

The Expressive Power of Swap Logic

Raul Fervari

FaMAF, Universidad Nacional de Córdoba, Argentina
fervari@famaf.unc.edu.ar

Abstract. Modal logics are appropriate to describe properties of graphs. But usually these are *static* properties. We investigate dynamic modal operators that can change the model during evaluation. We define the logic \mathcal{SL} by extending the basic modal language with the \diamond modality, which is a diamond operator that has the ability to invert pairs of related elements in the domain while traversing an edge of the accessibility relation. We will investigate the expressive power of \mathcal{SL} , define a suitable notion of bisimulation and compare \mathcal{SL} with other dynamic logics.

Keywords: Modal logic, dynamic logics, hybrid logic, bisimulation, expressivity.

1 Introduction

There are many notions in language and science that have a *modal* character, e.g. the classical notion of necessity and possibility. Modal logics [5, 6] are logics designed to deal with these notions. In general, they are adequate to describe certain patterns of behaviour of the real world. For this reason, modal logics are not just useful in mathematics or computer science; they are used in philosophy, linguistics, artificial intelligence and game theory, to name a few.

Intuitively, modal logics extend the classical logical systems with operators that represent the modal character of some situation. In particular, the basic modal logic (\mathcal{BML}) is an extension of propositional logic, with a new operator. We now define formally the syntax and semantics of \mathcal{BML} .

Definition 1 (Syntax). Let PROP be an infinite, countable set of propositional symbols. The set FORM of \mathcal{BML} formulas over PROP is defined as:

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

where $p \in \text{PROP}$ and $\varphi, \psi \in \text{FORM}$. We use $\Box\varphi$ as a shorthand for $\neg\diamond\neg\varphi$, while \perp , \top and $\varphi \vee \psi$ are defined as usual.

Definition 2 (Semantics). A model \mathcal{M} is a triple $\mathcal{M} = \langle W, R, V \rangle$, where W is a non-empty set; $R \subseteq W \times W$ is the accessibility relation; and $V : \text{PROP} \rightarrow \mathcal{P}(W)$ is a valuation. Let w be a state in \mathcal{M} , the pair (\mathcal{M}, w) is called a pointed model; we will usually drop parenthesis and write \mathcal{M}, w . Given a pointed model \mathcal{M}, w and a formula φ we say that \mathcal{M}, w satisfies φ ($\mathcal{M}, w \models \varphi$) when

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

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

As shown in Definition 2, modal logics describe characteristics of relational structures. But these are *static* characteristics of the structure, i.e. properties never change after the application of certain operations. If we want to describe *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 logic with classical semantics) becomes less clear. We can always resort to modeling the whole space of possible evolutions of the system as a graph, but this soon becomes unwieldy. It would be more elegant to use truly dynamic modal logics with operators that can mimic the changes that structure will undergo. This is not a new idea, and a clear example of this kind of logics is the *sabotage logic* $\mathcal{SM}\mathcal{L}$ introduced by van Benthem in [4].

Consider the following *sabotage game*. It is played on a graph with two players, Runner and Blocker. Runner can move on the graph from node to accessible node, starting from a designated point, and with the goal of reaching a given final point. He should move one edge at a time. Blocker, on the other hand, can delete one edge from the graph, every time it is his turn. Of course, Runner wins if he manages to move from the origin to the final point in the graph, while Blocker wins otherwise. van Benthem proposes transforming the sabotage game into a modal logic; this idea has been studied in several other works [8, 11]. In particular, they defined the operator of sabotage as:

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

where $\mathcal{M}_E = \langle W, R \setminus E, V \rangle$, with $E \subseteq R$.

It is clear that the \diamond operator *changes* the model in which a formula is evaluated. As van Benthem puts it, \diamond is an “external” modality that takes evaluation to another model, obtained from the current one by deleting some transition. It has been proved that solving the sabotage game is PSpace-hard [4], while the model checking problem of the associated modal logic is PSpace-complete and the satisfiability problem is undecidable [8, 9]. It has been investigated in these articles that the logic fails to have two nice model theoretical properties: *the finite model property* (if a formula is satisfiable, then it is satisfiable in a finite model) and *the tree model property* (every satisfiable formula is satisfied in the root of a tree-like model).

Another family of model changing logics is memory logics [1, 3, 10]. The semantics of these languages is specified on models that come equipped with a set of states called the *memory*. The simplest memory logic includes a modality $\textcircled{\mathbb{R}}$ that *stores* the current point of evaluation into memory, and a modality $\textcircled{\mathbb{K}}$ that verifies whether the current state of evaluation has been memorized. The memory can be seen as a special proposition symbol whose extension grows whenever the $\textcircled{\mathbb{R}}$ modality is used. In contrast with sabotage logics, the basic memory logic *expands* the model with an ever increasing set of memorized elements. The general properties of memory logics are similar to those of sabotage logics: a PSpace-complete model checking problem, an undecidable satisfiability problem, and failure of both the finite model and the tree model properties.

In this article, we will investigate a model changing operator that in the general case doesn’t shrink nor expand the model. Instead, it has the ability to swap the direction of a traversed arrow. The $\overleftarrow{\diamond}$ operator is a \diamond operator — to be true at a state w it requires the existence of an accessible state v where evaluation will continue— but it changes the accessibility relation during evaluation —the pair (w, v) is deleted, and the pair (v, w) is added to the accessibility relation.

A picture will help understand the dynamics of $\overleftarrow{\diamond}$. The formula $\overleftarrow{\diamond}\diamond\top$ is true in a model with two related states:



As we can see in the picture, evaluation starts at state w with the arrow pointing from w to v , but after evaluating the $\overleftarrow{\diamond}$ operator, it continues at state v with the arrow now pointing from v to w . In this article, we will study the expressive power of \mathcal{SL} and will compare it with \mathcal{BML} and some dynamic logics.

2 Swap Logic

Now we introduce syntax and semantics for \mathcal{SL} . We will define first some notation that will help us describe models with swapped accessibility relations.

Definition 3. Let R and S be two binary relations over a set W . We define the relation $R^S = (R \setminus S^{-1}) \cup S$. When S is a singleton $\{(v, w)\}$ we write R^{vw} instead of $R^{\{(v, w)\}}$.

Intuitively R^S stands for R with some edges swapped around (S contains the edges in their final position). The following property is easy to verify:

Proposition 1. Let $R, S, S' \subseteq W^2$ be binary relations over an arbitrary set W , then $(R^S)^{S'} = R^{S^{S'}}$.

We extend \mathcal{BML} with a new operator $\overleftarrow{\diamond}$. For $(v, w) \in R^{-1}$, let $\mathcal{M}^{vw} = \langle W, R^{vw}, V \rangle$. Similarly, for $S \subseteq R^{-1}$, $\mathcal{M}^S = \langle W, R^S, V \rangle$. Then we define the semantics of the new operator as follows:

$$\mathcal{M}, w \models \overleftarrow{\diamond}\varphi \text{ iff for some } v \in W \text{ s.t. } (w, v) \in R, \mathcal{M}^{vw}, v \models \varphi.$$

The semantic condition for $\overleftarrow{\diamond}$ looks quite innocent but, as we will see in the following example, it is actually very expressive.

Example 1. Define $\Box^0\varphi$ as φ , $\Box^{n+1}\varphi$ as $\Box\Box^n\varphi$, and let $\Box^{(n)}\varphi$ be a shorthand for $\bigwedge_{1 \leq i \leq n} \Box^i\varphi$. The formula $\varphi = p \wedge \Box^{(3)}\neg p \wedge \overleftarrow{\diamond}\diamond p$ is true at a state w in a model, only if w has a reflexive successor. Notice that no equivalent formula exists in the basic modal language (satisfiable formulas in the basic modal language can always be satisfied at the root of a tree model).

Let us analyse the formula in detail. Suppose we evaluate φ at some state w of an arbitrary model. The ‘static’ part of the formula $p \wedge \Box^{(3)}\neg p$ makes sure that p is true in w and that no p state is reachable within a three steps neighbourhood of w (in particular, the evaluation point cannot be reflexive). Now, the ‘dynamic’ part of the formula $\overleftarrow{\diamond}\diamond p$ will do the rest. Because $\overleftarrow{\diamond}\diamond p$ is true at w , there should be an R -successor v where $\diamond p$ holds once the accessibility relation has been updated to R^{vw} . Now, v has to reach a p -state in exactly two R^{vw} -steps to satisfy $\diamond p$. But the only p state sufficiently close for this to happen is w which is reachable in one step. As w is not reflexive, v has to be reflexive so that we can linger at v for one loop and reach p in the correct number of states.

Example 1 shows that the tree model property fails for \mathcal{SL} . It should be a warning about the expressivity of \mathcal{SL} , which is certainly above that of the \mathcal{BML} .

3 Bisimulation and Expressive Power

In most modal logics, bisimulations are binary relations linking elements of the domains that have the same atomic information, and preserving the relational structure of the model [6]. This will not suffice for \mathcal{SL} where we also need to capture the dynamic behaviour of the $\overleftarrow{\diamond}$ operator. The proper notion of \mathcal{SL} -bisimulations links states together with the context of potentially swapped edges.

Definition 4 (Swap Bisimulations). *Given models $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$, together with points $w \in W$ and $w' \in W'$ we say that they are bisimilar and write $\mathcal{M}, w \approx \mathcal{M}', w'$ if there is a relation $Z \subseteq (W \times \mathcal{P}(W^2)) \times (W' \times \mathcal{P}(W'^2))$ such that $(w, R)Z(w', R')$ satisfying the following conditions. Whenever $(w, S)Z(w', S')$ then*

- (Atomic Harmony) for all $p \in \text{PROP}$, $\mathcal{M}, w \models p$ iff $\mathcal{M}', w' \models p$;
- (Zig) If wSv , there is $v' \in W'$ s.t. $w'S'v'$ and $(v, S)Z(v', S')$;
- (Zag) If $w'S'v'$, there is $v \in W$ s.t. wSv and $(v, S)Z(v', S')$;
- (S-Zig) If wSv , there is $v' \in W'$ s.t. $w'S'v'$ and $(v, S^{vw})Z(v', S'^{v'w'})$;
- (S-Zag) If $w'S'v'$, there is $v \in W$ s.t. wSv and $(v, S^{vw})Z(v', S'^{v'w'})$.

Theorem 1 (Invariance for Swap Logic.). *Let $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$ two models, $w \in W$, $w' \in W'$, $S \subseteq R^{-1}$ and $S' \subseteq R'^{-1}$. If $\mathcal{M}, w \approx \mathcal{M}', w'$ then for any formula $\varphi \in \mathcal{SL}$, $\mathcal{M}^S, w \models \varphi$ iff $\mathcal{M}'^{S'}, w' \models \varphi$.*

Proof. The proof is by structural induction on \mathcal{SL} formulas. The base case holds by Atomic Harmony, and the \wedge and \neg cases are trivial.

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

[$\overleftarrow{\diamond}\varphi$ case:] For the left to the right direction suppose $\mathcal{M}, w \models \overleftarrow{\diamond}\varphi$. Then there is $v \in W$ s.t. wRv and $\mathcal{M}^{vw}, v \models \varphi$. Because Z is a bisimulation, by (S-zig) we have $v' \in W'$ s.t. $w'R'v'$ and $(v, R^{vw})Z(v', R'^{v'w'})$. By inductive hypothesis, $\mathcal{M}'^{v'w'}, v' \models \varphi$ and by definition $\mathcal{M}', w' \models \overleftarrow{\diamond}\varphi$. For the other direction use (S-zag). \square

Example 2. The two models in row A of Table 1 are \mathcal{SL} -bisimilar. The simplest way to check this is to recast the notion of \mathcal{SL} -bisimulation as an Ehrenfeucht-Fraïssé game as the one used for BML , but where Spoiler can also swap arrows when moving from a node to an adjacent node. It is clear that Duplicator has a winning strategy.

We are now ready to investigate the expressive power of \mathcal{SL} .

Definition 5 ($\mathcal{L} \leq \mathcal{L}'$). *We say that \mathcal{L}' is at least as expressive as \mathcal{L} (notation $\mathcal{L} \leq \mathcal{L}'$) if there is a function Tr between formulas of \mathcal{L} and \mathcal{L}' such that for every model \mathcal{M} and every formula φ of \mathcal{L} we have that $\mathcal{M} \models_{\mathcal{L}} \varphi$ iff $\mathcal{M} \models_{\mathcal{L}'} \text{Tr}(\varphi)$. \mathcal{M} is seen as a model of \mathcal{L} on the left and as a model of \mathcal{L}' on the right, and we use in each case the appropriate semantic relation $\models_{\mathcal{L}}$ or $\models_{\mathcal{L}'}$ as required.*

We say that \mathcal{L}' is strictly more expressive than \mathcal{L} (notation $\mathcal{L} < \mathcal{L}'$) if $\mathcal{L} \leq \mathcal{L}'$ but not $\mathcal{L}' \leq \mathcal{L}$. And we say that \mathcal{L} and \mathcal{L}' are uncomparable (notation $\mathcal{L} \not\leq \mathcal{L}'$) if $\mathcal{L} \not\leq \mathcal{L}'$ and $\mathcal{L}' \not\leq \mathcal{L}$.

A	
B	
C	
D	

Table 1. Table of comparison between models.

The definition above requires that both logics evaluate their formulas over the same class of models. However, we can abuse of this detail when operators of some logic do not interact with the part of the model that is different of the other one. For example, memory logics operate over models that contains an additional set to store visited states (a memory), and in our results we will evaluate formulas with an initial empty memory. When we evaluate \mathcal{SL} -formulas over memory logics models with an empty memory, we can as well forget about that memory and treat them as standard Kripke models. The same occurs with hybrid models (where there are nominals and \downarrow operator in the language).

Our first result is fairly straightforward as it builds upon Example 3: \mathcal{BML} is strictly less expressive than \mathcal{SL} .

Theorem 2. $\mathcal{BML} < \mathcal{SL}$.

Proof. We have to provide a translation from \mathcal{BML} formulas to \mathcal{SL} . This is trivial as \mathcal{BML} is a fragment of \mathcal{SL} . To prove $\mathcal{SL} \not\leq \mathcal{BML}$ consider the models in row B of Table 1. They are bisimilar for \mathcal{BML} but, as we already mentioned, the \mathcal{SL} formula $\overleftarrow{\diamond}\diamond\Box\perp$ distinguishes them. \square

Now we will compare \mathcal{SL} with the hybrid logic $\mathcal{H}(\cdot, \downarrow)$, whose operator \downarrow is a dynamic operator.

Definition 6 (Hybrid Logic $\mathcal{H}(\cdot, \downarrow)$). Let the signature $\langle \text{PROP}, \text{NOM} \rangle$ be given, with $\text{NOM} \subseteq \text{PROP}$. We extend \mathcal{BML} with two new operators $n:\varphi$ and $\downarrow n.\varphi$, where $p \in \text{PROP}$, $n \in \text{NOM}$ and $\varphi, \psi \in \text{FORM}$.

A hybrid model \mathcal{M} is a triple $\langle W, R, V \rangle$ as usual, but $V : \text{PROP} \rightarrow \mathcal{P}(W)$ is such that $V(p)$ is a singleton if $p \in \text{NOM}$. Let w be a state in \mathcal{M} , the semantics of the new operators is defined as:

$$\begin{aligned} \langle W, R, V \rangle, w \models n:\varphi & \text{ iff } \langle W, R, V \rangle, v \models \varphi \text{ where } V(n) = \{v\} \\ \langle W, R, V \rangle, w \models \downarrow n.\varphi & \text{ iff } \langle W, R, V_n^w \rangle, w \models \varphi, \end{aligned}$$

where V_n^w is defined by $V_n^w(n) = \{w\}$ and $V_n^w(m) = V(m)$ when $m \neq n$.

Formulas like $\downarrow n.\varphi$, can be intuitively read as “after naming the current state n , φ holds”. For instance, $\downarrow n.\diamond n$ means “the current state is reflexive”, and $\Box\downarrow n.\diamond n$ means “all accessible states are reflexive”

Theorem 3. $\mathcal{SL} < \mathcal{H}(\cdot, \downarrow)$.

Proof. We will define an adequate translation from \mathcal{SL} to $\mathcal{H}(\cdot, \downarrow)$. As models of $\mathcal{H}(\cdot, \downarrow)$ have a fixed frame, we will encode the configuration of the model (i.e., the current state of swapped links) at the syntactic level. To do so, we will take advantage of the binder \downarrow and nominals to name the pair of points of the model where a swap should occur.

Let F be a non-empty set of pairs of nominals. Intuitively, this set will play a similar role than the set S of Definition 3. However, it contains purely syntactic information in the form of pair of nominals, as opposed to a direct description of swapped edge of the model. Moreover, it stores swapped links in the order of their former state. That is, F (as in “forbidden”) represents the set of links that *cannot* be taken any more. We use F to keep track of the configuration of the model in which a subformula must be satisfied. In fact, we will make heavy use of nominals to exactly determine if there are links that we should not take, when translating modal logic connectors.

The complexity of the translation lies in the operators \diamond and $\overleftarrow{\diamond}$. The translation of $\diamond\varphi$ subformulas considers two cases:

- either the diamond is satisfied using a successor by a link that either has not been swapped, or is a swapped reflexive link. To know which accessible worlds in the hybrid model should not be accessible in the swap model, we first have to determine where we are. That is, for every forbidden pair $(x, y) \in F$, decide if x is true or not at the current point. Then for all true x , we enforce that either $\neg y$ is true at the destination point, or x is true (for the reflexive case). Of course, the translation of φ has to be true at that destination point.
- or the diamond is satisfied by taking a swapped edge (y, x) . In this case the current point of evaluation should be y and we should continue evaluation at x .

The cases for $\overleftarrow{\diamond}$ subformulas are similar, but we should record that the used link is now swapped:

- either $\overleftarrow{\diamond}$ is satisfied by traversing a new edge, in which case we name it (x, y) and we add it to the set F .
- or $\overleftarrow{\diamond}$ is satisfied by taking an already swapped edge $(x, y) \in F$, in which case the current point of evaluation is y , we should continue at x and remove (x, y) from F .

Formally then, the equivalence-preserving translation from \mathcal{SL} to $\mathcal{H}(\cdot, \downarrow)$ is defined as follows:

Definition 7. Let $F \subseteq \text{NOM} \times \text{NOM}$. Define $(\cdot)'_F$ from formulas of \mathcal{SL} to formulas of $\mathcal{H}(\cdot, \downarrow)$ as

Let $F \subseteq \text{NOM} \times \text{NOM}$. Define $(\cdot)'_F$ from formulas of \mathcal{SL} to formulas of $\mathcal{H}(\cdot, \downarrow)$ as

$$\begin{aligned}
(p)'_F &= p \\
(\neg\varphi)'_F &= \neg(\varphi)'_F \\
(\psi \wedge \varphi)'_F &= (\psi)'_F \wedge (\varphi)'_F \\
(\diamond\varphi)'_F &= \bigwedge_{c \in \mathcal{P}(F)} (\bigwedge_{xy \in F} (\neg)_{xy \notin c} x) \rightarrow \diamond((\bigwedge_{xy \in c} \neg y \vee x) \wedge (\varphi)'_F) \\
&\quad \vee \bigvee_{xy \in F} y \wedge x: (\varphi)'_F \\
(\overleftarrow{\diamond}\varphi)'_F &= \bigwedge_{c \in \mathcal{P}(F)} (\bigwedge_{xy \in F} (\neg)_{xy \notin c} x) \rightarrow \downarrow i \diamond (\downarrow j (\bigwedge_{xy \in c} \neg y \vee x) \wedge (\varphi)'_{F \cup ij}) \\
&\quad \vee \bigvee_{xy \in F} y \wedge x: (\varphi)'_{F \setminus xy}
\end{aligned}$$

where i and j are nominals that do not appear in F .

We introduced two \mathcal{SL} -bisimilar models in Example 2. Finally the formula $\downarrow x.\diamond\neg x$ distinguishes them, being true in \mathcal{M}_2, w' and false in \mathcal{M}_1, w . \square

Summing up, \mathcal{SL} is strictly in between \mathcal{BML} and $\mathcal{H}(\cdot, \downarrow)$. Let us now compare \mathcal{SL} with a family of dynamic modal logics called *memory logics*. Memory logics [1, 10] are modal logics with the ability to *store* the current state of evaluation into a set, and to consult whether the current state of evaluation belongs to this set. This set is also called the memory.

Definition 8 (Memory Logics). *The set FORM of formulas of $\mathcal{ML}(\mathbb{F}, \mathbb{K})$ over PROP is defined as in Definition 1 but adding a new zero-ary operator \mathbb{K} and a new unary operator $\mathbb{F}\varphi$. The same holds for $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$, but adding $\langle\langle r \rangle\rangle\varphi$ and deleting the classical \diamond .*

A model $\mathcal{M} = \langle W, R, V, S \rangle$ is an extension of an \mathcal{SL} model with a memory $S \subseteq W$. Let w be a state in \mathcal{M} , we define satisfiability as:

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

A formula φ of $\mathcal{ML}(\mathbb{F}, \mathbb{K})$ or $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$ is satisfiable if there exists a model $\langle W, R, V, \emptyset \rangle$ such that $\langle W, R, V, \emptyset \rangle, w \models \varphi$.

In the definition of satisfiability, the empty initial memory ensures that no point of the model satisfies the unary predicate \mathbb{K} unless a formula $\mathbb{F}\varphi$ or $\langle\langle r \rangle\rangle\varphi$ has previously been evaluated there. The memory logic $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$ does not have the \diamond operator, and its expressive power is strictly weaker than $\mathcal{ML}(\mathbb{F}, \mathbb{K})$ [3, 10]. We now show that the expressive power of \mathcal{SL} is uncomparable with both $\mathcal{ML}(\mathbb{F}, \mathbb{K})$ and $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$.

Theorem 4. $\mathcal{SL} \not\geq \mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$.

Proof. As we mentioned, no \mathcal{SL} formula distinguishes the models in row A of Table 1, but $\langle\langle r \rangle\rangle\neg\mathbb{K}$ is satisfiable in \mathcal{M}_2, w' but not in \mathcal{M}_1, w . \square

Theorem 5. $\mathcal{ML}(\mathbb{F}, \mathbb{K}) \not\geq \mathcal{SL}$.

Proof. The models in row C of Table 1 are bisimilar in $\mathcal{ML}(\mathbb{F}, \mathbb{K})$. Indeed they are \mathcal{BML} bisimilar and acyclic, hence \mathbb{K} is always false after taking an accessibility relation.

The formula $\overset{\leftarrow}{\diamond}\overset{\leftarrow}{\diamond}\overset{\leftarrow}{\diamond}\overset{\leftarrow}{\diamond}\square\perp$ is satisfiable in \mathcal{M}_2, w' but not in \mathcal{M}_1, w . \square

Corollary 1. *The expressive powers of $\mathcal{ML}(\mathbb{F}, \mathbb{K})$ and \mathcal{SL} are uncomparable. The same holds for $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$ and \mathcal{SL} .*

Proof. From Theorems 4 and 5, given that $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K}) < \mathcal{ML}(\mathbb{F}, \mathbb{K})$ [3]. \square

We now compare the expressive powers of $\mathcal{ML}(\mathbb{F}, \mathbb{K})$ and \mathcal{SL} with the sabotage modal logic \mathcal{SML} .

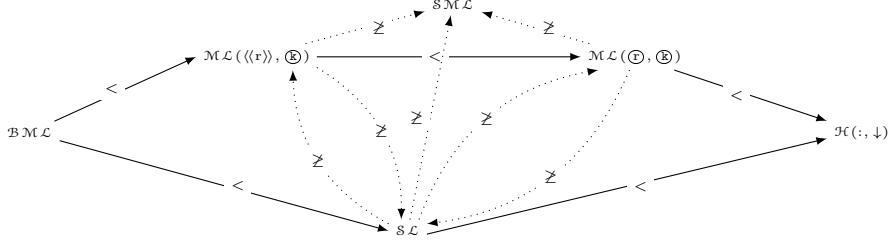
Theorem 6. $\mathcal{SL} \not\geq \mathcal{SML}$.

Proof. Consider again models in Example 2, that are \mathcal{SL} -bisimilar. The formula $\diamond\diamond\top$ distinguishes them (it is satisfiable in \mathcal{M}_2, w' but not in \mathcal{M}_1, w). \square

Theorem 7. $\mathcal{ML}(\mathbb{R}, \mathbb{K}) \not\cong \mathcal{SM}\mathcal{L}$.

Proof. Models in row D of Table 1 are bisimilar in $\mathcal{ML}(\mathbb{R}, \mathbb{K})$, but the formula $\diamond\diamond\top$ is satisfiable in \mathcal{M}_2, w' but not in \mathcal{M}_1, w . \square

The following picture sums up the results of this section (relationships between $\mathcal{ML}(\mathbb{R}, \mathbb{K})$, $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$ and $\mathcal{H}(\cdot, \downarrow)$ have been extracted from [3]).



It remains an open question the other directions of the comparison between $\mathcal{SM}\mathcal{L}$ and the other logics, because no notion of bisimulation for $\mathcal{SM}\mathcal{L}$ has been provided yet.

4 Conclusions

In this paper we have extended the basic modal language with the $\overleftarrow{\diamond}$ operator, which is a diamond operator that has the ability to invert pairs of related elements in the domain while traversing an edge of the accessibility relation. Other dynamic languages that can modify the model have been investigated in the literature (e.g., sabotage logics [4, 8], memory logics [3, 10], hybrid logics [2, 7]), and we have discussed the relation between these languages and $\mathcal{S}\mathcal{L}$. In particular, we have introduced an adequate notion of bisimulation for $\mathcal{S}\mathcal{L}$ and used it to show that the expressive power of $\mathcal{S}\mathcal{L}$ lies strictly in between the expressive powers of $\mathcal{B}\mathcal{M}\mathcal{L}$ and the hybrid logic $\mathcal{H}(\cdot, \downarrow)$, while it is incomparable with the expressive powers of the memory logics $\mathcal{ML}(\mathbb{R}, \mathbb{K})$ and $\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{K})$.

Many theoretical aspects of $\mathcal{S}\mathcal{L}$ remain to be investigated. For example, it would be interesting to obtain an axiomatic characterization which is sound and complete. The task is probably non trivial, as the logic fails to be closed under uniform substitution. A proper axiomatization will require an adequate definition of when a formula is free to substitute another formula in an axiom. More generally, it is a challenge to study the properties of other dynamic operators beside $\overleftarrow{\diamond}$. For example, two operators which would be closer to those investigated by sabotage logics would be one that access an arbitrary element in the model (similar to a global modality) and makes it a successor of the current point of evaluation, complemented with one that access a successor of the current point of evaluation and deletes the edge between the two. Investigating further examples of this kind of operator will let us have a clearer picture of the gains and losses of working with logics which can dynamically modify the model.

Acknowledgments:

This work was partially supported by grants ANPCyT-PICT-2008-306, ANPCyT-PIC-2010-688, the FP7-PEOPLE-2011-IRSES Project “Mobility between Europe and Argentina applying Logics to Systems” (MEALS) and the Laboratoire Internationale Associé “INFINIS”.

References

1. Areces, C.: Hybrid logics: The old and the new. In: Arrazola, X., Larrazabal, J. (eds.) Proc. of LogKCA-07. pp. 15–29. San Sebastian, Spain (2007)
2. Areces, C., ten Cate, B.: Hybrid logics. In: Blackburn, P., Wolter, F., van Benthem, J. (eds.) Handbook of Modal Logics, pp. 821–868. Elsevier (2006)
3. Areces, C., Figueira, D., Figueira, S., Mera, S.: The expressive power of memory logics. *The Review of Symbolic Logic* 4(2), 290–318 (2011)
4. van Benthem, J.: An essay on sabotage and obstruction. In: *Mechanizing Mathematical Reasoning*. pp. 268–276 (2005)
5. Blackburn, P., van Benthem, J.: Modal logic: A semantic perspective. In: *Handbook of Modal Logic*. Elsevier North-Holland (2006)
6. Blackburn, P., de Rijke, M., Venema, Y.: *Modal Logic*, Cambridge Tracts in Theoretical Comp. Scie., vol. 53. Cambridge University Press (2001)
7. Blackburn, P., Seligman, J.: Hybrid languages. *Journal of Logic, Language and Information* 4, 251–272 (1995)
8. Löding, C., Rohde, P.: Model checking and satisfiability for sabotage modal logic. In: Pandya, P., Radhakrishnan, J. (eds.) FSTTCS. LNCS, vol. 2914, pp. 302–313. Springer (2003)
9. Löding, C., Rohde, P.: Solving the sabotage game is PSPACE-hard. In: *Mathematical Foundations of Computer Science 2003*, Lecture Notes in Computer Science, vol. 2747, pp. 531–540. Springer, Berlin (2003)
10. Mera, S.: *Modal Memory Logics*. Ph.D. thesis, Univ. de Buenos Aires and UFR STMIA - Ecole Doctorale IAEM Lorraine Dép. de Form. Doct. en Informat. (2009)
11. Rohde, P.: *On games and logics over dynamically changing structures*. Ph.D. thesis, RWTH Aachen (2006)