6) now avoids a cycle of size three between any other s-state reachable in two steps and an immediate successor of the evaluation state. Finally, (8) and (9) ensure that the evaluation state is indeed a spy state, i.e., that it is linked to every other state of the input model.

$\mathsf{Tr}_{\langle \mathsf{sb} \rangle}$ starts by placing the translation $(\ )'$ of φ in a successor of the evaluation state. Boolean cases are obvious. For the diamond case, $\Diamond\psi$ is satisfied if there is a successor v where ψ holds, but we must ensure that v is not the spy state. For $(ⓡ\psi)'$, we do a round-trip of sabotaging from the current state to the spy state. Note that after reaching the spy state an edge does come back to the same state where it came from, since the only accessible state where $\neg\Diamond s$ holds is the one we are memorizing. For $(ⓚ)'$, we check whether there is an edge pointing to some s-state.

Definition 8. *Define* $\tau_{\langle \mathsf{sb} \rangle}(\varphi) = \mathit{Struct}_{\langle \mathsf{sb} \rangle} \wedge \mathsf{Tr}_{\langle \mathsf{sb} \rangle}(\varphi)$*, where:*

$$
\begin{aligned}
\mathit{Struct}_{\langle \mathsf{sb} \rangle} = {} & s \wedge \Box\neg s & (1)\\
& \wedge \Box\Diamond s & (2)\\
& \wedge [\mathsf{sb}][\mathsf{sb}](s \to \Box\Diamond s) & (3)\\
& \wedge \Box[\mathsf{sb}](s \to \Diamond\Box\neg s) & (4)\\
& \wedge \Box\Box(\neg s \to \Diamond s) & (5)\\
& \wedge \Box[\mathsf{sb}](s \to [\mathsf{sb}](\Box\neg s \to \Box\Box(s \to \Box\Diamond s))) & (6)\\
& \wedge \Box[\mathsf{sb}](s \to \Box(\Box\neg s \to \Box\Box(s \to \Diamond\Box\neg s))) & (7)\\
& \wedge \Box\Box\Box(s \to \Box\Diamond s) & (8)\\
& \wedge \Box\Box[\mathsf{sb}](s \to \Diamond\Box\neg s) & (9)
\end{aligned}
$$

$\mathsf{Tr}_{\langle \mathsf{sb} \rangle}(\varphi) = \Diamond(\varphi)'$, *with:*

$$
\begin{aligned}
(p)' &= p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi\\
(ⓚ)' &= \neg\Diamond s\\
(\neg\psi)' &= \neg(\psi)'\\
(\psi \wedge \chi)' &= (\psi)' \wedge (\chi)'\\
(\Diamond\psi)' &= \Diamond(\neg s \wedge (\psi)')\\
(ⓡ\psi)' &= \langle \mathsf{sb} \rangle(s \wedge \langle \mathsf{sb} \rangle(\neg\Diamond s \wedge (\psi)'))
\end{aligned}
$$

Proposition 1. *If* $\langle W, R, V\rangle, w \models \mathit{Struct}_{\langle \mathsf{sb} \rangle}$*, then for every state* $v \in W \setminus \{w\}$ *there exists exactly one state* v' *such that* $(v, v'), (v', v) \in R$ *and* $v' \in V(s)$*.*

Lemma 1. *Let* φ *be an* $\mathsf{ML}(ⓡ, ⓚ)$*-formula in PNF that does not contain the propositional symbol* s*. Then,* φ *is satisfiable iff* $\tau_{\langle \mathsf{sb} \rangle}(\varphi)$ *is satisfiable.*

Proof. **(⇐)** Suppose $\langle W, R, V\rangle, s \models \tau_{\langle \mathsf{sb} \rangle}(\varphi)$. Let $W' = W \setminus V(s)$, $R' = R \cap (W' \times W')$ and $V'(p) = V(p) \cap W'$ for all $p \in \mathsf{PROP}$. By definition of $\mathsf{Tr}_{\langle \mathsf{sb} \rangle}$ there is $w' \in W'$ such that $(s, w') \in R$ and $\langle W, R, V\rangle, w' \models (\varphi)'$.

Now, let ψ be a sub-formula of φ, $v \in W'$, $S \subseteq W'$ and $R_S = R \setminus \{(v, s), (s, v) \mid v \in S\}$. We prove by structural induction on ψ that $\langle W', R', V', S\rangle, v \models \psi$ if, and only if, $\langle W, R_S, V\rangle, v \models (\psi)'$. In particular, this will prove that $\langle W', R', V', \emptyset\rangle, w' \models \varphi$ if, and only if, $\langle W, R, V\rangle, w' \models (\varphi)'$.

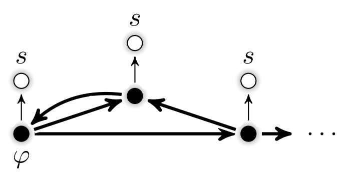
The propositional, Boolean and modal cases are trivial. For $\psi = ⓚ$, we should prove that $\langle W', R', V', S\rangle, v \models ⓚ$ if, and only if, $\langle W, R_S, V\rangle, v \models \neg\Diamond s$. However this is immediate by definition of S and R_S and Proposition 1.

For the last case, consider $\psi = \neg \textcircled{k} \wedge \textcircled{r}\chi$ (remember that formulas are in PNP), so we should prove that $\langle W', R', V', S\rangle, v \models \neg \textcircled{k} \wedge \textcircled{r}\chi$ if, and only if, $\langle W, R_S, V\rangle, v \models \Diamond s \wedge \langle \mathsf{sb}\rangle(s \wedge \langle \mathsf{sb}\rangle(\neg\Diamond s \wedge (\chi)'$. Again, the equivalence is immediate by Proposition 1.

($\Rightarrow$) Suppose $\langle W, R, V, \emptyset\rangle, w \models \varphi$. We build a model for $\tau_{\langle \mathsf{sb}\rangle}(\varphi)$ by adding the necessary parts to this model, that are, the spy state and the round-trip paths. Define $\langle W', R', V'\rangle$ as follows. Let $s \notin W$ some state, $W' = W \cup \{s\}$, $R' = R \cup \{(x, s), (s, x) \mid x \in W\}$, $V'(s) = \{s\}$ and $V'(p) = V(p)$ for $p \in \mathsf{PROP} \setminus \{s\}$. By construction, $\langle W', R', V'\rangle, s \models \mathit{Struct}_{\langle \mathsf{sb}\rangle}$, so Proposition 1 holds. We prove that for all ψ sub-formula of φ, $v \in W$, $S \subseteq W$ and $R'_S = R' \setminus \{(x, s), (s, x) \mid x \in S\}$, $\langle W, R, V, S\rangle, v \models \psi$ iff $\langle W', R'_S, V'\rangle, v \models (\psi)'$. This can be done by structural induction on ψ using Proposition 1. This proves that $\langle W, R, V, \emptyset\rangle, w \models \varphi$ iff $\langle W', R', V'\rangle, s \models \tau_{\langle \mathsf{sb}\rangle}(\varphi)$, so $\tau_{\langle \mathsf{sb}\rangle}(\varphi)$ is satisfiable.

Global Sabotage. In [16] it is shown that multimodal sabotage logic 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 $\mathsf{ML}(\textcircled{r}, \textcircled{k})$. The notation $\Box^i\varphi$ is defined as $\Box^0\varphi = \varphi$ and $\Box^{n+1}\varphi = \Box\Box^n\varphi$.

One piece of data needed to build $\tau_{\langle \mathsf{gsb}\rangle}(\varphi)$ is the modal depth of the input formula ($\mathsf{md}(\varphi)$). Up to the depth indicated by this value, $\mathit{Struct}_{\langle \mathsf{gsb}\rangle}(\varphi)$ adds to every state a transition to some state where s holds (In fact, this latter state can be shared among several states of the input model.) It is as if each state of the input model had a flag that could be turned on to identify the state. Thus, remembering some state is simulated with $\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\textcircled{r})$ by deleting the edge between the state and its s-successor. For $\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\textcircled{k})$, we check whether the current state has an s-successor. The idea is illustrated in the following image.



Definition 9. *Define* $\tau_{\langle \mathsf{gsb}\rangle}(\varphi) = \mathit{Struct}_{\langle \mathsf{gsb}\rangle}(\varphi) \wedge \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\varphi)$, *where:*

$$\mathit{Struct}_{\langle \mathsf{gsb}\rangle}(\varphi) = \neg s \wedge \bigwedge_{0 \le i \le \mathsf{md}(\varphi)} \Box^i(\neg s \ \rightarrow \Diamond s)$$

$$\begin{array}{ll}
\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(p) & = p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi \\
\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\textcircled{k}) & = \neg\Diamond s \\
\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\neg\psi) & = \neg\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi) \\
\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi \wedge \chi) & = \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi) \wedge \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\chi) \\
\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\Diamond\psi) & = \Diamond(\neg s \wedge \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi)) \\
\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\textcircled{r}\psi) & = \langle \mathsf{gsb}\rangle(\neg\Diamond s \wedge \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi))
\end{array}$$

Proposition 2. *Let $dist(a, b)$ the minimal number of R-steps to reach some state b from some state a. Let φ some memory logic formula. If $\langle W, R, V\rangle, w \models Struct_{\langle \mathsf{gsb}\rangle}(\varphi)$, then for all $x \in W$ such that $dist(w, x) \leq md(\varphi)$, x has a successor where s holds.*

Proposition 3. *If $\langle W, R, V\rangle, w \models \Diamond s \wedge \langle \mathsf{gsb}\rangle \neg\Diamond s$, then w has one and only one successor where s holds.*

Lemma 2. *Let φ be an $\mathsf{ML}(ⓡ, ⓚ)$-formula in PNF that does not contain the propositional symbol s. Then, φ is satisfiable iff $\tau_{\langle \mathsf{gsb}\rangle}(\varphi)$ is satisfiable.*

Proof. ($\Leftarrow$) Suppose $\langle W, R, V\rangle, w \models \tau_{\langle \mathsf{gsb}\rangle}(\varphi)$. Let $W' = W \backslash V(s)$, $R' = R \cap (W' \times W')$, and $V'(p) = V(p) \cap W'$ for $p \in \mathsf{PROP} \setminus \{s\}$. We should prove that for all ψ sub-formula of φ of modal depth $\mathsf{md}(\psi) \leq \mathsf{md}(\varphi) - dist(w, v)$, $v \in W'$ accessible from w within $\mathsf{md}(\varphi)$ steps, $S \subseteq W'$, and $R_S = R \setminus \{(x, y) \mid |x \in S, y \in V(s)\}$, then $\langle W', R', V', S\rangle, v \models \psi$ iff $\langle W, R_S, V\rangle, v \models \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi)$.

The proof is by structural induction on ψ. The non-memory cases are easy. For the ⓚ case, we should show that $\langle W', R', V', S\rangle, v \models ⓚ$ iff $\langle W, R_S, V\rangle, v \models \neg\Diamond s$, this is immediate by Proposition 2 and the definitions of S and R_S.

Then for the remaining case, we have to show that $\langle W', R', V', S\rangle, v \models \neg ⓚ \wedge ⓡ\chi$ iff $\langle W, R_S, V\rangle, v \models \Diamond s \wedge \langle \mathsf{gsb}\rangle(\neg\Diamond s \wedge \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\chi))$, which can be proved using the definition of $\models$, IH and Proposition 3.

($\Rightarrow$) Suppose $\langle W, R, V, \emptyset\rangle, w \models \varphi$. Let $s \notin W$. Define $\langle W', R', V'\rangle$, where $W' = W \cup \{s\}$, $R' = R \cup \{(v, s) \mid v \in W\}$, $V'(s) = \{s\}$, and $V'(p) = V(p)$, for $p \in \mathsf{PROP}$ appearing in φ. It is easy to check that $\langle W', R', V'\rangle, w \models Struct_{\langle \mathsf{gsb}\rangle}(\varphi)$, hence Proposition 2 holds. Then, let us prove that for all ψ sub-formula of φ of modal depth $\mathsf{md}(\psi) \leq \mathsf{md}(\varphi) - dist(w, v)$, $v \in W$ accessible from w within $\mathsf{md}(\varphi)$ steps, $S \subseteq W$ and $R'_S = R' \setminus \{(x, s) \mid x \in S\}$, we have the equivalence $\langle W, R, V, S\rangle, v \models \psi$ iff $\langle W', R'_S, V'\rangle, v \models \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\psi)$. This is done by structural induction on ψ. For the case ⓚ the equivalence is immediate, and for the case $\neg ⓚ \wedge ⓡ\chi$, Proposition 3 provides the equivalence needed.

Bridge Logic

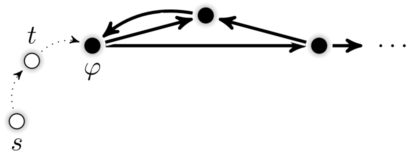
Local Bridge. For local bridge logic, we use a spy state that is initially disconnected from the input model. When some state should be memorized, the spy state gets connected (in both directions) to it. This construction is quite special since we do not have pre-built gadgets in the input model, as they get built on demand.

Let us first show the following result, that enables us to force the evaluation state to be the only one in the model to satisfy s:

Lemma 3. *Let $\varphi = s \wedge \Box\bot \wedge [\mathsf{br}](s \rightarrow [\mathsf{br}]\neg s)$. If $\mathsf{M}, w \models \varphi$, then w is the only state in the model M where s holds.*

Proof. First, w obviously satisfies s and does not have any successor. Now, we have $\mathsf{M}, w \models [\mathsf{br}](s \rightarrow [\mathsf{br}]\neg s)$. In particular this means that $\mathsf{M}^+_{ww}, w \models s \rightarrow [\mathsf{br}]\neg s$, hence $\mathsf{M}^+_{ww}, w \models [\mathsf{br}]\neg s$. Since in M^+_{ww}, the state w is only connected to itself, this means that for all $v \neq w$, we have $\mathsf{M}^{+}_{ww,wv}, v \models \neg s$, this also means that $\mathsf{M}, v \not\models s$ for all $v \neq w$.

For Bridge Logics, $Struct_{\langle \mathsf{br} \rangle}$ adds to the input model a spy state in which s holds. By Lemma 3, (1) in Definition 10 ensures that the evaluation state has no successor and is the only state in the model where s holds. And (2) ensures that there are no edges from $\neg s$-states (anywhere in the model) to the spy state. The idea is illustrated in the following image, where t is a propositional symbol used in $\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi)$ and dotted lines represent edges created with the $\langle \mathsf{br} \rangle$ operator.



Definition 10. *Define* $\tau_{\langle \mathsf{br} \rangle}(\varphi) = Struct_{\langle \mathsf{br} \rangle} \wedge \mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi)$, *where:*

$$\begin{aligned} Struct_{\langle \mathsf{br} \rangle} = {} & s \wedge \Box\bot \wedge [\mathsf{br}](s \to [\mathsf{br}]\neg s) && (1) \\ & \wedge [\mathsf{br}](\neg s \to \Box\neg s) && (2) \end{aligned}$$

$\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi) = \langle \mathsf{br} \rangle(\neg s \wedge t \wedge \langle \mathsf{br} \rangle(\neg s \wedge \neg t \wedge (\varphi)'))$, *with:*

$$\begin{aligned} (p)' &= p \quad \text{for } p \in \mathsf{PROP} \text{ appearing in } \varphi \\ (ⓚ)' &= \Diamond s \\ (\neg\psi)' &= \neg(\psi)' \\ (\psi \wedge \chi)' &= (\psi)' \wedge (\chi)' \\ (\Diamond\psi)' &= \Diamond(\neg s \wedge \neg t \wedge (\psi)') \\ (ⓡ\psi)' &= \langle \mathsf{br} \rangle(s \wedge \langle \mathsf{br} \rangle(\neg s \wedge \Diamond s \wedge (\psi)')) \end{aligned}$$

$\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi)$ first creates two edges until a $\neg s$-state, where the translation of φ holds. For $\mathsf{Tr}_{\langle \mathsf{br} \rangle}(ⓡ)$ we do a round-trip of bridging from the current state to the spy state. Note that the second part of this round-trip has to be from the spy state to the remembered state, since it is the only way to satisfy $\langle \mathsf{br} \rangle(\Diamond s)$. Also note that this would not work if the s state was directly connected to the input model; this is why we use the intermediate t-state. For $\mathsf{Tr}_{\langle \mathsf{br} \rangle}(ⓚ)$ we check whether there is an edge to a state where s holds.

Proposition 4. *Let $\langle W, R, V \rangle$ a model such that there is a unique state s where s holds, there is no state $x \in W$ such that $(x, s) \in R$, and there is a component $C \subseteq W$ such that $s \notin C$ and for all $y \in C$, $(s, y) \notin R$. Let $S \subseteq C$ and $R_S = R \cup \{(x, s), (s, x) \mid x \in S\}$.*

Then in the model $\langle W, R_S, V \rangle$, evaluating the formula $\langle \mathsf{br} \rangle(s \wedge \langle \mathsf{br} \rangle \Diamond s)$ at some state $y \in C \setminus S$ changes the evaluation state to s, then again to the same state y adding the edges (y, s) and (s, y) to the relation.

Lemma 4. *Let φ be an $\mathsf{ML}(ⓡ, ⓚ)$-formula in PNF that does not contain the propositional symbols s and t. Then, φ is satisfiable iff $\tau_{\langle \mathsf{br} \rangle}(\varphi)$ is satisfiable.*

Proof. ($\Leftarrow$) Suppose $\langle W, R, V\rangle, s \models \tau_{\langle \mathsf{br}\rangle}(\varphi)$. Define $\mathsf{M}' = \langle W', R', V', \emptyset\rangle$ with $W' = (W \setminus V(s)) \setminus V(t)$, $R' = R \cap (W' \times W')$, and $V'(p) = V(p) \cap W'$ for all $p \in \mathsf{PROP}$. By definition of $\mathsf{Tr}_{\langle \mathsf{br}\rangle}$ there is $w' \in W'$ such that $s \neq w'$ and $\langle W, R, V\rangle, w' \models (\varphi)'$.

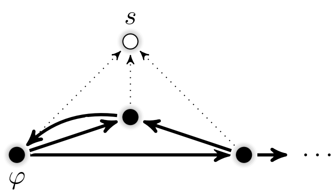
Let ψ a sub-formula of φ, $v \in W'$, $S \subseteq W'$, and $R_S = R \cup \{(x, s), (x, v) \mid x \in S\}$, then we will prove that $\langle W', R', V', S\rangle, v \models \psi$ iff $\langle W, R_S, V\rangle, v \models (\psi)'$.

We prove it by structural induction on ψ. For the $\neg \textcircled{k} \wedge \textcircled{r}\chi$ case, suppose $\langle W', R', V', S\rangle, v \models \neg \textcircled{k} \wedge \textcircled{r}\chi$. By definition, this is equivalent to $\langle W', R', V', S \cup \{v\}\rangle, v \models \chi$ with $v \notin S$, Then, by definition of R_S and inductive hypothesis we get $\langle W, (R_S)^{+}_{\{(v,s),(s,v)\}}, V\rangle, s \models (\chi)'$, with $(v, s) \notin R_S$ and $(s, v) \notin R_S$. By Proposition 4, this is equivalent to $\langle W, R_S, V\rangle, v \models \neg\Diamond s \wedge \langle \mathsf{br}\rangle(s \wedge \langle \mathsf{br}\rangle(\Diamond s \wedge (\chi)'))$. thus we have $\langle W, R_S, V\rangle, v \models (\neg \textcircled{k} \wedge \textcircled{r}\chi)'$.

($\Rightarrow$) Suppose $\langle W, R, V, \emptyset\rangle, w \models \varphi$. Let $s, t \notin W$. Define $\mathsf{M}' = \langle W', R, V'\rangle$ such that $W' = W \cup \{s, t\}$, $V'(s) = \{s\}$, $V'(t) = \{t\}$ and $V'(p) = V(p)$ for $p \in \mathsf{PROP}$ appearing in φ. We can easily check that $\langle W', R, V'\rangle, s \models \mathit{Struct}_{\langle \mathsf{br}\rangle}$, and we can also check by structural induction on φ that $\langle W, R, V, S\rangle, w \models \varphi$ iff $\langle W', R_S, V'\rangle, s \models \mathsf{Tr}_{\langle \mathsf{br}\rangle}(\varphi)$, where $R_S = R \cup \{(v, s), (s, v) \mid v \in S\}$.

Global Bridge. The global bridge operator is able to add edges in the model. This is why, to mark some state, we use this operator to add an edge to some s-state. Then, we enforce that the initial model does not have any reachable s-state.

Here $\mathit{Struct}_{\langle \mathsf{gbr}\rangle}(\varphi)$ ensures that no state of the input model has s-successors. Storing a state in the memory is simulated by creating an edge to an s-state, and checking whether the current state of evaluation is in the memory is simulated by checking the presence of an s-successor. Observe that we could have either one state where s holds or (possibly) different s-states for each state of the input model.



Definition 11. *Define* $\tau_{\langle \mathsf{gbr}\rangle}(\varphi) = \mathit{Struct}_{\langle \mathsf{gbr}\rangle}(\varphi) \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\varphi)$, *where:*

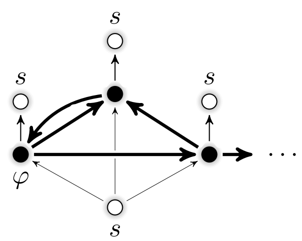
$$
\begin{array}{ll}
\mathit{Struct}_{\langle \mathsf{gbr}\rangle}(\varphi) = & \bigwedge_{0 \leq i \leq \mathsf{md}(\varphi)+1} \Box^i \neg s \\
\mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(p) & = p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi \\
\mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\textcircled{k}) & = \Diamond s \\
\mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\neg\psi) & = \neg \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi) \\
\mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi \wedge \chi) & = \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi) \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\chi) \\
\mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\Diamond\psi) & = \Diamond(\neg s \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi)) \\
\mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\textcircled{r}\psi) & = \langle \mathsf{gbr}\rangle(\Diamond s \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi))
\end{array}
$$

Lemma 5. *Let φ be an $\mathsf{ML}(\textcircled{r}, \textcircled{k})$-formula in PNF that does not contain the propositional symbol s. Then, φ is satisfiable iff $\tau_{\langle \mathsf{gbr}\rangle}(\varphi)$ is satisfiable.*

Swap Logic

Local Swap. We introduce a new version of the translation given in [4] that uses only one propositional symbol. The idea is that we have each state pointing to some states called *switch states*, and memorizing a state is represented by swapping such edges. Then, no edge pointing to a switch means that the state has been memorized. We use the notation $\Box^{(n)}\varphi$ for $\bigwedge_{1\leq i\leq n} \Box^i\varphi$.

In this case $Struct_{\langle\mathsf{sw}\rangle}$ adds "switch states", which are in one-to-one correspondence with the states of the input model, together with a spy state. By (2) in Definition 12, each $\neg s$-state at one, two and three steps from the evaluation state, has a unique dead-end successor where s holds (switch state). By (3) and (4), switch states (corresponding to states at distance 1, 2 and 3) can be reached from the evaluation state by a unique path. (5) makes the evaluation state a spy state. All these conjuncts together ensure that switch states are independent one from another. The idea is illustrated in the following image.



Definition 12. *Define* $\tau_{\langle\mathsf{sw}\rangle} = Struct_{\langle\mathsf{sw}\rangle} \wedge \mathsf{Tr}_{\langle\mathsf{sw}\rangle}(\varphi)$*, where:*

$$
\begin{aligned}
&Struct_{\langle\mathsf{sw}\rangle} = \\
&\quad s \wedge \Box\neg s && (1)\\
&\quad \wedge\ \Box^{(3)}(\neg s \rightarrow Uniq) && (2)\\
&\quad \wedge\ \Box[\mathsf{sw}](s \rightarrow \Box\Box\Box(s \rightarrow \Box\bot) && (3)\\
&\quad \wedge\ \Box\Box[\mathsf{sw}](s \rightarrow \Box\Box\Box(s \rightarrow \Box\bot) && (4)\\
&\quad \wedge\ [\mathsf{sw}][\mathsf{sw}](\neg s \rightarrow \langle\mathsf{sw}\rangle(s \wedge \Diamond((\Box\neg s) \rightarrow \Diamond\Diamond(s \wedge \Diamond\neg\Diamond s)))) && (5)
\end{aligned}
$$

$$Uniq = \Diamond(s \wedge \Box\bot) \wedge [\mathsf{sw}](s \rightarrow \Box\neg\Diamond s)$$

$\mathsf{Tr}_{\langle\mathsf{sw}\rangle}(\varphi) = \Diamond(\varphi)'$*, with:*

$$
\begin{aligned}
(p)' &= p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi\\
(\textcircled{k})' &= \neg\Diamond s\\
(\neg\psi)' &= \neg(\psi)'\\
(\psi \wedge \chi)' &= (\psi)' \wedge (\chi)'\\
(\Diamond\psi)' &= \Diamond(\neg s \wedge (\psi)')\\
(\textcircled{r}\,\psi)' &= \langle\mathsf{sw}\rangle(s \wedge \Diamond(\psi)')
\end{aligned}
$$

For $\mathsf{Tr}_{\langle\mathsf{sw}\rangle}(\textcircled{r}\,\varphi)$ we traverse and swap the edge between the current state and its switch state, and come back to the same state. For $\mathsf{Tr}_{\langle\mathsf{sw}\rangle}(\textcircled{k})$, we check whether the current state has not an edge to its switch state.

Proposition 5. *Let* $\langle W, R, V\rangle, s \models \mathit{Struct}_{\langle\mathsf{sw}\rangle}$*,* $W' = W \setminus V(s)$ *and* $S \subseteq W'$*. Then* $T = \{(v', v) \mid v \in S \wedge (v, v') \in R \wedge v' \in V(s)\}$ *is a bijection.*

Lemma 6. *Let* φ *be an* $\mathsf{ML}(ⓡ, ⓚ)$*-formula in PNF that does not contain the propositional symbol* s*. Then,* φ *is satisfiable iff* $\tau_{\langle\mathsf{sw}\rangle}(\varphi)$ *is satisfiable.*

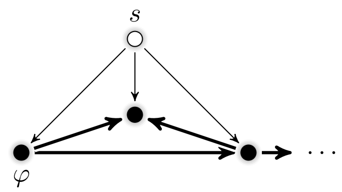
Proof. ($\Leftarrow$) From a pointed model $\langle W, R, V\rangle, w$ of $\tau_{\langle\mathsf{sw}\rangle}(\varphi)$ we can extract a pointed model $\langle W', R', V', \emptyset\rangle, w'$ satisfying φ following the same definition as in the proof of Lemma 1.

For all ψ sub-formula of φ, $v \in W'$, $S \subseteq W'$, $T = \{(v', v) \mid v \in S \wedge (v, v') \in R \wedge v' \in V(s)\}$ and $R_S = (R \backslash T^{-1}) \cup T$, we will prove that $\langle W', R', V', S\rangle, v \models \psi$ if, and only if, $\langle W, R_S, V\rangle, v \models (\psi)'$.

We do it by structural induction on ψ. We prove the $\neg ⓚ \wedge ⓡ\chi$ case. Suppose $\langle W', R', V', S\rangle, v \models \neg ⓚ \wedge ⓡ\chi$. Then by definition, $v \notin S$ and $\langle W', R', V', S \cup \{v\}\rangle, v \models \chi$, and by Proposition 5, we have $(v, v') \in R_S$ for a unique $v' \in V(s)$. Then, by definition of R_S and inductive hypothesis we get $\langle W, (R_S)^*_{v'v}, V\rangle, v \models (\chi)'$. By definition of $\models$ and by Proposition 5, $\langle W, (R_S)^*_{v'v}, V\rangle, v' \models s \wedge \Diamond(\chi)'$, and again, $\langle W, R_S, V\rangle, v \models \Diamond s \wedge \langle\mathsf{sw}\rangle(s \wedge \Diamond(\chi)')$, thus we have, equivalently, $\langle W, R_S, V\rangle, v \models (\neg ⓚ \wedge ⓡ\chi)'$.

($\Rightarrow$) Suppose $\langle W, R, V, \emptyset\rangle, w \models \varphi$. Let sw be a bijective function between W and a set U such that $U \cap W = \emptyset$, and $s \notin U \cup W$. Define $\mathsf{M}' = \langle W', R', V'\rangle$ such that $W' = W \cup \{s\} \cup U$, $R' = R \cup \{(s, w) \mid w \in W\} \cup \{(w, sw(w)) \mid w \in W\}$, $V'(s) = \{s\} \cup U$, and $V'(p) = V(p)$ for $p \in \mathsf{PROP}$ appearing in φ. It is easy to check that $\langle W', R', V'\rangle, s \models \mathit{Struct}_{\langle\mathsf{sw}\rangle}$, in particular, Proposition 5 is relevant. Then, we can easily prove that for all ψ sub-formula of φ, $v \in W$, $S \subseteq W$, $T = \{(sw(v), v) \mid v \in S\}$ and $R'_S = (R' \backslash T^{-1}) \cup T$, we have the equivalence $\langle W, R, V, S\rangle, v \models \psi$ iff $\langle W', R'_S, V'\rangle, v \models (\psi)'$. This is done by structural induction on ψ.

Global Swap. The global swap operator is able to change the direction of some edge in the model. In particular, we are interested in the ability to swap, for some state, an incoming edge (undetectable for the basic modal logic) into an outgoing edge. This is why this translation is similar to the one of global bridge logic. Initially, the model does not have any reachable state where s holds. As for global sabotage and global bridge, there may be many states where s holds in the model with edges to states of the input model. The idea is illustrated in the following image, where only one s state is shown.



Definition 13. *Define* $\tau_{\langle\mathsf{gsw}\rangle}(\varphi) = \mathit{Struct}_{\langle\mathsf{gsw}\rangle}(\varphi) \wedge \mathsf{Tr}_{\langle\mathsf{gsw}\rangle}(\varphi)$*, where:*

$$\mathit{Struct}_{\langle\mathsf{gsw}\rangle}(\varphi) = \bigwedge_{0 \le i \le \mathsf{md}(\varphi)+1} \Box^i \neg s$$

$$
\begin{array}{ll}
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(p) & = p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\text{ⓚ}) & = \Diamond s \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\neg\psi) & = \neg \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi) \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi \wedge \chi) & = \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi) \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\chi) \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\Diamond\psi) & = \Diamond(\neg s \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi)) \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\text{ⓡ}\psi) & = \langle \mathsf{gsw} \rangle(\Diamond s \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi))
\end{array}
$$

Proposition 6. *Let $\langle W, R, V\rangle, w \models \neg\Diamond s \wedge \langle \mathsf{gsw} \rangle \Diamond s$. Then, by the semantics of the global swap operator, there exists a state $v \in W \setminus \{w\}$ such that $(v, w) \in R$ and $v \in V(s)$.*

Lemma 7. *Let φ be an ML(ⓡ, ⓚ)-formula in PNF that does not contain the propositional symbol s. Then, φ is satisfiable iff $\tau_{\langle \mathsf{gsw} \rangle}(\varphi)$ is satisfiable.*

5 Conclusions

We exploited the similarities between memory logic and relation-changing logics to obtain simple and non-redundant undecidability proofs. We first presented an undecidability result for memory logics in the monomodal case, by adapting the proof introduced in [18] for $\mathcal{ALC}\mathsf{self}$. Then, we presented translations from the satisfiability problem of monomodal memory logics to all six relation-changing modal logics. Both results combined show undecidability of the satisfiability problem for relation-changing modal logics in a very simple way. These results complete the picture of the computational behaviour of relation-changing logics, given that we already know that model checking for them is PSPACE-complete [2,4,15,5].

This high complexity of the logics is a consequence of the degree of liberty we give to the operators. By replacing arbitrary modifications with conditional modifications (i.e., according to a pre- and a post-condition) it is possible to decrease the complexity and get decidable logics (e.g., as in [9,10]).

A related problem is the one of finite satisfiability. Indeed, for many applications of dynamic epistemic logic, we are only interested in looking for finite models. Finite satisfiability is known to be undecidable for multimodal global sabotage logic [20], and decidable for monomodal local sabotage and local swap logics [6]. It remains to see the status of this problem for all remaining cases.

Acknowledgements. This work was partially supported by grant ANPCyT-PICT-2013-2011, STIC-AmSud "Foundations of Graph Structured Data (FoG)", SeCyT-UNC, the Laboratoire International Associé "INFINIS", and the EU Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant No. 690974, project MIREL: MIning and REasoning with Legal texts.

References

1. C. Areces. Hybrid Logics: The Old and the New. In *Proceedings of LogKCA-07*, pages 15–29, 2007.
2. C. Areces, R. Fervari, and G. Hoffmann. Moving Arrows and Four Model Checking Results. In *Logic, Language, Information and Computation*, volume 7456 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2012.

3. C. Areces, R. Fervari, and G. Hoffmann. Tableaux for Relation-Changing Modal Logics. In *Frontiers of Combining Systems*, volume 8152 of *Lecture Notes in Computer Science*, pages 263–278, 2013.
4. C. Areces, R. Fervari, and G. Hoffmann. Swap Logic. *Logic Journal of the IGPL*, 22(2):309–332, 2014.
5. C. Areces, R. Fervari, and G. Hoffmann. Relation-Changing Modal Operators. *Logic Journal of the IGPL*, 23(4):601–627, 2015.
6. C. Areces, R. Fervari, G. Hoffmann, and M. Martel. Relation-Changing Logics as Fragments of Hybrid Logics. In D. Cantone and G. Delzanno, editors, *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Italy*, volume 226 of *EPTCS*, pages 16–29, 2016.
7. 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.
8. C. Areces and B. ten Cate. Hybrid Logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logic*, pages 821–868. Elsevier, 2007.
9. C. Areces, H. van Ditmarsch, R. Fervari, and F. Schwarzentruber. Logics with Copy and Remove. In *Logic, Language, Information, and Computation*, volume 8652 of *Lecture Notes in Computer Science*, pages 51–65. Springer, 2014.
10. C. Areces, H. van Ditmarsch, R. Fervari, and F. Schwarzentruber. The Modal Logic of Copy and Remove. *To Appear in Information and Computation, special issue of WoLLIC 2014*, 2015.
11. G. Aucher and F. Schwarzentruber. On the complexity of dynamic epistemic logic. In *Proceedings of TARK 2013*, Chennai, India, January 2013.
12. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.
13. P. Blackburn and J. Seligman. Hybrid Languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.
14. P. Blackburn and J. van Benthem. Modal Logic: A Semantic Perspective. In *Handbook of Modal Logic*, pages 1–84. Elsevier, 2007.
15. R. Fervari. *Relation-Changing Modal Logics*. PhD thesis, Universidad Nacional de Córdoba, Argentina, 2014.
16. C. Löding and P. Rohde. Model Checking and Satisfiability for Sabotage Modal Logic. volume 2914 of *Lecture Notes in Computer Science*, pages 302–313, 2003.
17. C. Lutz. Complexity and Succinctness of Public Announcement Logic. In *Proceedings of the 5th International joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 137–143, New York, NY, USA, 2006.
18. M. Marx. Narcissists, stepmothers and spies. In *Proceedings of DL02, volume 53. CEUR*, 2002.
19. S. Mera. *Modal Memory Logics*. PhD thesis, Université Henri Poincaré, Nancy, France and Universidad de Buenos Aires, Argentina, 2009.
20. P. Rohde. *On games and logics over dynamically changing structures*. PhD thesis, RWTH Aachen, 2006.
21. J. van Benthem. An Essay on Sabotage and Obstruction. In *Mechanizing Mathematical Reasoning*, pages 268–276, 2005.
22. H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Synthese Library. Springer, 2007.