On the Complexity of Modal Separation Logics

Stephane Demri, Raul Fervari

On the Complexity of Modal Separation Logics

Stéphane Demri

LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, France

Raul Fervari

FAMAF, Universidad Nacional de Córdoba & CONICET, Argentina

Abstract

We introduce a modal separation logic MSL whose models are memory states from separation logic and the logical connectives include modal operators as well as separating conjunction and implication from separation logic. With such a combination of operators, some fragments of MSL can be seen as genuine modal logics whereas some others capture standard separation logics, leading to an original language to speak about memory states. We analyse the decidability status and the computational complexity of several fragments of MSL, leading to surprising results, obtained by designing proof methods that take into account the modal and separation features of MSL. For example, the satisfiability problem for the fragment of MSL with $\Diamond$, the inequality modality $\langle \neq \rangle$ and separating conjunction $*$ is shown TOWER-complete whereas the restriction either to $\Diamond$ and $*$ or to $\langle \neq \rangle$ and $*$ is only NP-complete.

Keywords: separation logics, relation-changing logics, satisfiability, model-checking, complexity, expressive power.

1 Introduction

Combining modalities and separating connectives. Separation logic is known as an assertion language to perform verification, by extending Hoare-Floyd logic in order to verify programs with mutable data structures [27,34]. Local reasoning is a key feature of separation logic and the separating conjunction $*$ allows us to state properties in disjoint parts of the memory. Moreover, the separating implication $-\!\!*$ asserts that whenever a fresh heap satisfies a property, its composition with the current heap satisfies another property. Hence, the separating connectives $*$ and $-\!\!*$ allow us to evaluate formulae in alternative models, which is a feature shared with many modal logics such as sabotage logics [39,29], logics of public announcements (see e.g., [30]), interval temporal logics [26] or relation-changing logics [4,1].

Many other examples of such logics can be found in the literature (see also [18]) but the modalities involved in such logics can be of a different nature. For instance, combinations of epistemic logics and abstract separation

logics (such as variants of BI) can be found in [17,23]. Sometimes, the concept of separation is different and perform at a different level, for instance a simple separation logic is introduced in [25] in which separation is performed on valuations instead of being performed on heaps. A slightly different approach including description logics [5] was investigated in [24,15]. An interesting attempt to get a logic (namely CT^2) that captures both a very expressive description logic and a separation logic (the symbolic heap fragment) can be found in [15].

Our motivations. Most existing logics combining (epistemic, temporal, etc.) modalities and separating connectives are multi-dimensional logics and the modal dimension is often orthogonal with the separation dimension (see e.g. [10,17,23]), which allows to get proof methods combining adequately the modal part and the separation part. Our intention in this work is to introduce a modal separation logic whose models are Kripke-style structures that can be also viewed as memory states from separation logic, without being multi-dimensional. As a gain, it is possible to study the computational effects of the interaction between modalities and separating connectives but within a uniform framework and to push further the expressive power of the underlying modal logics as well as the expressive power of the underlying separation logics. Adding modalities to separation logics happens to be an original means to work on fragments of first-order separation logics. So, the logic MSL introduced herein can be understood as a *hybrid* separation logic, by analogy to hybrid versions of modal logics [7]. Note that a hybrid extension of Boolean BI is defined in [13], in which nominals are interpreted by heaps whereas herein, the nominals are interpreted by locations.

Our contributions. We introduce the logic MSL whose models are Kripke-style structures with domain $\mathbb{N}$ (understood as the set of locations) and the accessibility relation is finite and functional (understood as some heap $\mathfrak{h} : \mathbb{N} \rightarrow_{fin} \mathbb{N}$). In MSL, the modal connectives are $\Diamond$ and the inequality modality $\langle \neq \rangle$ [19] whereas the separating connectives are the separating conjunction $*$ and separating implication $-\!\!*$ (also known as the magic wand operator). These connectives allow to update dynamically the model under evaluation. Therefore, in MSL, $\Diamond$ provides a means to move within the model following the accessibility relation, $\langle \neq \rangle$ adds the possibility to jump to (almost) any location of the model, and the connectives $*$ and $-\!\!*$, removes or adds edges in the model respectively. The closest logic to MSL is probably the modal logic of heaps MLH [21] since they share the same class of frames. However, there are differences, notably MSL has propositional variables (unlike MLH whose atomic formulae are truth constants) and MSL does not contain the converse modality and the reflexive transitive closure modality. Moreover, MSL shares with some logics from [28,12] the feature of having propositional variables whose interpretation is unrestricted but in such logics, the propositional variables are interpreted as sets of memory states whereas in MSL, the variables are interpreted as sets of locations, as usual for modal logics.

- MSL restricted to $\Diamond$ and $*$, written $\mathrm{MSL}(*, \Diamond)$, can be viewed as the min-

imal modal separation logic as it witnesses a simple interaction between $\Diamond$ and, on the other side $*$ and $\mathtt{emp}$ (formula stating that the heap domain is empty). By showing a small model property, we establish that the satisfiability problem for MSL$(*, \Diamond)$ is NP-complete. The same result is shown for MSL$(*, \langle\neq\rangle)$ by adapting arguments for the logic of elsewhere [36,20]. To obtain the NP upper bound, we need to show that underlying model-checking problems are in P, which requires a refined analysis as the model checking problem for propositional separation logic (even restricted to $*$) is already PSPACE-complete [14].

- As far as decidability is concerned, we show that the satisfiability problem for MSL$(*, \Diamond, \langle\neq\rangle)$ is decidable by translation into the weak monadic second-order theory of one unary function shown decidable in [33]. This extends the decidability proof of 1SL$(*)$ from [11] as, now, propositional variables need to be taken into account. More surprisingly, even though both MSL$(*, \Diamond)$ and MSL$(*, \langle\neq\rangle)$ are NP-complete, we establish that the satisfiability problem for MSL$(*, \Diamond, \langle\neq\rangle)$ is TOWER-hard by reduction from the nonemptiness problem for star-free expressions [31,37,35]. To do so, we show an essential property: the formula $\exists\, \mathtt{x}, \mathtt{y}\ \mathtt{ls}(\mathtt{x}, \mathtt{y})$ from separation logic (see e.g. [6,16]) can be expressed in MSL$(*, \Diamond, \langle\neq\rangle)$, which allows us to encode finite words. The notion of TOWER-completeness is borrowed from [35].
- Using the fact that $\mathtt{ls}(\mathtt{x}, \mathtt{y})$ can be expressed in MSL$(*, \Diamond, \langle\neq\rangle)$ we also establish that MSL (i.e. MSL$(*, \Diamond, \langle\neq\rangle)$ augmented with the magic wand $-\!\!*$) admits an undecidable satisfiability problem by using the recent result from [22] about the undecidability of propositional separation logic (with $*$ and $-\!\!*$) augmented with the list segment predicate $\mathtt{ls}$.
- Along the paper, we also investigate variants of MSL (or some of its fragments) by slightly modifying the semantics or by adding other modal connectives. For instance, we provide a reduction from the satisfiability problem for MSL$(*, \Diamond)$ when the models are arbitrary countable Kripke-style models into global sabotage logic over general models [3].

2 Preliminaries

In this section we introduce the modal separation logic MSL, as well as several fragments that we briefly compare with propositional separation logic.

2.1 Modal separation logic MSL

Let PROP $= \{p_1, q_1, p_2, q_2, \ldots\}$ be a countably infinite set of propositional variables. Formulae for the logic MSL are defined by the grammar below:

$$\phi ::= p \mid \mathtt{emp} \mid \neg\phi \mid \phi \vee \phi \mid \Diamond\phi \mid \langle\neq\rangle\phi \mid \phi * \phi \mid \phi -\!\!* \phi,$$

where $p \in$ PROP. An MSL *model* is a tuple $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V}\rangle$ such that $\mathfrak{R} \subseteq \mathbb{N}\times\mathbb{N}$ is finite and functional, and $\mathfrak{V} :$ PROP $\to \mathcal{P}(\mathbb{N})$. Since separation logics are interpreted on structures representing heaps, our formulas are interpreted on models where the accessibility relation is finite and functional. The models

$\mathfrak{M}_1 = \langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle$ and $\mathfrak{M}_2 = \langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle$ are *disjoint* if $\mathfrak{R}_1 \cap \mathfrak{R}_2 = \emptyset$; when this holds, $\mathfrak{M}_1 \uplus \mathfrak{M}_2$ denotes the model corresponding to the disjoint union of $\mathfrak{M}_1$ and $\mathfrak{M}_2$, and $\mathfrak{M}_1 \subseteq \mathfrak{M}_2$ means that $\mathfrak{M}_1$ and $\mathfrak{M}_2$ have the same valuation and $\mathfrak{R}_1 \subseteq \mathfrak{R}_2$. Given $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and $\mathfrak{l} \in \mathbb{N}$, the satisfaction relation $\models$ is defined below (clauses for Boolean connectives are omitted):

$$
\begin{array}{lcl}
\mathfrak{M}, \mathfrak{l} \models p & \stackrel{\text{def}}{\Leftrightarrow} & \mathfrak{l} \in \mathfrak{V}(p) \\
\mathfrak{M}, \mathfrak{l} \models \mathtt{emp} & \stackrel{\text{def}}{\Leftrightarrow} & \mathfrak{R} = \emptyset \\
\mathfrak{M}, \mathfrak{l} \models \Diamond \phi & \stackrel{\text{def}}{\Leftrightarrow} & \mathfrak{M}, \mathfrak{l}' \models \phi, \text{ for some } \mathfrak{l}' \in \mathbb{N} \text{ such that } (\mathfrak{l}, \mathfrak{l}') \in \mathfrak{R} \\
\mathfrak{M}, \mathfrak{l} \models \langle \neq \rangle \phi & \stackrel{\text{def}}{\Leftrightarrow} & \mathfrak{M}, \mathfrak{l}' \models \phi, \text{ for some } \mathfrak{l}' \in \mathbb{N} \text{ such that } \mathfrak{l}' \neq \mathfrak{l} \\
\mathfrak{M}, \mathfrak{l} \models \phi_1 * \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l} \models \phi_1 \text{ and } \langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l} \models \phi_2, \\
 & & \text{for some partition } \{\mathfrak{R}_1, \mathfrak{R}_2\} \text{ of } \mathfrak{R} \\
\mathfrak{M}, \mathfrak{l} \models \phi_1 \mathrel{-\!\!*} \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \text{for all } \mathfrak{M}' = \langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle \text{ such that } \mathfrak{R} \cup \mathfrak{R}' \text{ is finite and} \\
 & & \text{functional, and } \mathfrak{R} \cap \mathfrak{R}' = \emptyset, \\
 & & \text{we have } \mathfrak{M}', \mathfrak{l} \models \phi_1 \text{ implies } \langle \mathbb{N}, \mathfrak{R} \cup \mathfrak{R}', \mathfrak{V} \rangle, \mathfrak{l} \models \phi_2.
\end{array}
$$

The semantics for the modal operators and the separating connectives is the standard one, see e.g. [8,34]. Other standard connectives or formulae are used:

- $[\neq]\phi \stackrel{\text{def}}{=} \neg\langle\neq\rangle\neg\phi$ and $\Box\phi \stackrel{\text{def}}{=} \neg\Diamond\neg\phi$,
- $\langle \mathrm{U} \rangle \phi \stackrel{\text{def}}{=} \phi \vee \langle\neq\rangle\phi$ and $[\mathrm{U}]\phi \stackrel{\text{def}}{=} \neg\langle \mathrm{U} \rangle \neg\phi$,
- $\langle\, ! \,\rangle \phi \stackrel{\text{def}}{=} \langle \mathrm{U} \rangle (\phi \wedge [\neq]\neg\phi)$ (unicity of the satisfaction of ϕ),
- the atomic formula $\mathtt{size} = 1$ is a shortcut for $\neg\mathtt{emp} \wedge \neg(\neg\mathtt{emp} * \neg\mathtt{emp})$.

The *satisfiability problem* for the logic MSL, takes as input a formula ϕ and asks whether there exist an MSL model $\mathfrak{M}$ and a location $\mathfrak{l}$ such that $\mathfrak{M}, \mathfrak{l} \models \phi$.

Not only our study includes MSL but above all, we also deal with fragments. For instance, the fragment with Boolean connectives and $\Diamond$ is the *basic modal logic* ML. Otherwise, as a convention, we always consider the Boolean part and the emptiness constant $\mathtt{emp}$, and we put between parentheses the rest of (separating or modal) connectives we are considering. The main logics we consider are MSL$(*, \Diamond)$, MSL$(*, \langle\neq\rangle)$ and MSL$(*, \Diamond, \langle\neq\rangle)$.

2.2 Nominals, program variables and separation logic in a nutshell

In all the fragments of MSL containing the inequality modality [19], it is known that nominals from hybrid logics [7] can be used since stating that p holds true in a unique location can be expressed by $\langle\, ! \,\rangle p$. So, we can freely use nominals. Syntactically, nominals are taken from PVAR $= \{\mathtt{x}, \mathtt{y}, \ldots\}$, that is actually also used as the set of *program variables* in separation logic (see below). Indeed, nominals and program variables are both interpreted by locations, as noticed in [24]. So, checking the satisfiability status of a formula ϕ containing $\mathtt{x}_1, \ldots, \mathtt{x}_n$ actually amounts to checking the satisfiability status of $(\bigwedge_{1 \leq i \leq n} \langle\, ! \,\rangle \mathtt{x}_i) \wedge \phi$. A formula ϕ is said to be *global* iff its satisfaction does not depend on the location and we simply write $\mathfrak{M} \models \phi$ (instead of $\mathfrak{M}, \mathfrak{l} \models \phi$). Below, we show why these formulae are important to compare MSL with separation logics.

Indeed, MSL behaves as a standard modal logic since the satisfaction relation has three arguments (a model, a location and a formula) but it can be also presented as a separation logic so that the satisfaction relation takes only two arguments, a model and a global formula. Let us briefly explain why separation logic can be viewed as a fragment of MSL. A *memory state* is a pair $(\mathfrak{s}, \mathfrak{h})$ such that $\mathfrak{s} : \text{PVAR} \to \mathbb{N}$ (the *store*) and $\mathfrak{h} : \mathbb{N} \to_{\text{fin}} \mathbb{N}$ is a partial function with finite domain (the *heap*). Models of the separation logic $\text{SL}(*, -\!\!*)$ are memory states. When the respective domains of the heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are disjoint, we write $\mathfrak{h}_1 \uplus \mathfrak{h}_2$ to denote the heap corresponding to the disjoint union of $\mathfrak{h}_1$ and $\mathfrak{h}_2$. Formulae of $\text{SL}(*, -\!\!*)$ are built from

$$\phi ::= \mathtt{x} = \mathtt{y} \mid \mathtt{x} \hookrightarrow \mathtt{y} \mid \mathtt{emp} \mid \neg\phi \mid \phi \wedge \phi \mid \phi * \phi \mid \phi -\!\!* \phi,$$

where $\mathtt{x}, \mathtt{y} \in \text{PVAR}$. The satisfaction relation $\models$ is defined as follows:

$$\begin{array}{lcl}
(\mathfrak{s}, \mathfrak{h}) \models \mathtt{x} = \mathtt{y} & \stackrel{\text{def}}{\Leftrightarrow} & \mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y}) \\
(\mathfrak{s}, \mathfrak{h}) \models \mathtt{emp} & \stackrel{\text{def}}{\Leftrightarrow} & \text{dom}(\mathfrak{h}) = \emptyset \\
(\mathfrak{s}, \mathfrak{h}) \models \mathtt{x} \hookrightarrow \mathtt{y} & \stackrel{\text{def}}{\Leftrightarrow} & \mathfrak{s}(\mathtt{x}) \in \text{dom}(\mathfrak{h}) \text{ and } \mathfrak{h}(\mathfrak{s}(\mathtt{x})) = \mathfrak{s}(\mathtt{y}) \\
(\mathfrak{s}, \mathfrak{h}) \models \phi_1 * \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \text{there are } \mathfrak{h}_1 \text{ and } \mathfrak{h}_2 \text{ such that } \mathfrak{h}_1 \uplus \mathfrak{h}_2 = \mathfrak{h}, \\
 & & (\mathfrak{s}, \mathfrak{h}_1) \models \phi_1 \text{ and } (\mathfrak{s}, \mathfrak{h}_2) \models \phi_2 \\
(\mathfrak{s}, \mathfrak{h}) \models \phi_1 -\!\!* \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \text{for all } \mathfrak{h}_1, \text{ if } (\text{dom}(\mathfrak{h}_1) \cap \text{dom}(\mathfrak{h}) = \emptyset \text{ and} \\
 & & (\mathfrak{s}, \mathfrak{h}_1) \models \phi_1), \text{ then } (\mathfrak{s}, \mathfrak{h} \uplus \mathfrak{h}_1) \models \phi_2.
\end{array}$$

Any memory state $(\mathfrak{s}, \mathfrak{h})$ can be viewed as the MSL model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ such that $\mathfrak{R} = \{(\mathfrak{l}, \mathfrak{h}(\mathfrak{l})) \mid \mathfrak{l} \in \text{dom}(\mathfrak{h})\}$ and the restriction of $\mathfrak{V}$ to PVAR is equal to $\mathfrak{s}$. Actually, any formula ϕ of $\text{SL}(*, -\!\!*)$ is satisfiable iff $t(\phi)$ is satisfiable in MSL where t is homomorphic for Boolean and separating connectives and,

$$t(\mathtt{x} = \mathtt{y}) \stackrel{\text{def}}{=} \langle \text{U} \rangle (\mathtt{x} \wedge \mathtt{y}) \qquad t(\mathtt{emp}) \stackrel{\text{def}}{=} \mathtt{emp} \qquad t(\mathtt{x} \hookrightarrow \mathtt{y}) \stackrel{\text{def}}{=} \langle \text{U} \rangle (\mathtt{x} \wedge \Diamond \mathtt{y}).$$

It is worth noting that each formula $t(\phi)$ is a global formula of MSL.

2.3 Alternative semantics

A *general model* $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ is such that $\mathfrak{W}$ is an arbitrary countable set, $\mathfrak{R} \subseteq \mathfrak{W} \times \mathfrak{W}$ and $\mathfrak{V} : \text{PROP} \to \mathcal{P}(\mathfrak{W})$. This corresponds to standard (countable) Kripke structures with no frame condition. A *finite and functional model* $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ is such that $\mathfrak{W}$ is a finite set, $\mathfrak{R} \subseteq \mathfrak{W} \times \mathfrak{W}$ is functional and $\mathfrak{V}$ is a valuation. Without loss of generality, we assume $\mathfrak{W} \subseteq \mathbb{N}$. Each syntactic fragment $\mathfrak{L}$ of MSL gives rise to the logic $\mathfrak{L}^f$ (resp. $\mathfrak{L}^g$) where the models for $\mathfrak{L}^f$ are finite and functional models (resp. are general models). When $\mathfrak{L}$ includes $-\!\!*$, the definition of $\models$ for $\mathfrak{L}^g$ is updated as follows:

$$\begin{array}{lcl}
\mathfrak{M}, \mathfrak{l} \models \phi_1 -\!\!* \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \text{for all } \mathfrak{M}' = \langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle \text{ such that } \mathfrak{R} \cap \mathfrak{R}' = \emptyset \\
 & & \mathfrak{M}', \mathfrak{l} \models \phi_1 \text{ implies } \langle \mathfrak{W}, \mathfrak{R} \cup \mathfrak{R}', \mathfrak{V} \rangle, \mathfrak{l} \models \phi_2.
\end{array}$$

Note that the formula $(\top -\!\!* \neg((\neg\mathtt{emp}) -\!\!* \bot))$ is valid for MSL but not for MSL^f. The *model-checking problem for* MSL^f is defined in the usual way. As MSL can be viewed as a fragment of second-order logic (the second-order feature is needed to internalise the semantics of separating connectives), the

model-checking problem for MSL^f is in PSPACE. More surprisingly, we show that the restriction to either $\mathrm{MSL}^f(*, \langle\neq\rangle)$ or $\mathrm{MSL}^f(*, \Diamond)$ is in P, whereas the restriction to $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ is already untractable.

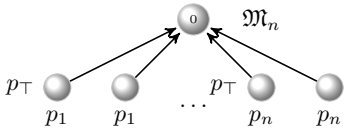
Lemma 2.1 *The model-checking problem for* $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ *is* PSPACE*-hard.*

Proof Let $\mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$ be a QBF formula with $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_n\} \subseteq \{\exists, \forall\}$ and ϕ is a propositional formula built over $\{p_1, \ldots, p_n\}$ and the Boolean connectives $\wedge$, $\vee$ and $\neg$ (only in front of atomic propositions). Satisfiability problem for QBF formulae is known to be PSPACE-complete [38].

In the reduction of $\varphi = \mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$, we introduce a finite and functional model $\mathfrak{M}_n = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V}\rangle$ with $\mathfrak{W} = [0, 2n]$ such that $\mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$ is satisfiable iff $\mathfrak{M}_n, 0 \models t(\varphi)$, where $t(\cdot)$ is recursively defined below. The truth of the propositional variable p_i in QBF subformulae is encoded by the satisfaction of the formula $\langle\neq\rangle(p_i \wedge p_\top \wedge \Diamond\top)$ from $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$.

First, let us complete the definition of $\mathfrak{M}_n$ over the propositional variables $\{p_\top, p_1, \ldots, p_n\}$.

$$\begin{array}{ll} \mathfrak{V}(p_i) & \stackrel{\text{def}}{=} \{i, n+i\}, \text{ for all } i = 1, \ldots, n \\ \mathfrak{V}(p_\top) & \stackrel{\text{def}}{=} [1, n] \\ \mathfrak{R} & \stackrel{\text{def}}{=} \{(i, 0) \mid i \in [1, 2n]\}. \end{array}$$



Let us define the map t as follows (homomorphic for Boolean connectives):

$$\begin{array}{lcl} t(p_i) & \stackrel{\text{def}}{=} & \langle\neq\rangle(p_i \wedge p_\top \wedge \Diamond\top) \\ t(\exists\ p_i\ \psi) & \stackrel{\text{def}}{=} & (\mathtt{size} = 1 \wedge \langle\neq\rangle(p_i \wedge \Diamond\top)) * t(\psi) \\ t(\forall\ p_i\ \psi) & \stackrel{\text{def}}{=} & \neg((\mathtt{size} = 1 \wedge \langle\neq\rangle(p_i \wedge \Diamond\top)) * \neg t(\psi)). \end{array}$$

For every $j \in [1, n+1]$, we write ϕ_j to denote the formula $\mathcal{Q}_j\ p_j\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$. By definition, we have $\phi_1 = \mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$ and by convention $\phi_{n+1} = \phi$.

Given a model $\mathfrak{M} \subseteq \mathfrak{M}_n$ and a propositional valuation v, we write $\mathfrak{M} \approx_j v$ to denote the fact that:

- For all $i \in [1, j-1]$, exactly one location in $\{i, n+i\}$ has an outgoing edge.
- For all $i \in [j, n]$, all the locations in $\{i, n+i\}$ have an outgoing edge.
- For all $i \in [1, j-1]$, i has an outgoing edge iff $v(p_i) = \top$.

By induction on j, one can show that for all $j \in [1, n+1]$, if $\mathfrak{M} \approx_j v$, then $\mathfrak{M}, 0 \models t(\phi_j)$ iff $v \models \phi_j$. Details are omitted. So, as $\mathfrak{M}_n \approx_1 v$ for any v, we have $v \models \varphi$ iff $\mathfrak{M}_n, 0 \models t(\phi_1)$. As φ is a closed formula and $\phi_1 = \varphi$, φ is satisfiable iff $\mathfrak{M}_n, 0 \models t(\varphi)$. □

MSL can be seen as a logic with the ability to add or remove edges from the accessibility relation, closely related to relation-changing modal logics [1]. Below, we discuss the connections between MSL and the *global sabotage logic* $\mathrm{MSL}^g(\Diamond, \langle\mathsf{gsb}\rangle)$. Formulae of $\mathrm{MSL}^g(\Diamond, \langle\mathsf{gsb}\rangle)$ extends those of ML by adding the operator $\langle\mathsf{gsb}\rangle$ interpreted over general models $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V}\rangle$ as:

$$\mathfrak{M}, \mathfrak{l} \models \langle\mathsf{gsb}\rangle\phi \stackrel{\text{def}}{\Leftrightarrow} \text{for some } (\mathfrak{l}', \mathfrak{l}'') \in \mathfrak{R},\ \mathfrak{M}^-_{\mathfrak{l}', \mathfrak{l}''}, \mathfrak{l} \models \phi,$$

where $\mathfrak{M}^{-}_{\mathfrak{l}',\mathfrak{l}''} = \langle \mathfrak{W}, \mathfrak{R} \setminus \{(\mathfrak{l}', \mathfrak{l}'')\}, \mathfrak{V} \rangle$. MSL$^g(\Diamond, \langle \mathsf{gsb} \rangle)$ can be encoded into MSL$^g(*, \Diamond)$ by the translation t that is homomorphic for Boolean connectives and for $\Diamond$ and, $t(\langle \mathsf{gsb} \rangle \phi) \stackrel{\text{def}}{=} (\mathtt{size} = 1) * t(\phi)$. We have ϕ is satisfiable iff $t(\phi)$ is satisfiable for MSL$^g(*, \Diamond)$. Similarly, MSL$(\Diamond, \langle \mathsf{gsb} \rangle)$ is the variant of MSL$^g(\Diamond, \langle \mathsf{gsb} \rangle)$ with MSL models.

3 Decision problems in Tower

Below, we establish that the satisfiability problem for MSL$(*, \Diamond, \langle \neq \rangle)$ is in Tower [35], the class of problems of time complexity bounded by a tower of exponentials, whose height is an elementary function of the input. To do so, we design a reduction to the satisfiability problem for MSL$^f(*, \Diamond, \langle \neq \rangle)$ and then we show that the satisfiability problem for MSL$^f(*, \Diamond, \langle \neq \rangle)$ is in Tower by translation into the weak MSO theory of one unary function. Notice that the difference between MSL$(*, \Diamond, \langle \neq \rangle)$ and MSL$^f(*, \Diamond, \langle \neq \rangle)$ is that models in MSL$(*, \Diamond, \langle \neq \rangle)$ have finite relations over an infinite set of locations, while the set of locations in MSL$^f(*, \Diamond, \langle \neq \rangle)$ models is finite. This proof is analogous to the decidability proof for 1SL$(*)$ in [11] but our main technical task is to solve the satisfiability problem for MSL$(*, \Diamond, \langle \neq \rangle)$ by using only propositional variables that hold true on a finite amount of locations. First, we show that locations satisfying the same propositional variables and with no successor satisfy the same formulae.

Lemma 3.1 *Let $p_1, \ldots, p_n$ be propositional variables, $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ be a model and $\mathfrak{l} \neq \mathfrak{l}'$ be locations such that $\mathfrak{R}(\mathfrak{l}) = \mathfrak{R}(\mathfrak{l}') = \emptyset$ and, $\mathfrak{l}$ and $\mathfrak{l}'$ agree on $p_1, \ldots, p_n$. For all ϕ in* MSL$(*, \Diamond, \langle \neq \rangle)$ *built over $p_1, \ldots, p_n$, $\mathfrak{M}, \mathfrak{l} \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$.*

Let ϕ in MSL$(*, \Diamond, \langle \neq \rangle)$ be built over $p_1, \ldots, p_n$. Let us define $T(\phi)$ as

$$T(\phi) \stackrel{\text{def}}{=} \phi \wedge \bigvee_{X \subseteq \{p_1, \ldots, p_n\}} \langle \mathrm{U} \rangle (\Box \bot \wedge \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \wedge \langle \neq \rangle (\Box \bot \wedge \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p)).$$

When $\langle \neq \rangle$ is not present, the second conjunct can be removed (see Lemma 4.3, where we take $T(\phi) = \phi$). Such a conjunct states that there are two distinct locations with no successor that agree on propositional variables from X and it is needed since $\langle ! \rangle p \wedge [\mathrm{U}] p$ is satisfiable for MSL$^f(*, \langle \neq \rangle)$ but not for the logic MSL$(*, \langle \neq \rangle)$.

Lemma 3.2 *ϕ is satisfiable in* MSL$(*, \Diamond, \langle \neq \rangle)$ *iff $T(\phi)$ is satisfiable in* MSL$^f(*, \Diamond, \langle \neq \rangle)$.

The complexity class Tower has been introduced in [35] and sits between the class of elementary problems and the class of primitive recursive problems.

Theorem 3.3 *The satisfiability problem for* MSL$(*, \Diamond, \langle \neq \rangle)$ *is in* Tower.

By Lemma 3.2, there is a reduction from the satisfiability problem for MSL$(*, \Diamond, \langle \neq \rangle)$ into the satisfiability problem for MSL$^f(*, \Diamond, \langle \neq \rangle)$ that works

in exponential time. There is also a (logspace) reduction from the satisfiability problem for $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ into the satisfiability problem for the weak MSO theory of one unary function whose structures are $\langle D, \mathfrak{f}, =\rangle$ where D is a countable domain, $\mathfrak{f}$ is a unary function ('weakness' refers to the fact that the monadic predicates are interpreted by *finite* sets). This theory is decidable, see e.g. [9, Corollary 7.2.11] and it can be shown in TOWER as it can be reduced to the satisfiability to the MSO theory of the infinite binary tree. In the proof of Theorem 3.3, the reduction from $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ simply internalises its semantics by using the (weak) second-order feature of the target logic.

In order to conclude this section, we consider the standard converse modality $\Diamond^{-1}$ (not originally in MSL), when it interacts with separating connectives. More precisely, $\mathfrak{M}, \mathfrak{l} \models \Diamond^{-1}\phi \stackrel{\text{def}}{\Leftrightarrow} \mathfrak{M}, \mathfrak{l}' \models \phi$, for some $\mathfrak{l}' \in \mathbb{N}$ such that $(\mathfrak{l}', \mathfrak{l}) \in \mathfrak{R}$. Although replacing $\Diamond$ by $\Diamond^{-1}$ does not sound as leading to a major variant, we will show that $\Diamond^{-1}$ already brings new difficulties.

Theorem 3.4 *The satisfiability problem for* $\mathrm{MSL}(*, \Diamond^{-1})$ *is* PSPACE*-hard as well as the model-checking problem for* $\mathrm{MSL}^f(*, \Diamond^{-1})$.

Note that $\mathrm{MSL}(*, \Diamond^{-1})$ contains $\mathrm{MSL}(\Diamond^{-1})$ that can be viewed as a slight variant of the modal logic K on finite trees, known to admit a PSPACE-complete satisfiability problem. So, PSPACE-hardness of the satisfiability problem for $\mathrm{MSL}(*, \Diamond^{-1})$ is quite expected.

By using the proof technique from the proof of Theorem 3.3, we can establish the result below where $\Diamond^{-1}$ is part of the modal operators.

Theorem 3.5 *The satisfiability problem for* $\mathrm{MSL}(*, \Diamond, \Diamond^{-1}, \langle\neq\rangle)$ *is in* TOWER.

As a conclusion, there is a huge gap for $\mathrm{MSL}(*, \Diamond^{-1})$ between the PSPACE-hardness for the satisfiability problem and the TOWER upper bound.

4 NP-complete fragments of MSL

In this section, we show that the satisfiability problems for $\mathrm{MSL}(*, \Diamond)$ and for $\mathrm{MSL}(*, \langle\neq\rangle)$ are NP-complete. In order to establish the NP upper bound, we reduce the problems to their variants with finite and functional models, we show a linear-size model property and finally, we prove that the model-checking problems are in P, dealing in each case with particular technical difficulties.

4.1 The minimal modal separation logic $\mathrm{MSL}(*, \Diamond)$

To show that $\mathrm{MSL}(*, \Diamond)$ has a linear-size model property (i.e., the cardinal of the relation can be bounded), we introduce an equivalence relation $\overset{s,n}{\sim}$ ($s \geq 0$ is a parameter about the number of edges and $n \geq 1$ is a parameter about the propositional variables) such that $\overset{s,n}{\sim}$-equivalent models satisfy the same formulae with less than s syntactic resources (to be defined) and built over $\{p_1, \ldots, p_n\}$. First, we need to explain how to decompose models with respect to the parameters s and n and, the relation $\overset{s,n}{\sim}$ is defined by using such a decomposition. As $\mathfrak{R}$ is functional, what matters is the structure of $\mathfrak{R}$ reduced

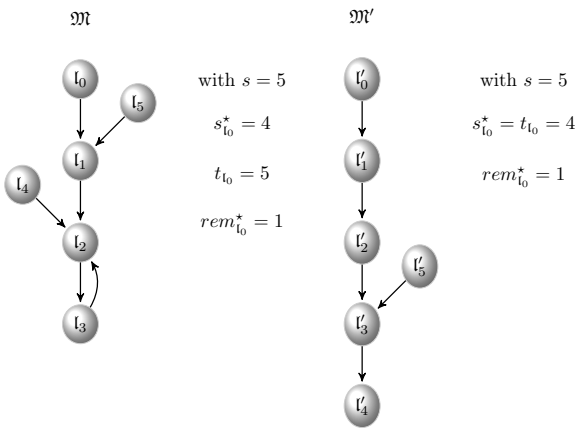


Figure 1. Decomposition.

to at most the s first steps from a given location as well as the total number of edges, counting up to s. Below, we show that this abstraction is correct with respect to the expressive power of MSL$(*, \Diamond)$, see e.g. Lemma 4.2. Let $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ be a model, $\mathfrak{l} \in \mathbb{N}$ and $s \geq 0$, we define $\mathfrak{W}_{\mathfrak{l},s}$ and $\mathfrak{R}_{\mathfrak{l},s}$ as follows.

- $\mathfrak{W}_{\mathfrak{l},s} \stackrel{\text{def}}{=} \{(i, \mathfrak{l}_i) \mid i \in [0,s],\ \exists\, \mathfrak{l}_0, \ldots, \mathfrak{l}_i,\ \mathfrak{l} = \mathfrak{l}_0 \mathfrak{R} \mathfrak{l}_1 \cdots \mathfrak{R} \mathfrak{l}_{i-1} \mathfrak{R} \mathfrak{l}_i\}$. We also write $t_{\mathfrak{l}} = \max\{i \mid (i, \mathfrak{l}_i) \in \mathfrak{W}_{\mathfrak{l},s}\}$ (so $t_{\mathfrak{l}} \leq s$).
- $\mathfrak{R}_{\mathfrak{l},s} \stackrel{\text{def}}{=} \{(\mathfrak{l}_i, \mathfrak{l}_{i+1}) \mid i \in [0, t_{\mathfrak{l}} - 1] \text{ and } (i, \mathfrak{l}_i), (i+1, \mathfrak{l}_{i+1}) \in \mathfrak{W}_{\mathfrak{l},s}\}$. We also write $s^\star_{\mathfrak{l}} \stackrel{\text{def}}{=} \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s})$ and $rem^\star_{\mathfrak{l}} = \min(s - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}), \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}))$. So, $s^\star_{\mathfrak{l}} \leq t_{\mathfrak{l}} \leq s$ and $s^\star_{\mathfrak{l}} + rem^\star_{\mathfrak{l}} \leq s$.

Let $\mathfrak{M}$, $\mathfrak{M}'$ be models, $\mathfrak{l}, \mathfrak{l}' \in \mathbb{N}$ and $s \geq 0$, $n \geq 1$ such that $\mathfrak{W}_{\mathfrak{l},s}$ and $\mathfrak{R}_{\mathfrak{l},s}$ are defined as above and $\mathfrak{W}'_{\mathfrak{l}',s}$ and $\mathfrak{R}'_{\mathfrak{l}',s}$ are related to $\mathfrak{M}'$, $\mathfrak{l}'$ and s. Let us define the relation $\overset{s,n}{\sim}$: $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}' \stackrel{\text{def}}{\Leftrightarrow}$ the conditions below are satisfied:

- We have $t_{\mathfrak{l}} = t_{\mathfrak{l}'} (\stackrel{\text{def}}{=} t)$. Say, $\mathfrak{W}_{\mathfrak{l},s} = \{(0, \mathfrak{l}_0), \ldots, (t, \mathfrak{l}_t)\}$ and $\mathfrak{W}'_{\mathfrak{l}',s} = \{(0, \mathfrak{l}'_0), \ldots, (t, \mathfrak{l}'_t)\}$.
- For all $i \in [0,t]$, $\mathfrak{l}_i$ in $\mathfrak{M}$ and $\mathfrak{l}'_i$ in $\mathfrak{M}'$ agree on $\{p_1, \ldots, p_n\} \subset \text{PROP}$.
- For all $i, j \in [0, t-1]$, we have $\mathfrak{l}_i = \mathfrak{l}_j$ iff $\mathfrak{l}'_i = \mathfrak{l}'_j$. Hence, $s^\star_{\mathfrak{l}} = s^\star_{\mathfrak{l}'} (\stackrel{\text{def}}{=} s^\star)$.
- We have $rem^\star_{\mathfrak{l}} = rem^\star_{\mathfrak{l}'} (\stackrel{\text{def}}{=} rem^\star)$.

The binary relation $\overset{s,n}{\sim}$ is an equivalence relation. In Figure 1, $\mathfrak{M}, \mathfrak{l}_0 \overset{4,n}{\sim} \mathfrak{M}', \mathfrak{l}'_0$ (assuming that $\mathfrak{l}_i$ and $\mathfrak{l}'_i$ agree on $\{p_1, \ldots, p_n\}$ for every $i \in [0,3]$, and, $\mathfrak{l}_2/\mathfrak{l}'_2$ and $\mathfrak{l}'_4$ agree too). By contrast, $\mathfrak{M}, \mathfrak{l}_0 \overset{5,n}{\sim} \mathfrak{M}', \mathfrak{l}'_0$ does not hold. Lemma 4.1 below is essential to justify that $\overset{s,n}{\sim}$ behaves properly with disjoint unions of models. Its proof is tedious as numerous cases are needed.

Lemma 4.1 *Let $s, s_1, s_2 \geq 1$ with $s = s_1 + s_2$, $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$ and $\mathfrak{M}_1, \mathfrak{M}_2$ be models such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$. There are models $\mathfrak{M}'_1$ and $\mathfrak{M}'_2$ such that $\mathfrak{M}' = \mathfrak{M}'_1 \uplus \mathfrak{M}'_2$, $\mathfrak{M}_1, \mathfrak{l} \overset{s_1,n}{\sim} \mathfrak{M}'_1, \mathfrak{l}'$ and $\mathfrak{M}_2, \mathfrak{l} \overset{s_2,n}{\sim} \mathfrak{M}'_2, \mathfrak{l}'$.*

Given a formula ϕ in MSL$(*, \Diamond)$, let us define its *esize* (written esize(ϕ)):

- esize$(p) \stackrel{\text{def}}{=}$ esize$(\mathtt{emp}) \stackrel{\text{def}}{=} 1$, esize$(\neg\phi) \stackrel{\text{def}}{=}$ esize(ϕ), esize$(\Diamond\phi) \stackrel{\text{def}}{=} 1 +$ esize(ϕ),
- esize$(\phi \wedge \psi) \stackrel{\text{def}}{=} \max($esize$(\phi),$ esize$(\psi))$, esize$(\phi * \psi) \stackrel{\text{def}}{=}$ esize$(\phi) +$ esize(ψ).

Note that esize(ϕ) is greater than the modal degree of ϕ, and approximatively, esize(ϕ) provides an upper bound on the number of edges that need to be considered in a model for ϕ (so it will play the role of the value s). For technical reasons, we have assumed that esize$(p) = 1$, so that esize$(\phi) \geq 1$ for any ϕ.

Lemma 4.2 *Let $s, n \geq 1$. For all formulae ϕ in* MSL$(*, \Diamond)$ *with* esize$(\phi) \leq s$ *and built over $p_1, \ldots, p_n$, we have $\mathfrak{M}, \mathfrak{l} \stackrel{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$ implies $\mathfrak{M}, \mathfrak{l} \models \phi$ iff $\mathfrak{M}', \mathfrak{l}' \models \phi$.*

The following quantitative result is crucial to get the NP upper bound.

Lemma 4.3 *Let ϕ be a formula in* MSL$(*, \Diamond)$. *ϕ is satisfiable iff ϕ is satisfiable in a finite and functional model with* $\mathsf{card}(\mathfrak{R}) \leq$ esize(ϕ).

It remains to show that the model-checking problem for MSL$^f(*, \Diamond)$ is in P. The main difficulty rests on the fact that evaluating an $*$-formula may require to consider an exponential number of pairs of disjoint submodels. Fortunately, only a polynomial amount of disjoint unions are shown relevant. At the beginning of this section, we defined a decomposition of any MSL model based on the parameter $s \geq 0$. Such a decomposition was useful to show Lemma 4.3. A similar decomposition can be done with finite and functional models. More precisely, let $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ be a finite and functional model and $\mathfrak{l} \in \mathfrak{W}$. One can easily define the set $\mathfrak{W}_{\mathfrak{l},s}$, the relation $\mathfrak{R}_{\mathfrak{l},s}$ and the values $t_{\mathfrak{l}}$, $s^\star_{\mathfrak{l}}$ and $rem^\star_{\mathfrak{l}}$. Consequently, an equivalence relation $\stackrel{s,n}{\sim}$ can be also defined on finite and functional pointed models leading to a natural variant of Lemma 4.2 involving finite and functional models instead of MSL models.

In order to check whether $\mathfrak{M}, \mathfrak{l} \models \phi$ holds, we start by building a submodel $\mathfrak{M}' = \langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle \subseteq \mathfrak{M}$ with $\mathsf{card}(\mathfrak{R}') \leq$ esize(ϕ) and check whether $\mathfrak{M}', \mathfrak{l} \models \phi$ holds. The submodel $\mathfrak{M}'$ can be built in polynomial time in the size of $\mathfrak{M}$ and in s. In forthcoming Algorithm 1, instead of working with models, we operate with slightly more abstract structures. An *abstract frame up to s* is a pair $\mathcal{F} = ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ where $r \geq 0$, $(\mathfrak{l}_0, \ldots, \mathfrak{l}_t) \in \mathbb{N}^+$ (standing for locations linked by edges) and the conditions below hold:

(truncation) $t^* + r \leq s$ and $t \leq s$ with $t^* = \mathsf{card}(\{(\mathfrak{l}_i, \mathfrak{l}_{i+1}) \mid i \in [0, t-1]\})$.
(maximality) $t < s$ implies there is no $i < t$ such that $\mathfrak{l}_i = \mathfrak{l}_t$.
(functionality) for all $i < j < t$, we have $\mathfrak{l}_i = \mathfrak{l}_j$ implies $t = s$ and $\mathfrak{l}_{i+1} = \mathfrak{l}_{j+1}$.

Given a finite and functional model $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$, $\mathfrak{l} \in \mathfrak{W}$, and $s \geq 0$, we write abst$(\mathfrak{M}, \mathfrak{l}, s)$ to denote the abstraction $((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ with $\{(0, \mathfrak{l}_0), \ldots, (t, \mathfrak{l}_t)\} = \mathfrak{W}_{\mathfrak{l},s}$ and $r = rem^\star_{\mathfrak{l}}$. An abstract frame is not explicitly equipped with a propositional valuation but in forthcoming Algorithm 1, we manipulate such structures as the associated propositional valuation will be systematically the one induced by the valuation of the input model. Let shrink$(\mathfrak{M}, \mathfrak{l}, s)$ be the finite and functional model $\mathfrak{M}' = \langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle$ such that $\mathfrak{R}' \stackrel{\text{def}}{=} \{(\mathfrak{l}_i, \mathfrak{l}_{i+1}) \mid i \in [0, t-1]\} \cup \{(\mathfrak{n}_1, \mathfrak{n}'_1), \ldots, (\mathfrak{n}_r, \mathfrak{n}'_r)\}$, where

$\{(\mathfrak{n}_1, \mathfrak{n}'_1), \ldots, (\mathfrak{n}_r, \mathfrak{n}'_r)\}$ is a set of r egdes in $\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}$ and the locations $\mathfrak{n}_1, \ldots, \mathfrak{n}_r$ are minimal. Lemma 4.4 below justifies the correctness of the abstraction.

Lemma 4.4 *Let $s \geq 0$, $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ be finite and functional and $\mathfrak{l} \in \mathfrak{W}$ with $\mathfrak{M}' = \mathrm{shrink}(\mathfrak{M}, \mathfrak{l}, s)$. Then $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}$ and $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathrm{abst}(\mathfrak{M}', \mathfrak{l}, s)$.*

Let us define a notion of disjoint union between abstract frames to mimic the disjoint union of models. Let $s = s_1 + s_2$, $s, s_1, s_2 \geq 1$, $\mathcal{F} = ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ be an abstract frame up to s, $\mathcal{F}_i = ((\mathfrak{l}^i_0, \ldots, \mathfrak{l}^i_{t_i}), r^i)$ be an abstract frame up to s_i, with $i \in \{1, 2\}$. We write $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2 \overset{\text{def}}{\Leftrightarrow}$ (i)–(v) below hold ($i \in \{1, 2\}$):

(i) $\max(t_1, t_2) \leq t$, $t_1 \times t_2 = 0$ and, if $t > 0$ then $t_1 + t_2 > 0$.
(ii) $(\mathfrak{l}^i_0, \ldots, \mathfrak{l}^i_{t_i}) = (\mathfrak{l}_0, \ldots, \mathfrak{l}_{t_i})$.
(iii) $0 < t_i < \min(s_i, t)$ implies $r^{3-i} > 0$.
(iv) $0 < t_i$ implies $r^1 + r^2 \leq r + t^* - t^*_i$.
(v) $0 < t_i$ and $r^1 + r^2 < r + t^* - t^*_i$ imply $r^i = s_i - t^*_i$ or $r^{3-i} = s_{3-i}$.

Though (i)–(v) sound reasonable at first glance, the best way to understand what is really needed, is by proving Lemma 4.5 and Lemma 4.6. Similarly, given abstract frames $\mathcal{F}_1 = ((\mathfrak{l}^1_0, \ldots, \mathfrak{l}^1_{t_1}), r^1)$ up to s_1 and $\mathcal{F}_2 = ((\mathfrak{l}^2_0, \ldots, \mathfrak{l}^2_{t_2}), r^2)$ up to s_2 with $s_1 \leq s_2$, we write $\mathcal{F}_1 \subseteq \mathcal{F}_2$ whenever $(\mathfrak{l}^1_0, \ldots, \mathfrak{l}^1_{t_1})$ is a factor of $(\mathfrak{l}^2_0, \ldots, \mathfrak{l}^2_{t_2})$ and, $r^1 + t^*_1 \leq r^2 + t^*_2$.

Algorithm 1 below operates with abstract frames and its correctness is partly based on forthcoming Lemma 4.5 and Lemma 4.6. For instance, Lemma 4.5 can be understood as a correctness result: disjoint unions of models lead to the satisfaction of the conditions (i)-(v) at the level of abstract frames.

Lemma 4.5 *Let $s = s_1 + s_2$ with $s, s_1, s_2 \geq 1$. Let $\mathfrak{M}$, $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be finite and functional models such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$. For all $\mathfrak{l} \in \mathfrak{W}$, we have $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1) \uplus \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$.*

By contrast, Lemma 4.6 below can be understood as a completeness result: the satisfaction of (i)-(v) can always be mimicked at the level of models.

Lemma 4.6 *Let $s = s_1 + s_2$ with $s, s_1, s_2 \geq 1$. Let $\mathfrak{M}$ be finite and functional, $\mathcal{F}_i$ be an abstract frame up to s_i ($i \in \{1, 2\}$) such that $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathcal{F}_1 \uplus \mathcal{F}_2$. There are $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathcal{F}_i = \mathrm{abst}(\mathfrak{M}_i, \mathfrak{l}, s_i)$ ($i \in \{1, 2\}$).*

What is essential is the fact that the number of non-equivalent decompositions is polynomial and not exponential in the size of $\mathcal{F}$, which is a serious guarantee to obtain a model checking algorithm running in polynomial time.

Lemma 4.7 *Let $s = s_1 + s_2$ with $s, s_1, s_2 \geq 1$, $\mathcal{F} = ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ be an abstract frame up to s. We have $\mathsf{card}(\{(\mathcal{F}_1, \mathcal{F}_2) \mid \mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2,\ \mathcal{F}_i \text{ up to } s_i\}) \leq 2(s+1)(s_1+1)(s_2+1)$.*

Algorithm 1 below uses first principles of dynamic programming as well as the map $\mathrm{shrink}(\mathcal{F}, s)$ defined as follows (abstract version of the shrink construction on models): $\mathrm{shrink}(((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r), s) \overset{\text{def}}{=} ((\mathfrak{l}_0, \ldots, \mathfrak{l}_{t'}), r')$ with

- $t' = \min(s, t)$,

- $t'_* = \mathsf{card}(\{\mathfrak{l}_1, \ldots, \mathfrak{l}_{t'}\})$ and,
- $r' = \min(s - t'_*, r + (\mathsf{card}(\{\mathfrak{l}_1, \ldots, \mathfrak{l}_t\}) - t'_*))$.

One can show that $\mathrm{shrink}(((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r), s) \subseteq ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ and $\mathrm{shrink}(((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r), s)$ is an abstract frame up to s. Algorithm 1 only computes values for $T(\mathrm{shrink}(\mathcal{F}, \mathrm{esize}(\psi_k)), k)$ as it would be time-consuming (and useless) to compute all the values $T(\mathcal{F}, k)$. This is enforced by the values in the **for** loops and by line 2. The map $\mathrm{shrink}(\cdot, \cdot)$ is also further needed for conjunctions as the measure $\mathrm{esize}(\cdot)$ involves a maximum for conjunctions.

Algorithm 1 Model Checking $\mathrm{MSL}^f(*, \Diamond)$

In: A finite and functional model $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$, a location $\mathfrak{l} \in \mathfrak{W}$, an $\mathrm{MSL}(*, \Diamond)$ formula ϕ
Out: Return 1 iff $\mathfrak{M}, \mathfrak{l} \models \phi$.

1: **function** $\mathrm{MC}(\mathfrak{M}, \mathfrak{l}, \phi)$
2: $((\mathfrak{l}_0, \ldots, \mathfrak{l}_L), R) := \mathrm{abst}(\mathfrak{M}, \mathfrak{l}, \mathrm{esize}(\phi))$ ▷ $\mathsf{card}(\{\mathfrak{l}_1, \ldots, \mathfrak{l}_L\}) + R \leq \mathrm{esize}(\phi)$
3: $\psi_1, \ldots, \psi_M$ subformulae of ϕ in increasing size ▷ $\psi_M = \phi$
4: **for** $k \leftarrow 1$ **to** M **do**
5: **for** $j \leftarrow L$ **downto** 0 **do**
6: **for** $len \leftarrow 0$ **to** $\max\{len' \in [0, L-j] \mid \mathsf{card}(\{\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len'}\}) \leq \mathrm{esize}(\psi_k)\}$ **do**
7: **for** $r \leftarrow 0$ **to** $\max\{r' \in [0, R] \mid \mathsf{card}(\{\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}\}) + r' \leq \mathrm{esize}(\psi_k)\}$ **do**
8: $\mathcal{F} := ((\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}), r)$ ▷ $\mathcal{F} = \mathrm{shrink}(\mathcal{F}, \mathrm{esize}(\psi_k))$
9: **case** ψ_k **of**
10: $\mathtt{emp}$: $T(\mathcal{F}, k) := 1$ if ($len = 0$ and $r = 0$), otherwise 0.
11: p: $T(\mathcal{F}, k) := 1$ if $\mathfrak{l}_j \in \mathfrak{V}(p)$, otherwise 0.
12: $\neg\psi_{k'}$: $T(\mathcal{F}, k) := 1 - T(\mathcal{F}, k')$ ▷ $k' < k$
13: $\psi_{k_1} \wedge \psi_{k_2}$: ▷ $k_1, k_2 < k$
14: $T(\mathcal{F}, k) := \min(T(\mathrm{shrink}(\mathcal{F}, \mathrm{esize}(\psi_{k_1})), k_1), T(\mathrm{shrink}(\mathcal{F}, \mathrm{esize}(\psi_{k_2})), k_2))$
15: $\Diamond\psi_{k'}$: if ($len > 0$), $\mathcal{F}' := \mathrm{shrink}(((\mathfrak{l}_{j+1}, \ldots, \mathfrak{l}_{j+len}), r), \mathrm{esize}(\psi_{k'}))$ ▷ $k' < k$
16: $T(\mathcal{F}, k) := 1$ if ($len > 0$) and $T(\mathcal{F}', k') = 1$, otherwise 0.
17: $\psi_{k_1} * \psi_{k_2}$: ▷ $k_1, k_2 < k$
18: $s_1 := \mathrm{esize}(\psi_{k_1})$; $s_2 := \mathrm{esize}(\psi_{k_2})$ ▷ $\mathrm{esize}(\psi_k) = s_1 + s_2$
19: $T(\mathcal{F}, k) := \max\{\min(T(\mathcal{F}_1, k_1), T(\mathcal{F}_2, k_2)) \mid \mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2,\ \mathcal{F}_i \text{ up to } s_i\}$
20: **end case**
21: **return** $T(((\mathfrak{l}_0, \ldots, \mathfrak{l}_L), R), M)$

Due to the organisation of the **for** loops, each time the algorithm computes $T(\mathcal{F}, k)$, it requires values of the form $T(\mathcal{F}', k')$, always with $\mathcal{F}' \subseteq \mathcal{F}$ and $k' < k$, so the algorithm is properly defined. The algorithm runs in polynomial time thanks to Lemma 4.7. The following lemma establishes that the algorithm is correct and explains what is the intention behind computing the values $T(\mathcal{F}, k)$.

Lemma 4.8 *For all $k \in [1, M]$, for all abstract frames $\mathcal{F} = ((\mathfrak{l}, \ldots), R')$ up to* $\mathrm{esize}(\psi_k)$ *with $\mathcal{F} \subseteq ((\mathfrak{l}_0, \ldots, \mathfrak{l}_L), R)$, when the model-checking algorithm ends, $T(\mathcal{F}, k) = 1$ iff for all finite and functional submodels $\mathfrak{M}' \subseteq \mathfrak{M}$ such that* $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{esize}(\psi_k)) = \mathcal{F}$, *we have $\mathfrak{M}', \mathfrak{l} \models \psi_k$.*

So the model checking problem for $\mathrm{MSL}^f(*, \Diamond)$ is in P, and we can conclude.

Theorem 4.9 *The satisfiability problem for* $\mathrm{MSL}(*, \Diamond)$ *is* NP*-complete.*

From Section 2, we recall that $\mathrm{MSL}(\Diamond, \langle \mathsf{gsb} \rangle)$ is defined as a fragment of $\mathrm{MSL}(*, \Diamond)$ with the translation $t(\langle \mathsf{gsb} \rangle \phi) \stackrel{\text{def}}{=} (\mathtt{size} = 1) * t(\phi)$ (global sabotage modal operator). As a corollary of Theorem 4.9, we obtain the result below.

Corollary 4.10 *The satisfiability problem of* $\mathrm{MSL}(\Diamond, \langle \mathsf{gsb} \rangle)$ *is* NP*-complete.*

4.2 The fragment $\mathrm{MSL}(*, \langle \neq \rangle)$

We also establish that the satisfiability problem for $\mathrm{MSL}(*, \langle \neq \rangle)$ is NP-complete and its model-checking problem is in P. To do so, we reduce the problems from $\mathrm{MSL}(*, \langle \neq \rangle)$ to $\mathrm{MSL}^f(*, \langle \neq \rangle)$ and we show a small model property. Given ϕ in $\mathrm{MSL}(*, \langle \neq \rangle)$, let us define its $*$*-weight* $w_*(\phi)$ as follows:

- $w_*(p) \stackrel{\text{def}}{=} 0$, $w_*(\mathsf{emp}) \stackrel{\text{def}}{=} 1$, $w_*(\neg\phi) \stackrel{\text{def}}{=} w_*(\langle \neq \rangle \phi) \stackrel{\text{def}}{=} w_*(\phi)$,
- $w_*(\phi \wedge \psi) \stackrel{\text{def}}{=} \max(w_*(\phi), w_*(\psi))$, $w_*(\phi * \psi) \stackrel{\text{def}}{=} w_*(\phi) + w_*(\psi)$.

Lemma 4.11 *Let* $\alpha \geq 0$ *and* $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ *and* $\mathfrak{M}' = \langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle$ *be* MSL *models such that* $\min(\mathsf{card}(\mathfrak{R}), \alpha) = \min(\mathsf{card}(\mathfrak{R}'), \alpha)$*. Then, for all locations* $\mathfrak{l}$ *and formulae* ϕ *in* $\mathrm{MSL}(*, \langle \neq \rangle)$ *such that* $w_*(\phi) \leq \alpha$*, we have* $\mathfrak{M}, \mathfrak{l} \models \phi$ *iff* $\mathfrak{M}', \mathfrak{l} \models \phi$*.*

As a corollary, if ϕ in $\mathrm{MSL}(*, \langle \neq \rangle)$ is satisfiable, then it has a model with at most $w_*(\phi)$ edges. Let us refine this. Let $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ be an MSL model with $\mathsf{card}(\mathfrak{R}) = \beta$, $\mathfrak{l} \in \mathbb{N}$ and ϕ be in $\mathrm{MSL}(*, \langle \neq \rangle)$ such that $\mathfrak{M}, \mathfrak{l} \models \phi$. Let $\psi_1, \ldots, \psi_N$ be the subformulae of ϕ such that $\langle \neq \rangle \psi_1, \ldots, \langle \neq \rangle \psi_N$ are the only subformulae of ϕ whose outermost connective is $\langle \neq \rangle$. For all $i \in [1, N]$ and all $\beta' \in [0, \beta]$, we define at most *two* locations $\mathfrak{l}_1^{i,\beta'}$ and $\mathfrak{l}_2^{i,\beta'}$ as follows.

- Given $\mathfrak{R}' \subseteq \mathfrak{R}$ with $\mathsf{card}(\mathfrak{R}') = \beta'$, we have $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle, \mathfrak{l}_1^{i,\beta'} \models \psi_i$ and $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle, \mathfrak{l}_2^{i,\beta'} \models \psi_i$. By Lemma 4.11, this definition makes sense as two models with the same valuation and with the same cardinal of the relation satisfy the same formulae.
- If possible we require that $\mathfrak{l}_1^{i,\beta'}$ and $\mathfrak{l}_2^{i,\beta'}$ are distinct, otherwise if there is only one location satisfying ψ_i in $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle$, we require $\mathfrak{l}_1^{i,\beta'} = \mathfrak{l}_2^{i,\beta'}$.
- If no location satisfies ψ_i in $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle$, then by default $\mathfrak{l}_1^{i,\beta'} = \mathfrak{l}_2^{i,\beta'} = \mathfrak{l}$.

Let $\mathfrak{W} \stackrel{\text{def}}{=} \{\mathfrak{l}\} \cup \{\mathfrak{l}_j^{i,\beta'} \mid j \in \{1,2\},\ i \in [1, N],\ \beta' \in [0, \beta]\}$.

Lemma 4.12 *We have* $\langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle, \mathfrak{l} \models \phi$*.*

So, $\mathrm{MSL}(*, \langle \neq \rangle)$ satisfies a small model property.

Corollary 4.13 *Let* ϕ *be a formula in* $\mathrm{MSL}(*, \langle \neq \rangle)$*.* ϕ *is satisfiable iff* ϕ *is* $\mathrm{MSL}^f(*, \langle \neq \rangle)$ *satisfiable in a model with* $\mathsf{card}(\mathfrak{W}) \leq 1 + 2|\phi| \times w_*(\phi)$*.*

It remains to characterise the complexity of the model-checking problem for $\mathrm{MSL}^f(*, \langle \neq \rangle)$.

Lemma 4.14 *The model-checking problem for* $\mathrm{MSL}^f(*, \langle \neq \rangle)$ *is in* P*.*

Proof Let $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ be a finite and functional model, $\mathfrak{l} \in \mathfrak{W}$, and ϕ be a formula in $\mathrm{MSL}(*, \langle \neq \rangle)$. Let $\psi_1, \ldots, \psi_M$ be the subformulae of ϕ ordered in increasing size. We assume $\mathfrak{W} = [0, K]$ for some $K \geq 0$, $\mathfrak{l} = 0$ and $\mathsf{card}(\mathfrak{R}) = \beta$. In order to determine whether $\mathfrak{M}, \mathfrak{l} \models \phi$, we use a labelling algorithm and we complete a table $T(i, j, k)$ with $i \in [0, K]$, $j \in [0, \beta]$ and $k \in [1, M]$ that takes the value 1 iff $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle, i \models \psi_k$ with $\mathsf{card}(\mathfrak{R}') = j$ (dynamic programming is

used here as usual). The polynomial-time upper bound is mainly due to the fact (see Lemma 4.12) that what matters in a partition $\{\mathfrak{R}'_1, \mathfrak{R}'_2\}$ of $\mathfrak{R}' \subseteq \mathfrak{R}$ is the respective cardinalities of $\mathfrak{R}'_1$ and $\mathfrak{R}'_2$.

Algorithm 2 Model Checking MSL$^f(*, \langle\neq\rangle)$

In: A finite and functional model $\mathfrak{M} = \langle [0, K], \mathfrak{R}, \mathfrak{V}\rangle$, $K \geq 0$, an MSL$(*, \langle\neq\rangle)$ formula ϕ
Out: Return 1 iff $\mathfrak{M}, 0 \models \phi$.

function MC$(\mathfrak{M}, \mathfrak{l}, \phi)$
  $\psi_1, \ldots, \psi_M$ subformulae of ϕ in increasing size ▷ $\psi_M = \phi$
  $\beta := \mathsf{card}(\mathfrak{R})$
  for $j \leftarrow 0$ **to** β **do**
    for $k \leftarrow 1$ **to** M **do**
      for $i \leftarrow 0$ **to** K **do**
        case ψ_k **of**
        emp: $T(i, j, k) := 1$ if $(j = 0)$, otherwise 0
        p: $T(i, j, k) := 1$ if $i \in \mathfrak{V}(p)$, otherwise 0
        $\neg\psi_{k'}$: $T(i, j, k) := 1 - T(i, j, k')$ ▷ $k' < k$
        $\psi_{k_1} \wedge \psi_{k_2}$: $T(i, j, k) := \min(T(i, j, k_1), T(i, j, k_2))$ ▷ $k_1, k_2 < k$
        $\langle\neq\rangle\psi_{k'}$: ▷ $k' < k$
          $T(i, j, k) := \max(T(1, j, k'), \ldots, T(i-1, j, k'), T(i+1, j, k'), \ldots, T(K, j, k'))$
        $\psi_{k_1} * \psi_{k_2}$: ▷ $k_1, k_2 < k$
          $T(i, j, k) := \max\{\min(T(i, I, k_1), T(i, J, k_2)) \mid I + J = j \text{ and } I, J \geq 0\}$
        end case
  return $T(0, \beta, M)$

It is worth noting that computing $T(i, j, k)$ always requires values $T(i', j', k')$ that have already got a value and the whole procedure requires polynomial-time in $\beta + M + K$. The correctness of $T(i, j, k) = 1$ iff $\langle\mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, i \models \psi_k$ with $\mathsf{card}(\mathfrak{R}') = j$ is then by an easy verification. The satisfaction of $\mathfrak{M}, \mathfrak{l} \models \phi$ is therefore stored in $T(0, \beta, M)$. □

Again, we are able to establish an NP upper bound.

Theorem 4.15 *The satisfiability problem for* MSL$(*, \langle\neq\rangle)$ *is* NP*-complete.*

5 MSL$(*, \Diamond, \langle\neq\rangle)$: a TOWER-complete fragment of MSL

In this section, we show that the satisfiability problem for MSL$(*, \Diamond, \langle\neq\rangle)$ is TOWER-complete. The upper bound is from Section 3 whereas the proof for TOWER-hardness consists of two parts. First, we show that there is a formula in MSL$(*, \Diamond, \langle\neq\rangle)$ that characterises the linear structures. Then, we reduce the nonemptiness problem for star-free expressions into the satisfiability problem.

5.1 Encoding linear structures

The goal of this section is to design a global formula in MSL$(*, \Diamond, \langle\neq\rangle)$, namely $\phi_{\exists\mathtt{1s}}$, such that for all models $\mathfrak{M}$, we have $\mathfrak{M} \models \phi_{\exists\mathtt{1s}}$ iff either $\mathfrak{R}$ is empty or $\mathfrak{R} = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ for some $n \geq 1$ such that for all $i \neq j \in [0, n]$, we have $\mathfrak{l}_i \neq \mathfrak{l}_j$. In that case, we say that $\mathfrak{M}$ is *linear*. Given a finite set $X \subseteq$ PROP, the relation $\{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ encodes the finite word $b_1 \cdots b_n$ where each letter b_j is equal to $\{p \in X \mid \mathfrak{l}_j \in \mathfrak{V}(p)\}$ (the labelling of the location $\mathfrak{l}_0$ is irrelevant for the encoding). When $\mathfrak{R}$ is empty, the pair $\mathfrak{M}, \mathfrak{l}$ encodes the empty string.

Note that $\phi_{\exists \mathtt{ls}}$ shall be free of propositional variables, which is not so surprising as it expresses a property about the structure of the model. This corresponds to the natural counterpart of the *list segment predicate* $\mathtt{ls}(\mathtt{x},\mathtt{y})$ in separation logic, defined as follows:

$$(\mathfrak{s},\mathfrak{h}) \models \mathtt{ls}(\mathtt{x},\mathtt{y}) \quad \stackrel{\text{def}}{\Leftrightarrow} \quad \begin{array}{l} \text{either } (\mathrm{dom}(\mathfrak{h}) = \emptyset \text{ and } \mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y})) \text{ or} \\ \mathfrak{h} = \{\mathfrak{l}_0 \mapsto \mathfrak{l}_1, \mathfrak{l}_1 \mapsto \mathfrak{l}_2, \ldots, \mathfrak{l}_{n-1} \mapsto \mathfrak{l}_n\} \text{ with } n \geq 1, \\ \mathfrak{l}_0 = \mathfrak{s}(\mathtt{x}),\ \mathfrak{l}_n = \mathfrak{s}(\mathtt{y}) \text{ and for all } i \neq j \in [0,n],\ \mathfrak{l}_i \neq \mathfrak{l}_j. \end{array}$$

So, the formula $\phi_{\exists \mathtt{ls}}$ expresses a property that corresponds to $\exists\ \mathtt{x},\mathtt{y}\ \mathtt{ls}(\mathtt{x},\mathtt{y})$ from (first-order) separation logic.

Given an MSL model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$, let us introduce a few notions that are helpful to build the formula $\phi_{\exists \mathtt{ls}}$. As $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ does not include $\Diamond^{-1}$ and $\langle \star \rangle$ (unlike MLH [21]), we need to characterise linear structures by combining intricate properties. By way of example, stating that each location has at most one predecessor can be easily expressed with $[\mathrm{U}](\neg(\Diamond^{-1}\top * \Diamond^{-1}\top))$, but, obviously, this formula does not belong to $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$.

A *loop* in $\mathfrak{M}$ is a sequence of locations $(\mathfrak{l}_0, \ldots, \mathfrak{l}_n)$ for some $n \geq 1$ such that $\mathfrak{l}_0 = \mathfrak{l}_n$ and for all $i \in [0, n-1]$, $(\mathfrak{l}_i, \mathfrak{l}_{i+1}) \in \mathfrak{R}$. $\mathfrak{M}$ *has at most one maximally connected component* (MCC) whenever for all $\mathfrak{l}, \mathfrak{l}'$ such that $\mathfrak{R}(\mathfrak{l})$ and $\mathfrak{R}(\mathfrak{l}')$ are non-empty, there is $\mathfrak{l}^+$ such that $(\mathfrak{l}, \mathfrak{l}^+) \in \mathfrak{R}^+$ and $(\mathfrak{l}', \mathfrak{l}^+) \in \mathfrak{R}^+$, where $\mathfrak{R}^+$ is the transitive closure of $\mathfrak{R}$. A location $\mathfrak{l}$ is a *leaf* in $\mathfrak{M}$ if $\mathfrak{R}(\mathfrak{l}) \neq \emptyset$ and $\mathfrak{R}^{-1}(\mathfrak{l}) = \emptyset$, and $\mathfrak{l}$ is a *pre-root* if $\mathfrak{R}(\mathfrak{l}) = \{\mathfrak{l}'\}$ for some $\mathfrak{l}'$ and $\mathfrak{R}(\mathfrak{l}') = \emptyset$. In Figure 2 we illustrate these concepts. This terminology making reference to trees is best understood if we think the definitions with respect to $\mathfrak{R}^{-1}$.

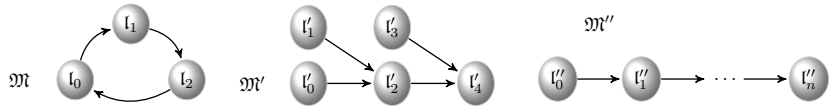


Figure 2. $\mathfrak{M}$ is a MCC and a loop, with no leaves and no pre-roots; $\mathfrak{M}'$ is a MCC with three leaves ($\mathfrak{l}'_0$, $\mathfrak{l}'_1$ and $\mathfrak{l}'_3$) and two pre-roots ($\mathfrak{l}'_2$ and $\mathfrak{l}'_3$); $\mathfrak{M}''$ is linear.

Obviously, if $\mathfrak{M}$ is linear, then it is loop-free, it has at most one MCC and has a unique leaf in case $\mathfrak{M}$ is non-empty. The result below states the converse, and below we explain how to express all these properties.

Lemma 5.1 *Let $\mathfrak{M}$ be an* MSL *model with a non-empty relation. $\mathfrak{M}$ is linear iff $\mathfrak{M}$ is loop-free and has a unique leaf.*

Let us introduce the global formula $\mathsf{Loop} \stackrel{\text{def}}{=} \top * (([\mathrm{U}]\Box\Diamond\top) \wedge \neg\mathtt{emp})$.

Lemma 5.2 *Let $\mathfrak{M}$ be an* MSL *model. $\mathfrak{M} \models \mathsf{Loop}$ iff $\mathfrak{M}$ has at least one loop.*

Let us consider the formulae below (whose semantics is given in Lemma 5.3).

$$\begin{array}{l} \mathsf{PRoot} \stackrel{\text{def}}{=} \Diamond\Box\bot; \quad \mathsf{UniqTreePRoot} \stackrel{\text{def}}{=} \neg\mathsf{Loop} \wedge ((\neg(\neg\mathtt{emp} * \neg\mathtt{emp})) \vee \langle\,!\,\rangle\mathsf{PRoot}) \\ \mathsf{Leaf} \stackrel{\text{def}}{=} (\Diamond\top \wedge \mathtt{size} = 1) \vee \\ \qquad (\Diamond\top \wedge \neg\mathsf{PRoot} \wedge ((\mathtt{size} = 1 \wedge \Diamond\top) * \mathsf{UniqTreePRoot})). \end{array}$$

Lemma 5.3 *Let $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ be a model and $\mathfrak{l} \in \mathbb{N}$.*

(I) *$\mathfrak{M}, \mathfrak{l} \models \mathsf{PRoot}$ iff $\mathfrak{l}$ is a pre-root.*
(II) *$\mathfrak{M}, \mathfrak{l} \models \mathsf{UniqTreePRoot}$ iff $\mathfrak{M}$ is loop-free and either $\mathfrak{R}$ is empty or ($\mathfrak{M}$ has at most one MCC and a unique pre-root).*
(III) *Assuming that $\mathfrak{M} \models \mathsf{UniqTreePRoot}$, we have $\mathfrak{M}, \mathfrak{l} \models \mathsf{Leaf}$ iff $\mathfrak{l}$ is a leaf.*

The proof is rather tedious and is intrinsically related to the definition of the formulae. Let $\phi_{\exists \mathtt{ls}}$ be $\mathtt{emp} \vee (\mathsf{UniqTreePRoot} \wedge \langle ! \rangle \mathsf{Leaf})$. By combination of the previous lemmas and using that if $\mathfrak{M}$ is linear and non-empty, then $\mathfrak{M}$ has at most one MCC and a unique pre-root, we get the result below.

Theorem 5.4 *Let $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ be a model. $\mathfrak{M} \models \phi_{\exists \mathtt{ls}}$ iff $\mathfrak{M}$ is linear.*

The formula $\mathtt{ls}(\mathtt{x}, \mathtt{y})$ can be therefore encoded by the formula below:

$$\phi_{\mathtt{ls}(\mathtt{x},\mathtt{y})} \stackrel{\text{def}}{=} \phi_{\exists \mathtt{ls}} \wedge ((\mathtt{emp} \wedge \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y})) \vee (\langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathsf{Leaf}) \wedge \langle \mathrm{U} \rangle (\mathsf{PRoot} \wedge \Diamond \mathtt{y}))).$$

5.2 The reduction

In this section, we show that the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ is Tower-hard by reduction from the nonemptiness problem for star-free expressions [31,35]. The proof takes advantage of Theorem 5.4 to encode finite words and separating conjunction will be helpful to encode concatenation, whereas complement and union operators in the star-free expressions are taken care by negation and disjunction, respectively. Our proof is reminiscent to developments from [21, Section 3] as it is essential to be able to encode finite words. Instead of reducing the satisfiability problem for Propositional Interval Temporal Logic [32] as done in [21, Section 3], we define a reduction from the nonemptiness problem for star-free expressions. A *star-free expression* e over some alphabet Σ is defined by

$$e ::= a \mid \varepsilon \mid e \cup e \mid ee \mid \sim e,$$

where $a \in \Sigma$ and ε denotes the empty string. Star-free expressions e are interpreted by languages $\mathrm{L}(e) \subseteq \Sigma^*$ as follows:

- $\mathrm{L}(a) \stackrel{\text{def}}{=} \{a\}$ for all $a \in \Sigma$; $\mathrm{L}(\varepsilon) \stackrel{\text{def}}{=} \{\varepsilon\}$; $\mathrm{L}(\sim e) \stackrel{\text{def}}{=} \Sigma^* \setminus \mathrm{L}(e)$;
- $\mathrm{L}(e \cup e') \stackrel{\text{def}}{=} \mathrm{L}(e) \cup \mathrm{L}(e')$; $\mathrm{L}(ee') \stackrel{\text{def}}{=} \{\mathfrak{w}\mathfrak{w}' \in \Sigma^* \mid \mathfrak{w} \in \mathrm{L}(e), \mathfrak{w}' \in \mathrm{L}(e')\}$.

The *nonemptiness problem* consists in checking whether $\mathrm{L}(e) \neq \emptyset$. The problem is shown decidable with a non elementary procedure in [31,37] and refined to Tower-completeness in [35].

Given a finite alphabet $\Sigma = \{a_1, \ldots, a_\alpha\}$, we use the models encoding finite words thanks to the formula $\phi_{\exists \mathtt{ls}}$ and furthermore, we require that $[\mathrm{U}] \bigvee_i \mathsf{a}_i$ where $\mathsf{a}_i \stackrel{\text{def}}{=} p_i \wedge \bigwedge_{j \neq i} \neg p_j$. So, for every $\mathfrak{w} \in \Sigma^*$, there is a pair $\mathfrak{M}, \mathfrak{l}$ encoding $\mathfrak{w}$. We define a relation $\rhd$ that establishes this correspondence: $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l} \stackrel{\text{def}}{\Leftrightarrow}$ $\mathfrak{M}$ is linear and

- If $\mathfrak{w} = \varepsilon$, then $\mathfrak{M}$ has an empty accessibility relation and $\mathfrak{l}$ is arbitrary.
- If $\mathfrak{w} = a_{i_1} \cdots a_{i_n}$ $(n \geq 1)$, then $\mathfrak{M}$ has n edges and $\mathfrak{l}$ is the unique leaf. With $\mathfrak{R} = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$, for all $k \in [1, \alpha]$, $\mathfrak{V}(p_k) = \{\mathfrak{l}_j \mid j \geq 1,\ i_j = k\}$.

The correspondence between finite words in Σ^* and pairs $\mathfrak{M}, \mathfrak{l}$ satisfies a nice property as far as splitting a word into two disjoint subwords is concerned.

Lemma 5.5 *Let $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$ with $\mathfrak{w} = \mathfrak{w}_1\mathfrak{w}_2 \in \Sigma^*$. There exist linear models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ and $\mathfrak{l}'$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathfrak{w}_1 \rhd \mathfrak{M}_1, \mathfrak{l}$ and $\mathfrak{w}_2 \rhd \mathfrak{M}_2, \mathfrak{l}'$.*

Another technical lemma is needed for the proof of Lemma 5.7.

Lemma 5.6 *Let $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$ with $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$ and, $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are linear. There are $\mathfrak{w}_1, \mathfrak{w}_2 \in \Sigma^*$ and $\mathfrak{l}' \in \mathbb{N}$ such that $\mathfrak{w} = \mathfrak{w}_1\mathfrak{w}_2$, $\mathfrak{w}_1 \rhd \mathfrak{M}_1, \mathfrak{l}$ and $\mathfrak{w}_2 \rhd \mathfrak{M}_2, \mathfrak{l}'$.*

Each expression e is translated as

$$T(e) \stackrel{\text{def}}{=} ([\mathrm{U}] \bigvee_i \mathsf{a}_i) \wedge \phi_{\exists \mathtt{ls}} \wedge (\mathsf{emp} \wedge t(e)) \vee (\neg \mathsf{emp} \wedge \mathsf{Leaf} \wedge t(e)),$$

where $t(\cdot)$ is recursively defined. The four disjuncts in $t(e_1e_2)$ below correspond to cases depending on the emptiness of subwords.

$$\begin{array}{llll} t(\varepsilon) & \stackrel{\text{def}}{=} \mathsf{emp} & t(a_i) & \stackrel{\text{def}}{=} (\Diamond \mathsf{a}_i) \wedge \mathtt{size} = 1 \\ t(\sim e) & \stackrel{\text{def}}{=} \neg t(e) & t(e_1 \cup e_2) & \stackrel{\text{def}}{=} t(e_1) \vee t(e_2) \\ t(e_1e_2) & \stackrel{\text{def}}{=} \psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4 & & \end{array}$$

$$\psi_1 \stackrel{\text{def}}{=} \mathsf{emp} \wedge t(e_1) \wedge t(e_2) \quad \psi_2 \stackrel{\text{def}}{=} (t(e_1) \wedge \mathsf{emp}) * t(e_2) \quad \psi_3 \stackrel{\text{def}}{=} t(e_1) * (t(e_2) \wedge \mathsf{emp})$$

$$\psi_4 \stackrel{\text{def}}{=} (\phi_{\exists \mathtt{ls}} \wedge \neg \mathsf{emp} \wedge t(e_1)) * (\phi_{\exists \mathtt{ls}} \wedge \neg \mathsf{emp} \wedge \langle \mathrm{U} \rangle (\mathsf{Leaf} \wedge t(e_2))).$$

In ψ_4, to evaluate $t(e_2)$, we move to the unique leaf of the linear structure.

Lemma 5.7 *Let $\mathfrak{w} \in \Sigma^*$, and $\mathfrak{M}$ be a linear model such that $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$. For every star-free expression e, we have $\mathfrak{w} \in \mathrm{L}(e)$ iff $\mathfrak{M}, \mathfrak{l} \models t(e)$.*

As a consequence,

Lemma 5.8 *Given $\alpha \geq 1$, $\Sigma = \{a_1, \ldots, a_\alpha\}$ and a star-free expression e built on Σ, $\mathrm{L}(e) \neq \emptyset$ iff the formula $T(e)$ is $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ satisfiable.*

Finally, we get the TOWER-completeness.

Theorem 5.9 *The satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ is TOWER-complete.*

6 When the magic wand strikes back

In this short section, we show that the satisfiability problem for MSL is actually undecidable by taking advantage of previous results. All the previous complexity results, deal with fragments that are $-\!\!*$-free. It is well-known that adding the separating connective $-\!\!*$ can dramatically augment the expressive power or the complexity, see e.g. [11]. Below, the expressive strength of $-\!\!*$ is again illustrated, via a reduction from propositional separation logic augmented with the list segment predicate $\mathtt{ls}$ [22]. By contrast, it is known that the modal logic for heaps MLH restricted to $*$ is decidable [21], but it is open whether the addition of $-\!\!*$ leads to undecidability.

First, note that the interval temporal logic with the operators C, D and T over the class of finite strict orders (equivalently, one may consider only the

finite intervals of $\mathbb{N}$) is shown to admit an undecidable satisfiability problem in [26] and to be non recursively enumerable. By contrast, the version of the logic in which the propositional valuation of an interval only depends on the first value of the interval (the locality condition) is decidable as satisfiability can be reduced to the satisfiability problem for first-order logic over $\langle \mathbb{N}, \leq, +1 \rangle$. As we have seen in the paper, the formula $\phi_{\exists \mathtt{ls}}$ can enforce a linear structure whose labelling depends on the first location (corresponding to the locality condition) but it is unclear how to reduce the undecidable version to MSL, even though there is a clear correspondence between the chop operator C and $*$, and between the operators D and T, and $-\!\!*$. Instead, our undecidability proof for (full) MSL is by reducing the satisfiability problem for $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$, recently shown undecidable in [22].

Notice that for the translation of $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ formulae, the most complex part is the encoding of the atomic formulae $\mathtt{ls}(\mathtt{x}, \mathtt{y})$. However, all this work has already been done in Section 5 when we encode linear structures with $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$. Then, what gives us undecidability is essentially the inclusion of the $-\!\!*$ operator.

Let us define the translation $t(\cdot)$ from $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ into MSL formulas, which is homomorphic for Boolean and separation connectives, and

$$t(\mathtt{emp}) \stackrel{\text{def}}{=} \mathtt{emp} \quad t(\mathtt{x} = \mathtt{y}) \stackrel{\text{def}}{=} \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y}) \quad t(\mathtt{x} \hookrightarrow \mathtt{y}) \stackrel{\text{def}}{=} \langle \mathrm{U} \rangle (\mathtt{x} \wedge \Diamond \mathtt{y})$$

$$t(\mathtt{ls}(\mathtt{x}, \mathtt{y})) \stackrel{\text{def}}{=} \phi_{\exists \mathtt{ls}} \wedge ((\mathtt{emp} \wedge \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y})) \vee (\langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathsf{Leaf}) \wedge \langle \mathrm{U} \rangle (\mathsf{PRoot} \wedge \Diamond \mathtt{y}))),$$

where $\mathtt{x}, \mathtt{y}$ are nominals and $\phi_{\exists \mathtt{ls}}$ defined as in Section 5.1. We get the result below.

Lemma 6.1 *Let ϕ be an $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ formula. ϕ is satisfiable iff $t(\phi)$ is satisfiable in* MSL.

As the satisfiability problem for $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ is recently shown undecidable [22], we get the following result.

Theorem 6.2 *The satisfiability problem for* MSL *is undecidable.*

Another consequence is the non-finite axiomatisability of MSL, which is inherited from $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$. As a corollary, the modal logic for heaps MLH (including $-\!\!*$) augmented with propositional variables is undecidable [21] as MSL is one of its fragments.

Furthermore, the satisfiability problem of $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$ is undecidable [2]. Therefore, when considering general models, the minimal modal separation logic $\mathrm{MSL}^g(*, \Diamond)$ is also undecidable (use a map t such that $t(\langle \mathsf{gsb} \rangle \phi) \stackrel{\text{def}}{=} (\mathtt{size} = 1) * t(\phi)$).

7 Conclusion

We have introduced the logic MSL and studied several of its fragments. For $\mathrm{MSL}(*, \Diamond)$, we proved that the satisfiability problem is NP-complete whereas the model-checking problem is in P. A similar complexity characterisation is provided for $\mathrm{MSL}(*, \langle \neq \rangle)$. Surprisingly, we have shown that the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ is TOWER-complete. A key element of our TOWER-

hardness proof is the ability to express the property $\exists\ \mathtt{x}, \mathtt{y}\ \mathtt{ls}(\mathtt{x}, \mathtt{y})$ from separation logic. Hence, we are able to show that MSL admits an undecidable satisfiability problem. Along the paper, we also investigated variants of MSL (or some of its fragments) by slightly modifying the semantics or by adding other modal connectives. For instance, we have proved that the satisfiability problem for $\mathrm{MSL}(\Diamond, \langle \mathsf{gsb} \rangle)$ is (only) NP-complete.

Most of the results are summarised in the table below.

	Model checking (with finite models)	Satisfiability
$\mathrm{MSL}(*, \Diamond)$, $\mathrm{MSL}(*, \langle \neq \rangle)$	P	NP-complete
$\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$	PSPACE-complete	TOWER-complete
MSL	PSPACE-complete	Undecidable
$\mathrm{MSL}(*, \Diamond^{-1})$	PSPACE-complete	PSPACE-hard, in TOWER
$\mathrm{MSL}(\Diamond, \langle \mathsf{gsb} \rangle)$	P	NP-complete

Understanding the effects of the interactions between modal operators and separating connectives is still to be strengthened and many interesting problems are left open. By way of example, we have shown that the satisfiability problem for $\mathrm{MSL}(*, \Diamond^{-1})$ is PSPACE-hard and in TOWER but a complexity characterisation is not yet known. Similarly, the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ is shown TOWER-complete, what about its slight variant $\mathrm{MSL}(*, \Diamond, \langle \mathrm{U} \rangle)$? The decidability status of MSL^f and MLH [21] is also open. Finally, the design of proof systems for modal separation logics remains a challenging question.

Ackowledgements. We would like to thank Alessio Mansutti (LSV, France) for helpful suggestions and enlightning discussions. This work was partially supported by ANPCyT-PICTs-2016-0215, SeCyT-UNC, and the Laboratoire International Associé INFINIS.

References

[1] Areces, C., R. Fervari and G. Hoffmann, *Relation-Changing Modal Operators*, IGPL **23** (2015), pp. 601–627.

[2] Areces, C., R. Fervari, G. Hoffmann and M. Martel, *Satisfiability for relation-changing logics*, JLC (2018), accepted, subject to minor revisions.

[3] Areces, C., R. Fervari, G. Hoffmann and M. Martel, *Undecidability of relation-changing modal logics*, in: *Dynamic Logic. New Trends and Applications - First International Workshop, DALI 2017, Brasilia, Brazil*, LNCS **10669** (2018), pp. 1–16.

[4] Aucher, G., P. Balbiani, L. Fariñas del Cerro and A. Herzig, *Global and local graph modifiers*, ENTCS **231** (2009), pp. 293–307.

[5] Baader, F., I. Horrocks, C. Lutz and U. Sattler, "An Introduction to Description Logic," CUP, 2017.

[6] Berdine, J., C. Calcagno and P. O'Hearn, *A decidable fragment of separation logic*, in: *FST&TCS'04*, LNCS **3328** (2004), pp. 97–109.

[7] Blackburn, P., *Representation, reasoning, and relational structures: a hybrid logic manifesto*, IGPL **8** (2000), pp. 339–365.

[8] Blackburn, P., M. de Rijke and Y. Venema, "Modal Logic," CUP, 2001.

[9] Börger, E., E. Grädel and Y. Gurevich, "The Classical Decision Problem," Perspectives in Mathematical Logic, Springer, 1997.

[10] Brochenin, R., S. Demri and E. Lozes, *Reasoning about sequences of memory states*, APAL **161** (2009), pp. 305–323.

[11] Brochenin, R., S. Demri and E. Lozes, *On the almighty wand*, IC **211** (2012), pp. 106–137.
[12] Brotherston, J. and M. Kanovich, *Undecidability of propositional separation logic and its neighbours*, JACM **61** (2014).
[13] Brotherston, J. and J. Villard, *Parametric completeness for separation theories*, in: *POPL'14* (2014), pp. 453–464.
[14] Calcagno, C., P. O'Hearn and H. Yang, *Computability and complexity results for a spatial assertion language for data structures*, in: *FSTTCS'01*, LNCS **2245** (2001), pp. 108–119.
[15] Calvanese, D., T. Kotek, M. Simkus, H. Veith and F. Zuleger, *Shape and content - A database-theoretic perspective on the analysis of data structures*, in: *IFM'14*, LNCS **8739** (2014), pp. 3–17.
[16] Cook, B., C. Haase, J. Ouaknine, M. Parkinson and J. Worrell, *Tractable reasoning in a fragment of separation logic*, in: *CONCUR'11*, LNCS **6901** (2011), pp. 235–249.
[17] Courtault, J.-R. and D. Galmiche, *A modal BI logic for dynamic resource properties*, in: *LFCS'13*, LNCS **7734** (2013), pp. 134–148.
[18] Dawar, A., P. Gardner and G. Ghelli, *Expressiveness and complexity of graph logic*, IC **205** (2007), pp. 263–310.
[19] de Rijke, M., *The modal logic of inequality*, JSL **57** (1992), pp. 566–584.
[20] Demri, S., *A simple tableau system for the logic of elsewhere*, in: *TABLEAUX'96*, LNAI **1071** (1996), pp. 177–192.
[21] Demri, S. and M. Deters, *Two-variable separation logic and its inner circle*, ACM ToCL **2** (2015).
[22] Demri, S., E. Lozes and A. Mansutti, *The effects of adding reachability predicates in propositional separation logic*, in: *FOSSACS'18*, LNCS **10803** (2018), pp. 476–493.
[23] Galmiche, D., P. Kimmel and D. Pym, *A substructural epistemic resource logic*, in: *ICLA'17*, LNCS **10119** (2017), pp. 106–122.
[24] Georgieva, L. and P. Maier, *Description logics for shape analysis*, in: *SEFM'05* (2005), pp. 321–331.
[25] Herzig, A., *A simple separation logic*, in: *WoLLIC'13*, LNCS **8071** (2013), pp. 168–178.
[26] Hodkinson, I., A. Montanari and G. Sciavicco, *Non-finite axiomatizability and undecidability of interval temporal logics with C, D, and T*, in: *CSL'08*, LNCS **5213** (2008), pp. 308–322.
[27] Ishtiaq, S. and P. O'Hearn, *BI as an assertion language for mutable data structures*, in: *POPL'01* (2001), pp. 14–26.
[28] Larchey-Wendling, D. and D. Galmiche, *Nondeterministic phase semantics and the undecidability of Boolean BI*, ACM ToCL **14** (2013).
[29] Löding, C. and P. Rohde, *Model checking and satisfiability for sabotage modal logic*, in: *FST&TCS'03*, LNCS **2914** (2003), pp. 302–313.
[30] Lutz, C., *Complexity and succinctness of public announcement logic*, in: *AAMAS'06* (2006), pp. 137–143.
[31] Meyer, A. and L. Stockmeyer, *Word problems requiring exponential time*, in: *STOC'73* (1973), pp. 1–9.
[32] Moszkowski, B., *Reasoning about digital circuits*, Technical Report STAN-CS-83-970, Dept. of Computer Science, Stanford University, Stanford, CA (1983).
[33] Rabin, M., *Decidability of second-order theories and automata on infinite trees*, Transactions of the American Mathematical Society **41** (1969), pp. 1–35.
[34] Reynolds, J., *Separation logic: a logic for shared mutable data structures*, in: *LICS'02* (2002), pp. 55–74.
[35] Schmitz, S., *Complexity hierarchies beyond Elementary*, ACM Transactions on Computation Theory **8** (2016), pp. 3:1–3:36.
[36] Segerberg, K., *A note on the logic of elsewhere*, Theoria **47** (1981), pp. 183–187.
[37] Stockmeyer, L., "The complexity of decision problems in automata theory and logic," Ph.D. thesis, Department of Electrical Engineering, MIT (1974).
[38] Stockmeyer, L., *The polynomial-time hierarchy*, TCS **3** (1977), pp. 1–21.
[39] van Benthem, J., *An Essay on Sabotage and Obstruction*, in: *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg Siekmann on the Occasion of his 69th Birthday* (2005), pp. 268–276.