The Power of Modal Separation Logics

Stéphane Demri[*] Raul Fervari[†]

[*]LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, France
demri@lsv.fr
[†]CONICET and Universidad Nacional de Córdoba, Argentina
rfervari@conicet.gov.ar

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, obtaining surprising results by design of 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 difference 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. We establish that the full logic MSL admits an undecidable satisfiability problem. Furthermore, we investigate variants of MSL with alternative semantics and we build bridges with interval temporal logics and with logics equipped with sabotage operators.

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 (see e.g. [46, 1]) in order to verify programs with mutable data structures [48, 58, 61, 59]. 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 [67, 50], logics of public announcements (see e.g., [51]), interval temporal logics [47], relation-changing logics [5, 2, 3], ambient logics or graph logics (see e.g. [18, 11, 27]), propositional team logics [43], second-order modal logics [37] or logics with reactive Kripke semantics [40].

Many other examples of such logics can be found in the literature (see also [27]) 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 [22, 24, 41, 23] and, combinations of temporal logics and propositional separation logics are introduced in [14].

Sometimes, the concept of separation is understood differently and performed at a different level, for instance a simple separation logic is introduced in [45] in which separation is performed on valuations (or equivalently on sets of propositional variables) instead of being performed on heaps. Other examples can be found in [30, 13]. A slightly different approach involving description logics [6] was investigated in [42, 20]. 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 [20].

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. [14, 22, 41, 23]), which allows us 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 [9]. Note that a hybrid extension of Boolean BI is defined in [17], in which nominals are interpreted by heaps whereas herein, the nominals are interpreted by locations.

We investigate the effects of adding new modalities or new separating connectives on the decidability status or on the computational complexity of the logic, following a standard approach to find a reasonable compromise between expressivity and computational properties. Unlike [42, 20], we use simple modal operators (no reflexive transitive closure or converse modality) but we shall include in the logical language standard separating connectives. However, we will be able to express some interesting properties such as $\mathtt{ls}(\mathtt{x},\mathtt{z}) * \mathtt{ls}(\mathtt{y},\mathtt{z}) * \top$ in a very simple way, unlike e.g. [42, Section 3] in which a very expressive description logic is needed. Finally, this work should not be understood as a means to propose alternative assertion languages to separation logics but rather as an investigation to better understand the interplay between modal operators and separating connectives in a modal framework. So far, we have little practical motivations, but our investigation is a first step towards understading new tractable fragments of separation logic, which perhaps can serve at a later stage for program verification.

Our contributions. We introduce the logic MSL whose models are Kripke-style structures with domain the set of natural numbers $\mathbb{N}$ (understood as the set of locations) and the accessibility relation is finite and weakly functional (understood as some heap $\mathfrak{h} : \mathbb{N} \rightarrow_{fin} \mathbb{N}$). In MSL, the modal connectives are $\Diamond$ and the difference modality $\langle\neq\rangle$ [28] whereas the separating connectives are the separating conjunction $*$ and separating implication $\mathrel{-\!\!*}$ (also known as the magic wand operator). These connectives allow us 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 $\mathrel{-\!\!*}$, remove or add edges in the model respectively. The closest logic to MSL is probably the modal logic of heaps MLH [31] 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 $\Diamond^{-1}$ and the reflexive transitive closure modality $\langle\star\rangle$ [44] (unlike MLH that has been actually designed to be easily translated into first-order separation logic restricted to two individual variables). Moreover, MSL shares with some logics from [49, 16] 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 (see also [31, Section 2.4]).

- MSL restricted to $\Diamond$ and $*$, written MSL$(*, \Diamond)$, can be viewed as the minimal 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 (see Theorem 19). The same result is shown for MSL$(*, \langle\neq\rangle)$ by adapting arguments for the logic of elsewhere [63, 29] (see Theorem 25). 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 [19].
- 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 [60] (see Theorem 4). This extends the decidability proof of $1\mathrm{SL}(*)$ from [15] as, now, propositional variables need to be taken into account. In the weak second-order theory, monadic predicates are interpreted as finite sets; so we need to show that the propositional variables in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ can hold true only in a finite amount of locations without altering the satisfiability status.

More surprisingly, even though both $\mathrm{MSL}(*, \Diamond)$ and $\mathrm{MSL}(*, \langle\neq\rangle)$ are NP-complete, we establish that the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ is Tower-complete. Tower-hardness is obtained by reduction from the nonemptiness problem for star-free expressions [54, 64, 62] (see Theorem 34). To do so, we show an essential property: the formula $\exists\, \mathtt{x}, \mathtt{y}\ \mathtt{ls}(\mathtt{x}, \mathtt{y})$ from (first-order) separation logic (see e.g. [8, 21]) can be expressed in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$, which allows us to encode finite words. The notion of Tower-completeness is borrowed from [62].

- Using the fact that $\mathtt{ls}(\mathtt{x}, \mathtt{y})$ can be expressed in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ (a direct consequence of the fact that $\exists\, \mathtt{x}, \mathtt{y}\ \mathtt{ls}(\mathtt{x}, \mathtt{y})$ can be expressed as the inequality modality is known to encode nominals), we also establish that MSL (i.e. $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ augmented with the magic wand $-\!\!*$) admits an undecidable satisfiability problem by using the recent result from [35] about the undecidability of propositional separation logic (with $*$ and $-\!\!*$) augmented with the list segment predicate $\mathtt{ls}$, which itself is based on [15, 32].
- 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 translation from formulae of global sabotage logic [4] into $\mathrm{MSL}(*, \Diamond)$ formulae. As a consequence, the satisfiability problem for the global sabotage logic over MSL models is NP-complete, whereas the satisfiability problem for $\mathrm{MSL}^g(*, \Diamond)$ (i.e. $\mathrm{MSL}(*, \Diamond)$ interpreted over arbitrary Kripke style models) is undecidable.

In Figure 1, we present a map illustrating the complexity and the decidability status of modal separation logics we study herein and other logics involved in the article (formal definitions can be found in the subsequent sections).

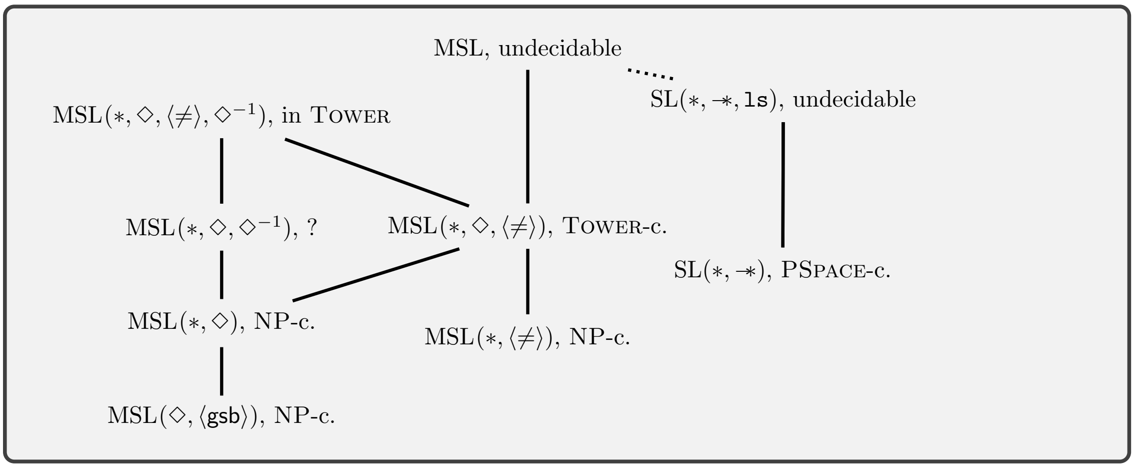

MSL, undecidable
MSL$(*, \Diamond, \langle\neq\rangle, \Diamond^{-1})$, in Tower
SL$(*, -\!\!*, \mathtt{ls})$, undecidable
MSL$(*, \Diamond, \Diamond^{-1})$, ?
MSL$(*, \Diamond, \langle\neq\rangle)$, Tower-c.
SL$(*, -\!\!*)$, PSpace-c.
MSL$(*, \Diamond)$, NP-c.
MSL$(*, \langle\neq\rangle)$, NP-c.
MSL$(\Diamond, \langle\mathsf{gsb}\rangle)$, NP-c.

Figure 1: A complexity map for modal separation logics.

The work presented in this article is an extension of [33]. Herein, we introduce all the contributions in detail, we provide examples and we complete the proofs for all the results.

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. In that way, we also take the opportunity to discuss a bit the expressivity of the language for MSL.

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:

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

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 weakly functional, and $\mathfrak{V} : \mathrm{PROP} \to \mathcal{P}(\mathbb{N})$. This means that for all $\mathfrak{l}, \mathfrak{l}', \mathfrak{l}''$, we have $\mathfrak{l}\mathfrak{R}\mathfrak{l}'$ and $\mathfrak{l}\mathfrak{R}\mathfrak{l}''$ imply $\mathfrak{l}' = \mathfrak{l}''$ (this is also known as being deterministic). In the sequel, by 'functional', we mean 'weakly functional'. 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 $\mathfrak{R}_1 \subseteq \mathfrak{R}_2$. We say $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ is a *partition* of $\mathfrak{R}$, if $\mathfrak{R}_1 \cap \mathfrak{R}_2 = \emptyset$ and $\mathfrak{R} = \mathfrak{R}_1 \cup \mathfrak{R}_2$. Sometimes, we will write $\mathfrak{R}(\mathfrak{l})$ for the set $\{\mathfrak{l}' \mid (\mathfrak{l}, \mathfrak{l}') \in \mathfrak{R}\}$. Given $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and $\mathfrak{l} \in \mathbb{N}$, the satisfaction relation $\models$ is defined below:

$$\begin{array}{lcl}
\mathfrak{M}, \mathfrak{l} \models p & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{l} \in \mathfrak{V}(p) \\
\mathfrak{M}, \mathfrak{l} \models \texttt{emp} & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{R} = \emptyset \\
\mathfrak{M}, \mathfrak{l} \models \neg\phi & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{M}, \mathfrak{l} \not\models \phi \\
\mathfrak{M}, \mathfrak{l} \models \phi_1 \vee \phi_2 & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{M}, \mathfrak{l} \models \phi_1 \text{ or } \mathfrak{M}, \mathfrak{l} \models \phi_2 \\
\mathfrak{M}, \mathfrak{l} \models \Diamond\phi & \overset{\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 & \overset{\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 & \overset{\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 & \overset{\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. [10, 61]. A similar composition operator has been introduced in graph logics [27]. Other standard connectives or formulae are used:

- $[\neq]\phi \overset{\text{def}}{=} \neg\langle\neq\rangle\neg\phi$ and $\Box\phi \overset{\text{def}}{=} \neg\Diamond\neg\phi$; $\langle \mathrm{U} \rangle\phi \overset{\text{def}}{=} \phi \vee \langle\neq\rangle\phi$ and $[\mathrm{U}]\phi \overset{\text{def}}{=} \neg\langle \mathrm{U} \rangle\neg\phi$,
- $\langle\,!\,\rangle\phi \overset{\text{def}}{=} \langle \mathrm{U} \rangle(\phi \wedge [\neq]\neg\phi)$ (unicity of the satisfaction of ϕ),
- the atomic formula $\texttt{size} = 1$ is a shortcut for $\neg\texttt{emp} \wedge \neg(\neg\texttt{emp} * \neg\texttt{emp})$. In fact, this formula can be generalised: $\texttt{size} \geq 0 \overset{\text{def}}{=} \top$, and for $n > 0$, $\texttt{size} \geq n \overset{\text{def}}{=} \overbrace{\neg\texttt{emp} * \cdots * \neg\texttt{emp}}^{n \text{ times}}$. Therefore, $\texttt{size} = n$ is a shortcut for $\texttt{size} \geq n \wedge \neg\texttt{size} \geq n + 1$.

The *satisfiability problem* for the logic MSL, takes as input a formula ϕ in MSL 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 $\texttt{emp}$, 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$(*, \Diamond, \langle\neq\rangle)$.

Besides, the closest logic with MSL is most probably the modal logic of heaps MLH defined in [31] but MLH has no propositional variables and contains also $\Diamond^{-1}$ (associated to the converse relation for $\Diamond$, see also Section 3.2) and the reflexive transitive closure modality $\langle\star\rangle$. Whereas MLH restricted to the separating connective $*$ (written MLH($*$)) has a decidable satisfiability problem [31], the decidability status of MLH is open. The choice of the modal operators for MLH has been guided in [31] by the ability to translate MLH into the first-order separation logic restricted to two individual variables, while showing that the restriction to $*$ (and therefore without the magic wand operator $\mathrel{-\!\!*}$) is TOWER-hard. Herein, we consider a subset of modal operators but with the presence of propositional variables, leading to TOWER-hardness of MSL($*, \Diamond, \langle\neq\rangle$). Of course, it is always possible to add more modal operators to MSL($*, \Diamond, \langle\neq\rangle$) or to MSL, but the computational properties of such extensions will certainly not improve.

2.2 Nominals, program variables and separation logic in a nutshell

In all the fragments of MSL containing the inequality modality [28], it is known that nominals from hybrid logics [9] can be used since stating that p holds true in a unique location can be expressed by $\langle\,!\,\rangle p$. For example, the formula $\phi_1 = \langle\,!\,\rangle p \wedge \langle\,!\,\rangle q \wedge \langle \mathrm{U} \rangle (p \wedge \Diamond q)$ states that p and q are nominals and there is an edge between the unique location where p holds true and the unique location where q holds true (possibly the same location). So, as soon as the inequality modality $\langle\neq\rangle$ is present, we can freely use nominals. Syntactically, nominals are taken from the set 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 [42]. So, herein, checking the satisfiability status of a formula ϕ containing $\mathtt{x}_1, \ldots, \mathtt{x}_n$ actually amounts to checking the satisfiability status of $(\bigwedge_{1 \leq i \leq n} \langle\,!\,\rangle \mathtt{x}_i) \wedge \phi$. Similarly, the formula $\phi_2 = \langle\,!\,\rangle p \wedge \langle\,!\,\rangle q \wedge \langle \mathrm{U} \rangle (p \wedge q)$ states that p and q are nominals and hold true exactly on the same location.

In order to provide some more examples about the expressive power of MSL, $\phi_3 = [\mathrm{U}]\Box\perp$ is logically equivalent to $\mathtt{emp}$, and $\neg\mathtt{emp} \wedge \neg(\neg\mathtt{emp} * \neg\mathtt{emp})$ states that the accessibility relation has a unique edge (also written $\mathtt{size} = 1$). Notice that even though $\mathtt{emp}$ can be taken as a shortcut for ϕ_3, it is not expressible in any of the fragments MSL($*, \Diamond$) and MSL($*, \langle\neq\rangle$), since it uses modalities from both logics. Therefore, we need to include it as primitive in the language.

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 $\mathfrak{M}, \mathfrak{l} \models \phi$). The above-mentioned formulae ϕ_1, ϕ_2, ϕ_3 and $\mathtt{size} = 1$ are global as well as any formula built from them using Boolean and separating connectives.

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

- $\mathfrak{s} : \mathrm{PVAR} \to \mathbb{N}$ (the *store*) and,

- $\mathfrak{h} : \mathbb{N} \to_{\mathrm{fin}} \mathbb{N}$ is a partial function with finite domain (the *heap*).

Models of the separation logic SL($*, \mathrel{-\!\!*}$) (with one record field) 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 SL($*, \mathrel{-\!\!*}$) are built from the grammar

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

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

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

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 $\mathfrak{R} = \{(\mathfrak{l},\mathfrak{h}(\mathfrak{l})) \mid \mathfrak{l}\in\mathrm{dom}(\mathfrak{h})\}$ and the restriction of $\mathfrak{V}$ to PVAR is defined as $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{s}(\mathtt{x})\}$. Actually, any formula ϕ of SL$(*, \mathrel{-\!\!*})$ is satisfiable iff $t(\phi)$ is satisfiable in MSL where t is homomorphic for Boolean and separating connectives and,

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

It is worth noting that each formula $t(\phi)$ is a global formula of MSL where $\mathtt{x}$ and $\mathtt{y}$ are program variables but in general can be enforced to behave as nominals in MSL. So, SL$(*, \mathrel{-\!\!*})$ can be understood as a syntactic fragment of MSL. For instance, the satisfiability problem for SL$(*, \mathrel{-\!\!*})$ is known to be PSpace-complete [19].

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 $\mathfrak{V} : \mathrm{PROP}\to\mathcal{P}(\mathfrak{W})$. 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 $\mathrel{-\!\!*}$, the definition of $\models$ for $\mathfrak{L}^g$ is therefore updated as follows:

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

Note that the formula $(\top \mathrel{-\!\!*} \neg((\neg\mathtt{emp}) \mathrel{-\!\!*} \bot))$ is valid for MSL but not for MSL^f.

The *model-checking problem for* MSL^f takes as input a formula in MSL^f built over some finite set of propositional variables $\{p_1,\ldots,p_n\}\subseteq\mathrm{PROP}$, a finite and functional model $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V}\rangle$ such that $\mathfrak{V}$ is restricted to $\{p_1,\ldots,p_n\}$ and $\mathfrak{l}\in\mathfrak{W}$, and we check whether $\mathfrak{M},\mathfrak{l}\models\phi$. This is a standard way to define the model-checking problem. As MSL can be viewed as a fragment of second-order logic in which second-order quantifications are performed with predicates of arity at most two (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 MSL$^f(*, \langle\neq\rangle)$ or MSL$^f(*, \Diamond)$ is in P (the model-checking problem for the pure modal logic MSL$^f(\Diamond, \langle\neq\rangle)$ is clearly in P too), whereas the restriction to MSL$^f(*, \Diamond, \langle\neq\rangle)$ is already untractable.

Lemma 1. *The model-checking problem for* MSL$^f(*, \Diamond, \langle\neq\rangle)$ *is* PSpace-*hard.*

We show below that the model-checking problem for MSL$^f(*, \Diamond, \langle\neq\rangle)$ is PSpace-hard by reduction from QBF. QBF formulae are built from propositional formulae with the addition of propositional quantifications of the form $\forall\ p$ and $\exists\ p$.

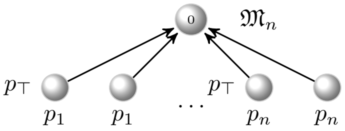
Proof. Let $\mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$ be a QBF formula with $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_n\}\subseteq\{\exists,\forall\}$ and ϕ is a propositional formula built over the atomic propositions in $\{p_1,\ldots,p_n\}$. The formula is said to be in prenex normal form and every QBF formula can be reduced in logarithmic space to an equivalent formula in such a form. Given a propositional valuation $v : \mathrm{PROP}\to\{\bot,\top\}$, we have $v\models\exists\ p\ \phi$

iff there is $b \in \{\bot, \top\}$ such that $v[p \mapsto b] \models \phi$. Similarly, $v \models \forall\ p\ \phi$ iff for all $b \in \{\bot, \top\}$, we have $v[p \mapsto b] \models \phi$. Satisfiability problem for QBF formulae is known to be PSpace-complete [65].

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 $\{p_\top, p_1, \ldots, p_n\}$.

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



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

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

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 $\phi_{n+1} = \phi$. Note that the atomic propositions in ϕ_j that are not in the scope of a propositional quantification belongs to the (possibly empty) set $\{p_i : i \in [1, j-1]\}$.

Given $\mathfrak{M} \subseteq \mathfrak{M}_n$ and a propositional valuation v, we write $\mathfrak{M} \approx_j v$ to denote the conjunction of the following properties.

1. For all $i \in [1, j-1]$, exactly one location in $\{i, n+i\}$ has an outgoing edge and it points to 0.

2. For all $i \in [j, n]$, all the locations in $\{i, n+i\}$ have an outgoing edge and they point to 0.

3. For all $i \in [1, j-1]$, i has an outgoing edge (and it is pointing to 0) iff $v(p_i) = \top$ (iff $n+i$ has no outgoing edges).

By induction on j, we will show that for all $j \in [1, n+1]$, if $\mathfrak{M} \approx_j v$, then $\mathfrak{M}, 0 \models t(\phi_j)$ iff $v \models \phi_j$.

Base case: $j = n+1$. Suppose $\mathfrak{M} \approx_{n+1} v$. The proof is by structural induction and the cases for the Boolean connectives in the induction step are by an easy verification. For $i \in [1, n]$, i.e. $i < n+1$, we have $\mathfrak{M}, 0 \models t(p_i)$, iff (by definition of t) $\mathfrak{M}, 0 \models \langle \neq \rangle(p_i \wedge p_\top \wedge \Diamond\top)$, iff by $\models$, there is $k \in [1, 2n]$ such that $\mathfrak{M}, k \models p_i \wedge p_\top \wedge \Diamond\top$. By construction of $\mathfrak{M}_n$ and (1.) in the definition of $\mathfrak{M} \approx_{n+1} v$, this is equivalent to $\mathfrak{M}, i \models p_i \wedge p_\top \wedge \Diamond\top$. By (3.) in the definition of $\mathfrak{M} \approx_{n+1} v$, $v(p_i) = \top$, which is equivalent to $v \models p_i$

Induction step: $1 \leq j < n+1$. The induction hypothesis is the following: for all $j' \in [j+1, n+1]$, if $\mathfrak{M}' \approx_{j'} v'$, then $\mathfrak{M}', 0 \models t(\phi_{j'})$ iff $v' \models \phi_{j'}$.

Suppose that $\mathfrak{M} \approx_j v$, and first suppose that $\mathfrak{M}, 0 \models t(\exists\ p_j\ \phi_{j+1})$. By definition of t, we have $\mathfrak{M}, 0 \models (\mathtt{size} = 1 \wedge \langle \neq \rangle(p_j \wedge \Diamond\top)) * t(\phi_{j+1})$. This means that there exists a partition of $\mathfrak{M}$ into $\mathfrak{M}', \mathfrak{M}''$ (i.e. $\mathfrak{M} = \mathfrak{M}' \uplus \mathfrak{M}''$) such that $\mathfrak{M}', 0 \models \mathtt{size} = 1 \wedge \langle \neq \rangle(p_i \wedge \Diamond\top)$ and $\mathfrak{M}'', 0 \models t(\phi_{j+1})$. Let $\mathfrak{R}'$ the accessibility relation associated to $\mathfrak{M}'$, then $\mathsf{card}(\mathfrak{R}') = 1$, and there exists $\mathfrak{l} \neq 0$ such that $\mathfrak{M}', \mathfrak{l} \models p_j$ and $(\mathfrak{l}, 0) \in \mathfrak{R}'$. Since $\mathfrak{M}'$ is a submodel of $\mathfrak{M}_n$, either $\mathfrak{l} = j$ or $\mathfrak{l} = n+j$. If $\mathfrak{l} = j$, then we have $\mathfrak{M}'' \approx_{j+1} v[p_j \mapsto \bot]$, otherwise $\mathfrak{M}'' \approx_{j+1} v[p_j \mapsto \top]$. Consequently, by (IH), either $v[p_j \mapsto \bot] \models \phi_{j+1}$ or $v[p_j \mapsto \top] \models \phi_{j+1}$, which is equivalent to $v \models \exists\ p_j\ \phi_{j+1}$. The proof for the other direction is similar. Moreover, the proof for $t(\forall\ p_j\ \phi_{j+1})$ is a direct consequence of the fact that $\forall\ p_j\ \phi_{j+1}$ is logically equivalent to $\neg\exists\ p_j\ \neg\phi_{j+1}$ and t faithfully reflects this duality.

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 [3], that are logics with operators to add, remove and swap around edges of the accessibility relation of the model. 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:

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

where $\mathfrak{M}^-_{\mathfrak{l}',\mathfrak{l}''} = \langle \mathfrak{W}, \mathfrak{R} \setminus \{(\mathfrak{l}', \mathfrak{l}'')\}, \mathfrak{V} \rangle$. $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$ can be encoded into $\mathrm{MSL}^g(*, \Diamond)$ by a translation map t from $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$-formulas to $\mathrm{MSL}^g(*, \Diamond)$-formulas, that is homomorphic for Boolean connectives and for $\Diamond$ and,

$$t(\langle \mathsf{gsb} \rangle \phi) \overset{\text{def}}{=} (\mathtt{size} = 1) * t(\phi).$$

We have ϕ is satisfiable iff $t(\phi)$ is satisfiable for $\mathrm{MSL}^g(*, \Diamond)$. Similarly, the logic $\mathrm{MSL}(\Diamond, \langle \mathsf{gsb} \rangle)$ is the variant of $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$ with MSL models.

3 Decision problems in Tower

Below, we establish that the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ is in Tower [62], the class of problems of time complexity bounded by a tower of exponentials, whose height is an elementary function of the input. Tower-hardness shall be established in Section 5.

3.1 Internalising the semantics

We will design a reduction to the satisfiability problem for $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ and then we will show that the satisfiability problem for $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ is in Tower by translation into the weak MSO theory of one unary function. Notice that the difference between $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ and $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ is that models in $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ have finite relations over an infinite set of locations, while the set of locations in $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ models is finite. This proof is analogous to the decidability proof for the separation logic $1\mathrm{SL}(*)$ in [15] but our main technical task is to solve the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ by using only propositional variables that hold true on a finite amount of locations. First, we show a preliminary result: locations satisfying the same propositional variables and with no successor satisfy the same formulae.

Lemma 2. *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 $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ built over $p_1, \ldots, p_n$, we have $\mathfrak{M}, \mathfrak{l} \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$.*

Proof. The proof is by structural induction on ϕ. When ϕ is a propositional variable in $p_1, \ldots, p_n$, the property trivially holds as by assumption $\mathfrak{l}$ and $\mathfrak{l}'$ agree on $p_1, \ldots, p_n$. Similarly, when $\phi = \mathtt{emp}$, the property holds obviously. Let us treat the different cases in the induction step (we omit the obvious cases for the Boolean connectives).

$\phi = \phi_1 * \phi_2$**.** Suppose that $\mathfrak{M}, \mathfrak{l} \models \phi$. By definition of $\models$, there are models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathfrak{M}_1, \mathfrak{l} \models \phi_1$ and $\mathfrak{M}_2, \mathfrak{l} \models \phi_2$. By the induction hypothesis, $\mathfrak{M}_1, \mathfrak{l}' \models \phi_1$ and $\mathfrak{M}_2, \mathfrak{l}' \models \phi_2$. Consequently, $\mathfrak{M}, \mathfrak{l}' \models \phi$.

$\phi = \Diamond \phi'$**.** By $\mathfrak{R}(\mathfrak{l}) = \mathfrak{R}(\mathfrak{l}') = \emptyset$, $\mathfrak{M}, \mathfrak{l} \not\models \phi$ and $\mathfrak{M}, \mathfrak{l}' \not\models \phi$.

$\phi = \langle \neq \rangle \phi'$**.** Suppose that $\mathfrak{M}, \mathfrak{l} \models \phi$. So, there is $\mathfrak{l}'' \neq \mathfrak{l}$ such that $\mathfrak{M}, \mathfrak{l}'' \models \phi'$. If $\mathfrak{l}'' \neq \mathfrak{l}'$, then $\mathfrak{M}, \mathfrak{l}' \models \phi$. Otherwise, if $\mathfrak{l}'' = \mathfrak{l}'$, then by the induction hypothesis, $\mathfrak{M}, \mathfrak{l} \models \phi'$ and therefore $\mathfrak{M}, \mathfrak{l}' \models \phi$. □

Let ϕ in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ be built over $p_1, \ldots, p_n$. Let us define $T(\phi)$ as the formula below:

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

When $\langle\neq\rangle$ is not present in ϕ, the second conjunct can be removed (see Lemma 12, 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 $\mathrm{MSL}^f(*, \langle\neq\rangle)$ but not for the logic $\mathrm{MSL}(*, \langle\neq\rangle)$.

Lemma 3. *Every formula ϕ is satisfiable in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ iff the formula $T(\phi)$ is satisfiable in $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$.*

Proof. First, suppose $\mathfrak{M}, \mathfrak{l} \models \phi$ with $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$. Let us build a finite and functional model $\mathfrak{M}' = \langle \mathfrak{W}', \mathfrak{R}', \mathfrak{V}' \rangle$ such that $\mathfrak{W}' \subseteq \mathbb{N}$ (to be defined below), $\mathfrak{R}' = \mathfrak{R}$ and $\mathfrak{V}'$ is the restriction of $\mathfrak{V}$ to $\mathfrak{W}'$.

- $\mathfrak{W}'_0 = \{\mathfrak{l}', \mathfrak{l}'' \mid (\mathfrak{l}', \mathfrak{l}'') \in \mathfrak{R}\} \subseteq \mathfrak{W}'$.
- For every $X \subseteq \{p_1, \ldots, p_n\}$, pick two locations $\mathfrak{l}_1^X$ and $\mathfrak{l}_2^X$ to be in $\mathbb{N} \setminus \mathfrak{W}'_0$ (or less than two, if there is only one, or none but optimise the number up to two) such that for all $i \in [1, n]$, we have $\mathfrak{l}_1^X \in \mathfrak{V}(p_i)$ iff $p_i \in X$ iff $\mathfrak{l}_2^X \in \mathfrak{V}(p_i)$. Furthermore, note that each location $\mathfrak{l}_j^X$ has no edge from it in $\mathfrak{R}$.

The set $\mathfrak{W}'$ is defined as the union of $\mathfrak{W}'_0 \cup \{\mathfrak{l}\}$ and the set of locations $\mathfrak{l}_1^X$ and $\mathfrak{l}_2^X$ for all X. By structural induction, one can show that for all $\mathfrak{l}' \in \mathfrak{W}'$, we have $\mathfrak{M}', \mathfrak{l}' \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$. Consequently, $\mathfrak{M}', \mathfrak{l} \models \phi$.

Here is the proof by induction. Actually, we show that given any finite $\mathbb{N} \supset \mathfrak{W}'' \supseteq \mathfrak{W}'$, we have $\mathfrak{M}'' = \langle \mathfrak{W}'', \mathfrak{R}', \mathfrak{V}'' \rangle, \mathfrak{l}' \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$ where $\mathfrak{V}''$ is the restriction of $\mathfrak{V}$ to $\mathfrak{W}''$. This slight generalisation is considered in order to handle the separating conjunction $*$. The base cases are by an easy verification as well as the cases in the induction step for the Boolean connectives.

- Suppose that $\mathfrak{M}, \mathfrak{l}' \models \Diamond\psi$. So, there is $\mathfrak{l}'' \in \mathfrak{R}(\mathfrak{l}')$ such that $\mathfrak{M}, \mathfrak{l}'' \models \psi$. So, $\mathfrak{l}'' \in \mathfrak{W}'_0 \subseteq \mathfrak{W}' \subseteq \mathfrak{W}''$ and by the induction hypothesis $\mathfrak{M}'', \mathfrak{l}'' \models \psi$. As $\mathfrak{R}' = \mathfrak{R}$, we conclude that $\mathfrak{M}'', \mathfrak{l}' \models \Diamond\psi$.

  Conversely, suppose that $\mathfrak{M}'', \mathfrak{l}' \models \Diamond\psi$. So, there is $\mathfrak{l}'' \in \mathfrak{R}'(\mathfrak{l}')$ such that $\mathfrak{M}'', \mathfrak{l}'' \models \psi$. By the induction hypothesis $\mathfrak{M}, \mathfrak{l}'' \models \psi$. As $\mathfrak{R}' = \mathfrak{R}$, we conclude that $\mathfrak{M}, \mathfrak{l}' \models \Diamond\psi$.

- Suppose that $\mathfrak{M}, \mathfrak{l}' \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R}$ such that $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, we have $\langle \mathfrak{W}'', \mathfrak{R}_1, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}'', \mathfrak{R}_2, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_2$ and therefore $\mathfrak{M}'', \mathfrak{l}' \models \psi_1 * \psi_2$. Here, it is essential to use the generalisation as $\mathfrak{W}''$ is most probably bigger than the set of locations that would be extracted from $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle$ or from $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle$.

  Conversely, suppose that $\mathfrak{M}'', \mathfrak{l}' \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R} = \mathfrak{R}'$ such that $\langle \mathfrak{W}'', \mathfrak{R}_1, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}'', \mathfrak{R}_2, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, we have that $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_2$, whence $\mathfrak{M}, \mathfrak{l}' \models \psi_1 * \psi_2$.

- Suppose that $\mathfrak{M}, \mathfrak{l}' \models \langle\neq\rangle\psi$. So, there is $\mathfrak{l}'' \neq \mathfrak{l}'$ such that $\mathfrak{M}, \mathfrak{l}'' \models \psi$. If either $\mathfrak{l}'' \in \mathfrak{W}'_0$ or $\mathfrak{l}'' = \mathfrak{l}$ or $\mathfrak{l}''$ is equal to some $\mathfrak{l}_i^X$, then $\mathfrak{l}'' \in \mathfrak{W}'$ and by the induction hypothesis, we have $\mathfrak{M}'', \mathfrak{l}'' \models \psi$, and therefore $\mathfrak{M}'', \mathfrak{l}' \models \langle\neq\rangle\psi$. Otherwise, necessarily, there is some $\mathfrak{l}_i^X$ such that $\mathfrak{l}''$ and $\mathfrak{l}_i^X$ agree on $p_1, \ldots, p_n$, $\mathfrak{l}''$ and $\mathfrak{l}_i^X$ are distinct, and $\mathfrak{R}(\mathfrak{l}'') = \mathfrak{R}(\mathfrak{l}_i^X) = \emptyset$. By Lemma 2, we get that $\mathfrak{M}, \mathfrak{l}_i^X \models \psi$ and therefore we can conclude that $\mathfrak{M}'', \mathfrak{l}'' \models \psi$ as in the first case.

  Conversely, $\mathfrak{M}'', \mathfrak{l}' \models \langle\neq\rangle\psi$. So, there is $\mathfrak{l}'' \neq \mathfrak{l}'$ such that $\mathfrak{M}'', \mathfrak{l}'' \models \psi$. By the induction hypothesis, we get $\mathfrak{M}, \mathfrak{l}'' \models \psi$ and therefore $\mathfrak{M}, \mathfrak{l}' \models \langle\neq\rangle\psi$.

Moreover, as $\mathbb{N}$ is infinite, there is a set $X \subseteq \{p_1, \ldots, p_n\}$ such that two distinct locations $\mathfrak{l}_1^X$ and $\mathfrak{l}_2^X$ can be found in $\mathfrak{M}$. So, obviously we have

$$\mathfrak{M}', \mathfrak{l} \models \bigvee_{X \subseteq \{p_1,\ldots,p_n\}} \langle \mathrm{U} \rangle (\Box \perp \wedge \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \wedge \langle \neq \rangle (\Box \perp \wedge \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p)).$$

Now, suppose that $\mathfrak{M}, \mathfrak{l} \models T(\phi)$ for some finite and functional model $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ with $\mathfrak{W} \subset \mathbb{N}$. Let us build the model $\mathfrak{M}' = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V}' \rangle$ such that $\mathfrak{V}$ is the restriction of $\mathfrak{V}'$ to $\mathfrak{W}$. We know that there are $\mathfrak{l}_1 \neq \mathfrak{l}_2$ and $X \subseteq \{p_1, \ldots, p_n\}$ such that $\mathfrak{M}, \mathfrak{l}_1 \models \Box \perp \wedge \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p$ and $\mathfrak{M}, \mathfrak{l}_2 \models \Box \perp \wedge \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p$. For all $\mathfrak{l}' \in \mathbb{N} \setminus \mathfrak{W}$ and $i \in [1, n]$, we set $\mathfrak{l}' \in \mathfrak{V}(p_i)$ iff $p_i \in X$. By structural induction, one can show that for all $\mathfrak{l}' \in \mathfrak{W}$, we have $\mathfrak{M}', \mathfrak{l}' \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$.

Here is the proof by induction. Again, the base cases are by an easy verification as well as the cases in the induction step for the Boolean connectives.

- Suppose that $\mathfrak{M}, \mathfrak{l}' \models \Diamond \psi$. So, there is $\mathfrak{l}'' \in \mathfrak{R}(\mathfrak{l}')$ such that $\mathfrak{M}, \mathfrak{l}'' \models \psi$. By the induction hypothesis $\mathfrak{M}', \mathfrak{l}'' \models \psi$ and we conclude that $\mathfrak{M}', \mathfrak{l}' \models \Diamond \psi$.

  Conversely, suppose that $\mathfrak{M}', \mathfrak{l}' \models \Diamond \psi$. So, there is $\mathfrak{l}'' \in \mathfrak{R}'(\mathfrak{l}')$ such that $\mathfrak{M}', \mathfrak{l}'' \models \psi$. By the induction hypothesis $\mathfrak{M}, \mathfrak{l}'' \models \psi$ as $\mathfrak{l}'' \in \mathfrak{W}$. We conclude that $\mathfrak{M}, \mathfrak{l}' \models \Diamond \psi$.

- Suppose that $\mathfrak{M}, \mathfrak{l}' \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R}$ such that $\langle \mathfrak{W}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, we have $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V}' \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V}' \rangle, \mathfrak{l}' \models \psi_2$ and therefore $\mathfrak{M}', \mathfrak{l}' \models \psi_1 * \psi_2$ (here no need for generalised induction hypothesis as the properties for $\mathfrak{l}_1^X$ and $\mathfrak{l}_2^X$ are preserved in $\langle \mathfrak{W}, \mathfrak{R}_1, \mathfrak{V} \rangle$ and $\langle \mathfrak{W}, \mathfrak{R}_2, \mathfrak{V} \rangle$).

  Conversely, suppose that $\mathfrak{M}', \mathfrak{l}' \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R}$ such that $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V}' \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V}' \rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, we have $\langle \mathfrak{W}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_2$, whence $\mathfrak{M}, \mathfrak{l}' \models \psi_1 * \psi_2$.

- Suppose that $\mathfrak{M}, \mathfrak{l}' \models \langle \neq \rangle \psi$. So, there is $\mathfrak{l}'' \neq \mathfrak{l}'$ such that $\mathfrak{M}, \mathfrak{l}'' \models \psi$. By the induction hypothesis, we have $\mathfrak{M}', \mathfrak{l}'' \models \psi$ and therefore $\mathfrak{M}', \mathfrak{l}' \models \langle \neq \rangle \psi$.

  Conversely, suppose that $\mathfrak{M}', \mathfrak{l}' \models \langle \neq \rangle \psi$. So, there is $\mathfrak{l}'' \neq \mathfrak{l}'$ such that $\mathfrak{M}', \mathfrak{l}'' \models \psi$. If $\mathfrak{l}'' \in \mathfrak{W}$, then by the induction hypothesis, we get $\mathfrak{M}, \mathfrak{l}'' \models \psi$ and therefore $\mathfrak{M}, \mathfrak{l}' \models \langle \neq \rangle \psi$. Otherwise as $\mathfrak{l}'' \in \mathbb{N} \setminus \mathfrak{W}$, by construction of $\mathfrak{M}'$, there some $i \in \{1, 2\}$ such that $\mathfrak{l}_i$ and $\mathfrak{l}''$ agree on $p_1, \ldots, p_n$, $\mathfrak{R}(\mathfrak{l}'') = \mathfrak{R}(\mathfrak{l}_i) = \emptyset$ and, $\mathfrak{l}_i \notin \{\mathfrak{l}', \mathfrak{l}''\}$. By Lemma 2, $\mathfrak{M}', \mathfrak{l}_i \models \psi$. As $\mathfrak{l}_i \in \mathfrak{W}$, by the induction hypothesis $\mathfrak{M}, \mathfrak{l}_i \models \psi$. Consequently, $\mathfrak{M}, \mathfrak{l}' \models \langle \neq \rangle \psi$. □

The complexity class TOWER has been introduced in [62] and sits between the class of elementary problems and the class of primitive recursive problems. Examples of standard problems that are TOWER-complete can be found in [62]. It is worth noting that to prove Theorem 4 below, we follow an approach known as the reduction method [39] that is based on the seminal result by Rabin [60].

Theorem 4. *The satisfiability problem for* $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ *is in* TOWER.

Proof. By Lemma 3, there is a reduction from the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ into the satisfiability problem for $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ that works in exponential time. Now, we show that there is a logspace reduction from the satisfiability problem for $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ into the satisfiability problem for the weak monadic second-order theory of one unary function whose structures are triple $\langle D, \mathfrak{f}, = \rangle$ where D is a countable domain, $\mathfrak{f}$ is a unary function, and $=$ is equality ('weakness' refers to the fact that the monadic predicates are interpreted by *finite* sets). This theory is decidable, see e.g. [12, Corollary 7.2.11] and it can be shown in TOWER as it can be reduced to the satisfiability to the monadic second-order theory of the infinite binary tree.

Let us define the translation $T(\phi)$ of the formula ϕ from $\mathrm{MSL}^f(*, \Diamond, \langle \neq \rangle)$ built over the propositional variables $p_1, \ldots, p_n$ (internalisation of the semantics):

$$\begin{gathered} \exists\ \mathbf{P}_{dom}, \mathbf{P}, P_1, \ldots, P_n\ \exists\ \mathtt{x}\ \textstyle\bigwedge_i P_i \subseteq \mathbf{P}_{dom} \wedge (\mathbf{P} \subseteq \mathbf{P}_{dom}) \wedge \\ (\forall\ \mathtt{z}\ \mathbf{P}(\mathtt{z}) \Rightarrow \exists\ \mathtt{z}'\ (\mathtt{z}' = f(\mathtt{z})) \wedge \mathbf{P}_{dom}(\mathtt{z}')) \wedge \mathbf{P}_{dom}(\mathtt{x}) \wedge t(\mathtt{x}, \phi, \mathbf{P}), \end{gathered}$$

where the map t is recursively defined as follows.

- $t(\mathtt{x}, p_i, \mathbf{P}) \stackrel{\text{def}}{=} P_i(\mathtt{x})$,
- $t(\mathtt{x}, \mathtt{emp}, \mathbf{P}) \stackrel{\text{def}}{=} \forall\ \mathtt{y}\ \neg \mathbf{P}(\mathtt{y})$,
- t is homomorphic for Boolean connectives,
- $t(\mathtt{x}, \Diamond\psi, \mathbf{P}) \stackrel{\text{def}}{=} \mathbf{P}(\mathtt{x}) \wedge \exists\ \mathtt{y}\ (\mathtt{y} = f(\mathtt{x})) \wedge t(\mathtt{y}, \psi, \mathbf{P})$,
- $t(\mathtt{x}, \langle\neq\rangle\psi, \mathbf{P}) \stackrel{\text{def}}{=} \exists\ \mathtt{y}\ (\mathtt{y} \neq \mathtt{x}) \wedge \mathbf{P}_{dom}(\mathtt{y}) \wedge t(\mathtt{y}, \psi, \mathbf{P})$,
- $t(\mathtt{x}, \phi_1 * \phi_2, \mathbf{P}) \stackrel{\text{def}}{=} \exists\ \mathbf{Q}, \mathbf{Q}'\ (\mathbf{P} = \mathbf{Q} \uplus \mathbf{Q}') \wedge t(\mathtt{x}, \phi_1, \mathbf{Q}) \wedge t(\mathtt{x}, \phi_2, \mathbf{Q}')$ where $\mathbf{P} = \mathbf{Q} \uplus \mathbf{Q}'$ is an abbreviation for
$$\forall\ \mathtt{z}\ (\mathbf{P}(\mathtt{z}) \Leftrightarrow (\mathbf{Q}(\mathtt{z}) \vee \mathbf{Q}'(\mathtt{z}))) \wedge \neg(\mathbf{Q}(\mathtt{z}) \wedge \mathbf{Q}'(\mathtt{z})).$$

'$P \subseteq Q$' is also an abbreviation for '$\forall\ \mathtt{x}\ P(\mathtt{x}) \Rightarrow Q(\mathtt{x})$'.

So, the map t combines the standard translation from modal logic to first-order logic (see e.g. [38, 56, 66, 55]) and the translation from separation logic into second-order logic (see e.g. [15]). One can show that ϕ is satisfiable in $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ iff $T(\phi)$ is satisfiable in the weak monadic second-order theory of one unary function; the proof is rather standard as the translation simply internalises the semantics for $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$. For instance, the prefix $\exists\ \mathbf{P}_{dom}, \mathbf{P}\ \exists\ P_1, \ldots, P_n$ with quantifications over monadic predicates, specifies what is the domain (thanks to $\mathbf{P}_{dom}$) and what is the functional accessibility relation thanks to $\mathbf{P}$.

Let us be a bit more precise. First, suppose that $T(\phi)$ is satisfiable. So there is a model $\mathfrak{M} = \langle D, \mathfrak{f}, = \rangle$ (D is a countable set and $\mathfrak{f}$ is a map $D \to D$) such that $\mathfrak{M} \models T(\phi)$ ($\models$ is overloaded here and denotes also the satisfaction relation for closed formulae from the weak monadic second-order theory, weakness meaning that the monadic predicates are interpreted by finite subsets of D). So, there is a valuation ρ (providing an interpretation for $\mathbf{P}_{dom}$, $\mathbf{P}$, P_1, ..., P_n and $\mathtt{x}$) such that
$$\begin{array}{ll}\mathfrak{M} \models_\rho & \bigwedge_i P_i \subseteq \mathbf{P}_{dom} \wedge (\mathbf{P} \subseteq \mathbf{P}_{dom}) \\ & \wedge(\forall\ \mathtt{z}\ \mathbf{P}(\mathtt{z}) \Rightarrow \exists\ \mathtt{z}'\ (\mathtt{z}' = f(\mathtt{z})) \wedge \mathbf{P}_{dom}(\mathtt{z}')) \wedge \mathbf{P}_{dom}(\mathtt{x}) \wedge t(\mathtt{x}, \phi, \mathbf{P}).\end{array}$$

Let us build a finite and functional model $\mathfrak{M}' = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ as follows:

- $\mathfrak{W} \stackrel{\text{def}}{=} \rho(\mathbf{P}_{dom})$ ($\mathfrak{W}$ is finite as the monadic predicates are interpreted weakly),
- $\mathfrak{R} \stackrel{\text{def}}{=} \{(\mathfrak{l}, \mathfrak{f}(\mathfrak{l}))\ \mid \mathfrak{l} \in \rho(\mathbf{P})\}$. Functionality of $\mathfrak{R}$ is immediate since $\mathfrak{f}$ is a map, and finiteness of $\mathfrak{R}$ is again due to the weak interpretation of the monadic predicate $\mathbf{P}$. Moreover, we have $\mathfrak{R} \subseteq \mathfrak{W} \times \mathfrak{W}$ as a consequence of $\mathfrak{M} \models_\rho \mathbf{P} \subseteq \mathbf{P}_{dom} \wedge (\forall\ \mathtt{z}\ \mathbf{P}(\mathtt{z}) \Rightarrow \exists\ \mathtt{z}'\ (\mathtt{z}' = f(\mathtt{z})) \wedge \mathbf{P}_{dom}(\mathtt{z}'))$.
- For all $i \in [1, n]$ and $\mathfrak{l} \in \mathfrak{W}$, we have $\mathfrak{l} \in \mathfrak{V}(p_i) \stackrel{\text{def}}{\Leftrightarrow} \mathfrak{l} \in \rho(P_i)$.

By structural induction, one can show that for all $\mathfrak{l} \in \mathfrak{W}$, for all subformulae ψ of ϕ, we have $\mathfrak{M} \models_{\rho[\mathtt{y} \mapsto \mathfrak{l}]} t(\mathtt{y}, \psi, \mathbf{P})$ iff $\mathfrak{M}', \mathfrak{l} \models \psi$. This is the place where one checks that the translation t simply internalises the semantics of $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ and the proof is very similar to the one for the standard translation from modal logic to first-order logic. Details are omitted here. Consequently, $\mathfrak{M}', \rho(\mathtt{x}) \models \phi$ and therefore ϕ is $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ satisfiable.

Conversely, suppose that ϕ is $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ satisfiable. So, there is a model $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ and $\mathfrak{l} \in \mathfrak{W}$ such that $\mathfrak{M}, \mathfrak{l} \models \phi$. As above, let us build a model $\mathfrak{M}' = \langle D, \mathfrak{f}, = \rangle$ for the weak monadic second-order theory and a valuation ρ as follows:

- $D \stackrel{\text{def}}{=} \mathfrak{W}$,
- For all $\mathfrak{l}' \in \mathfrak{W}$ such that $\mathfrak{R}(\mathfrak{l}') = \{\mathfrak{l}''\}$, $\mathfrak{f}(\mathfrak{l}') \stackrel{\text{def}}{=} \mathfrak{l}''$. If $\mathfrak{R}(\mathfrak{l}') = \emptyset$, then by default $\mathfrak{f}(\mathfrak{l}') \stackrel{\text{def}}{=} \mathfrak{l}$. So, $\mathfrak{f}$ is clearly a map $D \to D$.
- For all $i \in [1, n]$ and $\mathfrak{l} \in D$, $\mathfrak{l} \in \rho(P_i) \stackrel{\text{def}}{\Leftrightarrow} \mathfrak{l} \in \mathfrak{V}(p_i)$.
- $\rho(\mathtt{x}) \stackrel{\text{def}}{=} \mathfrak{l}$; $\rho(\mathbf{P}) \stackrel{\text{def}}{=} \{\mathfrak{l}'\ \mid\ \mathfrak{R}(\mathfrak{l}') \neq \emptyset\}$; $\rho(\mathbf{P}_{dom}) \stackrel{\text{def}}{=} \mathfrak{W}$.

It is quite easy to show that

$$\begin{array}{ll}\mathfrak{M}' \models_\rho & \bigwedge_i P_i \subseteq \mathbf{P}_{dom} \wedge (\mathbf{P} \subseteq \mathbf{P}_{dom}) \\ & \wedge(\forall\ \mathsf{z}\ \mathbf{P}(\mathsf{z}) \Rightarrow \exists\ \mathsf{z}'\ (\mathsf{z}' = f(\mathsf{z})) \wedge \mathbf{P}_{dom}(\mathsf{z}')) \wedge \mathbf{P}_{dom}(\mathtt{x}).\end{array}$$

Again, by structural induction, one can show that for all $\mathfrak{l}' \in D$, for all subformulae ψ of ϕ, we have $\mathfrak{M}' \models_{\rho[\mathsf{y} \mapsto \mathfrak{l}']} t(\mathsf{y}, \psi, \mathbf{P})$ iff $\mathfrak{M}, \mathfrak{l}' \models \psi$. As a consequence, $\mathfrak{M}' \models T(\phi)$ and therefore $T(\phi)$ is satisfiable. □

Another consequence of the previous translation and from Lemma 1 is the following:

Corollary 5. *The model checking problem for* $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle)$ *is* PSPACE*-complete.*

3.2 Adding the converse modality

In this section, we consider the standard converse modality $\Diamond^{-1}$ (not originally in MSL), when it interacts with separating connectives. More precisely,

$$\mathfrak{M}, \mathfrak{l} \models \Diamond^{-1}\phi \overset{\text{def}}{\Leftrightarrow} \mathfrak{M}, \mathfrak{l}' \models \phi, \text{ for some } \mathfrak{l}' \in \mathbb{N} \text{ such that } (\mathfrak{l}', \mathfrak{l}) \in \mathfrak{R}.$$

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

Proof. For the PSPACE-hardness of the model-checking for $\mathrm{MSL}^f(*, \Diamond^{-1})$, the proof is similar to the proof of Lemma 1, which is itself reminiscent to the PSPACE-hardness proof for the model-checking problem for relation-changing modal logics (see e.g [3]) or to the PSPACE-hardness of the satisfiability problem for quantifier-free separation logic restricted to the separating conjunction $*$, see e.g. [19]. Let $\varphi = \mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$ be a QBF formula in prenex normal form. We consider the finite and functional model $\mathfrak{M}_n = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ from the proof of Lemma 1 and we will be able to establish that φ is satisfiable iff $\mathfrak{M}_n, 0 \models t(\varphi)$ where $t(\cdot)$ is recursively defined below. This time, the truth of the propositional variable p_i in QBF subformulae is encoded by the satisfaction of the formula $\Diamond^{-1}(p_i \wedge p_\top)$. Let us define the map t as follows when t is homomorphic for Boolean connectives:

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

As in the proof of Lemma 1, for every $j \in [1, n+1]$, we write ϕ_j to denote $\mathcal{Q}_j\ p_j\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$. Given a model $\mathfrak{M} \subseteq \mathfrak{M}_n$ and a propositional valuation v, we write $\mathfrak{M} \approx_j v$ as in the proof of Lemma 1. Again, 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 $v \models \phi_j$, which allows to conclude that φ is satisfiable iff $\mathfrak{M}_n, 0 \models t(\varphi)$.

Now, let us show that the satisfiability problem for $\mathrm{MSL}(*, \Diamond^{-1})$ is PSPACE-hard. Let $\varphi = \mathcal{Q}_1\ p_1\ \cdots\ \mathcal{Q}_n\ p_n\ \phi$ be a QBF formula in prenex normal form. Again, we consider the finite and functional model $\mathfrak{M}_n = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ from the proof of Lemma 1. First, we define a formula Val_n such that for all MSL models $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}' \rangle$ and $\mathfrak{l} \in \mathbb{N}$, we have $\mathfrak{M}, \mathfrak{l} \models \mathsf{Val}_n$ iff there is $X \subseteq \mathbb{N}$ with $\mathsf{card}(X) = 2n + 1$ and $\mathfrak{l} \in X$ such that $\langle X, \mathfrak{R}', \mathfrak{l} \rangle$ and $\langle \mathfrak{W}, \mathfrak{R}, 0 \rangle$ are isomorphic structures. The formula Val_n is defined below:

$$\mathsf{Val}_n \overset{\text{def}}{=} \mathop{\Large\ast}_{1 \leq i \leq n} ((\Diamond^{-1}(p_i \wedge p_\top) \wedge \mathtt{size} = 1) * (\Diamond^{-1}(p_i \wedge \neg p_\top) \wedge \mathtt{size} = 1).$$

It is easy to show that φ is QBF satisfiable if and only if $\mathsf{Val}_n \wedge t(\varphi)$ is $\mathrm{MSL}(*, \Diamond^{-1})$ satisfiable. □

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 for $\mathrm{MSL}(*, \Diamond^{-1})$ is quite expected but our proof is rather simple.

By using the proof technique from the proof of Theorem 4, we can establish the result below where $\Diamond^{-1}$ is part of the modal operators.

Theorem 7. *The satisfiability problem for* $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle, \Diamond^{-1})$ *is in* TOWER*.*

Proof. Let us extend the definition of the map t from the proof of Theorem 4 with $t(\mathtt{x}, \Diamond^{-1}\psi, \mathbf{P}) \overset{\text{def}}{=} \exists\, \mathtt{y}\ (\mathbf{P}(\mathtt{y}) \wedge \mathtt{x} = f(\mathtt{y})) \wedge t(\mathtt{y}, \psi, \mathbf{P})$. One can prove that t is a logspace reduction from the satisfiability problem for $\mathrm{MSL}^f(*, \Diamond, \Diamond^{-1}, \langle\neq\rangle)$ into the satisfiability problem for the weak monadic second-order theory of one unary function. The proof of Lemma 3 can be adapted when $\Diamond^{-1}$ is added, which leads to the TOWER upper bound. □

By using the result mentioned above plus Theorem 6 we get:

Corollary 8. *The model checking problem for* $\mathrm{MSL}^f(*, \Diamond, \langle\neq\rangle, \Diamond^{-1})$ *is* PSPACE*-complete.*

As a conclusion, today, there is a huge gap for $\mathrm{MSL}(*, \Diamond^{-1})$ between the PSPACE-hardness for the satisfiability problem and the TOWER upper bound. An attempt to translate $\mathrm{MSL}(*, \Diamond^{-1})$ into a limited fragment a quantified CTL thanks to the relationships between separation and second-order quantification, happens to be not so promising in view of [7].

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.

In order to illustrate the specificity of the expressive power of the modal separation logic $\mathrm{MSL}(*, \Diamond)$, let us consider the formula below such that when interpreted on a location $\mathfrak{l}$, states that the model contains exactly a loop of length 2 visiting $\mathfrak{l}$:

$$\mathtt{size} = 2 \wedge \Diamond\Diamond\Diamond\top \wedge \neg(\neg\mathtt{emp} * \Diamond\Diamond\Diamond\top) \wedge \neg\Diamond(\neg\mathtt{emp} * \Diamond\Diamond\Diamond\top)$$

Obviously, such a property cannot be expressed in the modal logic Alt_1 or in the quantifier-free separation logic $\mathrm{SL}(*, -\!\!*)$. A more thorough analysis on the expressivity capabilities of the fragments $\mathrm{MSL}(*, \Diamond)$ and $\mathrm{MSL}(*, \langle\neq\rangle)$ has been perfomed in [34] and this complements nicely what is presented below.

4.1 The minimal modal separation logic $\mathrm{MSL}(*, \Diamond)$: linear-size model property

To show that $\mathrm{MSL}(*, \Diamond)$ has a linear-size model property (i.e., the cardinal of the relation can be linearly 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. Then, 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 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 $\mathrm{MSL}(*, \Diamond)$, see e.g. Lemma 11.

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.

- $\mathfrak{W}_{\mathfrak{l},s} \overset{\text{def}}{=} \{(i,\mathfrak{l}_i) \mid i \in [0,s],\ \exists\ \mathfrak{l}_0,\ldots,\mathfrak{l}_i,\ \mathfrak{l} = \mathfrak{l}_0\mathfrak{R}\mathfrak{l}_1 \cdots \mathfrak{R}\mathfrak{l}_{i-1}\mathfrak{R}\mathfrak{l}_i\}$. We also write $t_{\mathfrak{l}} = \max\{i \mid (i,\mathfrak{l}_i) \in \mathfrak{W}_{\mathfrak{l},s}\}$ (so $t_{\mathfrak{l}} \leq s$).
- $\mathfrak{R}_{\mathfrak{l},s} \overset{\text{def}}{=} \{(\mathfrak{l}_i,\mathfrak{l}_{i+1}) \mid i \in [0,t_{\mathfrak{l}}-1] \text{ and } (i,\mathfrak{l}_i),(i+1,\mathfrak{l}_{i+1}) \in \mathfrak{W}_{\mathfrak{l},s}\}$. We also write $s^\star_{\mathfrak{l}} \overset{\text{def}}{=} \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s})$ 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 $\overset{s,n}{\sim}$ between pointed models: $\mathfrak{M},\mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}',\mathfrak{l}' \overset{\text{def}}{\Leftrightarrow}$ the conditions below are satisfied:

- We have $t_{\mathfrak{l}} = t_{\mathfrak{l}'} (\overset{\text{def}}{=} t)$. 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 $\{p_1,\ldots,p_n\} \subset \mathrm{PROP}$.
- For all $i,j \in [0,t-1]$, we have $\mathfrak{l}_i = \mathfrak{l}_j$ iff $\mathfrak{l}'_i = \mathfrak{l}'_j$. Hence, $s^\star_{\mathfrak{l}} = s^\star_{\mathfrak{l}'}$ $(\overset{\text{def}}{=} s^\star)$.
- We have $rem^\star_{\mathfrak{l}} = rem^\star_{\mathfrak{l}'}$ $(\overset{\text{def}}{=} rem^\star)$.

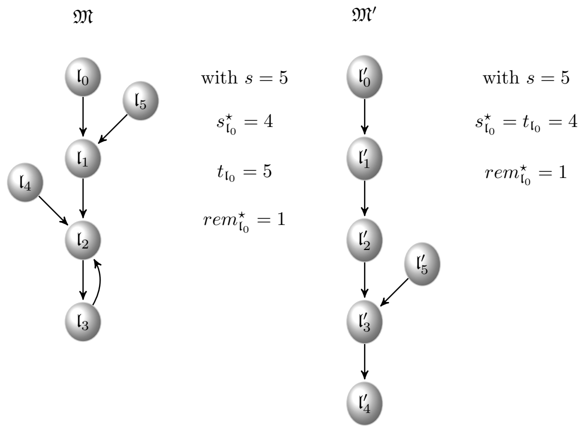


Figure 2: Decomposition.

The binary relation $\overset{s,n}{\sim}$ can be easily shown to be an equivalence relation. So, $\mathfrak{R}_{\mathfrak{l},s}$ and $\mathfrak{R}'_{\mathfrak{l}',s'}$ can be understood as isomorphic structures when $\mathfrak{M},\mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}',\mathfrak{l}'$. In Figure 2, $\mathfrak{M},\mathfrak{l}_0 \overset{4,n}{\sim} \mathfrak{M}',\mathfrak{l}'_0$ (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 9 below is essential to justify that the indistinguishability relation $\overset{s,n}{\sim}$ behaves properly with disjoint unions of models. Its proof is tedious as numerous cases are needed.

Lemma 9. *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 $\mathfrak{M}_2,\mathfrak{l} \overset{s_2,n}{\sim} \mathfrak{M}'_2,\mathfrak{l}'$.*

Proof. Let $\mathfrak{M} = \langle \mathbb{N},\mathfrak{R},\mathfrak{V}\rangle$, $\mathfrak{M}_1 = \langle \mathbb{N},\mathfrak{R}_1,\mathfrak{V}\rangle$, $\mathfrak{M}_2 = \langle \mathbb{N},\mathfrak{R}_2,\mathfrak{V}\rangle$ with $\mathfrak{W}_{\mathfrak{l},s} = \{(0,\mathfrak{l}_0),\ldots,(t,\mathfrak{l}_t)\}$ and $\mathfrak{W}'_{\mathfrak{l}',s} = \{(0,\mathfrak{l}'_0),\ldots,(t,\mathfrak{l}'_t)\}$ ($\mathfrak{l}_0 = \mathfrak{l}$ and $\mathfrak{l}'_0 = \mathfrak{l}'$). Moreover, we use the values $rem^\star_{1,\mathfrak{l}}$, $rem^\star_{2,\mathfrak{l}}$ based on the decomposition for $\mathfrak{M}_1$ and $\mathfrak{M}_2$ with s_1 and s_2, respectively. When $\mathfrak{R}(\mathfrak{l}) = \mathfrak{R}'(\mathfrak{l}') = \emptyset$, the proof is by an easy verification. Below, we assume that $\mathfrak{R}(\mathfrak{l}) \neq \emptyset$.

Let $\alpha \in [1,s]$ be such that $\mathfrak{R}_2(\mathfrak{l}_0) = \emptyset$, $\mathfrak{l}_0\mathfrak{R}_1\mathfrak{l}_1 \cdots \mathfrak{l}_{\alpha-1}\mathfrak{R}_1\mathfrak{l}_\alpha$ and for no $\alpha' \in [\alpha+1,s]$, we have $\mathfrak{l}_0\mathfrak{R}_1\mathfrak{l}_1 \cdots \mathfrak{l}_{\alpha'-1}\mathfrak{R}_1\mathfrak{l}_{\alpha'}$ (so α is optimal in some sense). Note that necessarily $\alpha \leq t$. We also define $\alpha^* = \mathsf{card}(\{(\mathfrak{l}_j,\mathfrak{l}_{j+1}) \mid j \in [0,\alpha-1]\})$, and therefore $\alpha^* \leq \alpha$. The symmetrical case with $\mathfrak{R}_1(\mathfrak{l}_0) = \emptyset$ and $\mathfrak{l}_0\mathfrak{R}_2\mathfrak{l}_1 \cdots \mathfrak{l}_{\alpha-1}\mathfrak{R}_2\mathfrak{l}_\alpha$, admits a similar treatment and it is omitted below. We perform a case analysis depending whether $\mathfrak{R}_{\mathfrak{l},s}$ contains s elements.

Case $\mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) = s$**.** So, $\alpha^* = \alpha$. By definition, this means that $s^\star = s$. As $\mathfrak{M},\mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}',\mathfrak{l}'$, we have $\mathsf{card}(\mathfrak{R}'_{\mathfrak{l}',s}) = s$ and $rem^\star_{\mathfrak{l}} = rem^\star_{\mathfrak{l}'} = 0$.

Subcase $\alpha \geq s_1$**.** Hence, $rem^\star_{1,\mathfrak{l}} = 0$. The relation $\mathfrak{R}'_2$ is defined as

$$\{(\mathfrak{l}'_j, \mathfrak{l}'_{j+1}) \mid j \in [s - rem^\star_{2,\mathfrak{l}}, s-1]\}$$

and $\mathfrak{R}'_1 = \mathfrak{R}' \setminus \mathfrak{R}'_2$. As $rem^\star_{2,\mathfrak{l}'} = rem^\star_{2,\mathfrak{l}} \leq s_2$, $\mathfrak{R}'_2(\mathfrak{l}) = \mathfrak{R}_2(\mathfrak{l}) = \emptyset$ and $\mathsf{card}(\mathfrak{R}_{1,\mathfrak{l},s_1}) = \mathsf{card}(\mathfrak{R}'_{1,\mathfrak{l}',s_1}) = s_1$ $(s_1 + rem^\star_{2,\mathfrak{l}'} \leq s)$, we have $\mathfrak{M}_1,\mathfrak{l} \overset{s_1,n}{\sim} \mathfrak{M}'_1,\mathfrak{l}'$ and $\mathfrak{M}_2,\mathfrak{l} \overset{s_2,n}{\sim} \mathfrak{M}'_2,\mathfrak{l}'$.

Subcase $\alpha < s_1$**.** In the case $rem^\star_{2,\mathfrak{l}} < s_2$, we have $\alpha + rem^\star_{1,\mathfrak{l}} = s_1$ as $\mathfrak{R}$ has at least $s = s_1 + s_2$ edges. As $\{(\mathfrak{l}'_\alpha, \mathfrak{l}'_{\alpha+1}), \ldots, (\mathfrak{l}'_{s-1}, \mathfrak{l}'_s)\} \subseteq \mathfrak{R}'$ and $(\mathfrak{l}_\alpha, \mathfrak{l}_{\alpha+1}) \in \mathfrak{R}_2$ (because $\alpha < s_1$ and $s_1 < s$), $\mathfrak{R}'_2$ is defined as

$$\{(\mathfrak{l}'_j, \mathfrak{l}'_{j+1}) \mid j \in [\alpha, \alpha + rem^\star_{2,\mathfrak{l}} - 1]\}$$

and $\mathfrak{R}'_1 = \mathfrak{R}' \setminus \mathfrak{R}'_2$. We have $rem^\star_{2,\mathfrak{l}'} = rem^\star_{2,\mathfrak{l}}$, $\mathfrak{R}'_2(\mathfrak{l}) = \mathfrak{R}_2(\mathfrak{l}) = \emptyset$, $\mathsf{card}(\mathfrak{R}_{1,\mathfrak{l},s_1}) = \mathsf{card}(\mathfrak{R}'_{1,\mathfrak{l}',s_1}) = \alpha$ and $rem^\star_{1,\mathfrak{l}} = rem^\star_{1,\mathfrak{l}'}$. As $rem^\star_{2,\mathfrak{l}} < s_2$ and $\alpha + rem^\star_{1,\mathfrak{l}} = s_1$, we get $\mathsf{card}(\mathfrak{R}' \setminus (\mathfrak{R}'_2 \cup \{(\mathfrak{l}'_j, \mathfrak{l}'_{j+1}) \mid j \in [1, \alpha - 1]\})) \geq rem_{1,\mathfrak{l}}$. So, we have $\mathfrak{M}_1,\mathfrak{l} \overset{s_1,n}{\sim} \mathfrak{M}'_1,\mathfrak{l}'$ and $\mathfrak{M}_2,\mathfrak{l} \overset{s_2,n}{\sim} \mathfrak{M}'_2,\mathfrak{l}'$.

In the case $rem^\star_{2,\mathfrak{l}} = s_2$, $\mathfrak{R}'_1$ is defined as the set below:

$$\{(\mathfrak{l}'_0, \mathfrak{l}'_1), \ldots, (\mathfrak{l}'_{\alpha-1}, \mathfrak{l}'_\alpha)\} \cup \{(\mathfrak{l}'_{\alpha+1}, \mathfrak{l}'_{\alpha+2}), \ldots, (\mathfrak{l}'_{\alpha+\beta}, \mathfrak{l}'_{\alpha+\beta+1})\}$$

with $\beta = rem^\star_{1,\mathfrak{l}}$ and $\mathfrak{R}'_2 = \mathfrak{R}' \setminus \mathfrak{R}'_1$.

As $\alpha + \beta \leq s_1$ and $s_1 + s_2 = s$, $\mathsf{card}(\{(\mathfrak{l}'_0, \mathfrak{l}'_1), \ldots, (\mathfrak{l}'_{s-1}, \mathfrak{l}'_s)\} \setminus \mathfrak{R}'_1) \geq s_2$ and therefore $rem^\star_{2,\mathfrak{l}'} = s_2$. So we have $\mathfrak{M}_1,\mathfrak{l} \overset{s_1,n}{\sim} \mathfrak{M}'_1,\mathfrak{l}'$ and $\mathfrak{M}_2,\mathfrak{l} \overset{s_2,n}{\sim} \mathfrak{M}'_2,\mathfrak{l}'$.

Case $\mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) = s^\star < s$**.** Let $S = \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) + rem^\star_{\mathfrak{l}}$ (also equal to $\mathsf{card}(\mathfrak{R}'_{\mathfrak{l}',s}) + rem^\star_{\mathfrak{l}'}$ and $S \leq s$) and $T = rem^\star_{\mathfrak{l}}$. In order to represent the T edges in $\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}$ (resp. in $\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s}$), we pick the edges $(\mathfrak{n}_1, \mathfrak{m}_1), \cdots, (\mathfrak{n}_T, \mathfrak{m}_T) \in \mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}$ and the edges $(\mathfrak{n}'_1, \mathfrak{m}'_1), \cdots, (\mathfrak{n}'_T, \mathfrak{m}'_T) \in \mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s}$. Again, we proceed by considering several cases.

Subcase $\alpha^* \geq s_1$**.** So $rem^\star_{1,\mathfrak{l}} = 0$. First observe that $s_1 + rem^\star_{2,\mathfrak{l}} \leq s^\star + rem^\star_{\mathfrak{l}} = S$. Indeed, if $S = s$, then as $rem^\star_{2,\mathfrak{l}} \leq s_2$ and $s = s_1 + s_2$, we get $s_1 + rem^\star_{2,\mathfrak{l}} \leq s_1 + s_2 = s = S$. Otherwise, $S < s$ and therefore $\mathsf{card}(\mathfrak{R}) = \mathsf{card}(\mathfrak{R}') = S$. So $rem^\star_{2,\mathfrak{l}} \leq S - \alpha^*$ and as $s_1 \leq \alpha^*$, we get $s_1 + rem^\star_{2,\mathfrak{l}} \leq S$. Consequently, $rem^\star_{2,\mathfrak{l}} \leq (s^\star - s_1) + rem^\star_{\mathfrak{l}}$. The relation $\mathfrak{R}'_2$ is defined as a set with $rem^\star_{2,\mathfrak{l}}$ edges among

$$\{(\mathfrak{l}'_{s_1}, \mathfrak{l}'_{s_1+1}), \ldots, (\mathfrak{l}'_{s^\star-1}, \mathfrak{l}'_{s^\star})\} \cup \{(\mathfrak{n}'_1, \mathfrak{m}'_1), \ldots, (\mathfrak{n}'_T, \mathfrak{m}'_T)\}.$$

As $rem^\star_{2,\mathfrak{l}} \leq (s^\star - s_1) + rem^\star_{\mathfrak{l}} = (s^\star - s_1) + T$, $\mathfrak{R}'_2$ can be properly populated and $\mathfrak{R}'_1 = \mathfrak{R}' \setminus \mathfrak{R}'_2$. It remains to check that the decomposition meets the required properties. But we have $rem^\star_{2,\mathfrak{l}} = rem^\star_{2,\mathfrak{l}'}$, $s^\star_{1,\mathfrak{l}'} = s_1$ and $rem^\star_{1,\mathfrak{l}'} = 0$.

Subcase $\alpha^* < s_1$**.** Again, we distinguish several cases.

Subcase $S < s$**.** So, $\mathsf{card}(\mathfrak{R}) = \mathsf{card}(\mathfrak{R}') = S$. If $rem^\star_{1,\mathfrak{l}} + \alpha^* < s_1$, then $\mathfrak{R}'_1$ is equal to $\{(\mathfrak{l}'_0, \mathfrak{l}'_1), \ldots, (\mathfrak{l}'_{\alpha-1}, \mathfrak{l}'_\alpha)\}$ plus $rem^\star_{1,\mathfrak{l}}$ disjoint edges from $\mathfrak{R}'$ (that exist as $\alpha^* + rem^\star_{1,\mathfrak{l}} + rem^\star_{2,\mathfrak{l}} \leq S$ and $\alpha^* = \mathsf{card}(\{(\mathfrak{l}'_0, \mathfrak{l}'_1), \ldots, (\mathfrak{l}'_{\alpha-1}, \mathfrak{l}'_\alpha)\})$). We have $\mathfrak{R}'_2 = \mathfrak{R}' \setminus \mathfrak{R}'_1$. Otherwise (i.e. $rem^\star_{1,\mathfrak{l}} + \alpha^* = s_1$), if $rem^\star_{2,\mathfrak{l}} < s_2$, then $\mathfrak{R}'_2$ is defined as a set with $rem^\star_{2,\mathfrak{l}}$ edges among

$$\{(\mathfrak{l}'_\alpha, \mathfrak{l}'_{\alpha+1}), \ldots, (\mathfrak{l}'_{s^\star-1}, \mathfrak{l}'_{s^\star})\} \cup \{(\mathfrak{n}'_1, \mathfrak{m}'_1), \ldots, (\mathfrak{n}'_T, \mathfrak{m}'_T)\}$$

and $\mathfrak{R}'_1 = \mathfrak{R}' \setminus \mathfrak{R}'_2$. The case $rem^\star_{1,\mathfrak{l}} + \alpha^* = s_1$ and $rem^\star_{2,\mathfrak{l}} = s_2$ does not happen as $S < s$ and $s = s_1 + s_2$.

Subcase $S = s$. If $rem^{\star}_{1,\mathfrak{l}} + \alpha^{*} < s_1$, then $\mathfrak{R}'_1$ is equal to the set $\{(\mathfrak{l}'_0, \mathfrak{l}'_1), \ldots, (\mathfrak{l}'_{\alpha-1}, \mathfrak{l}'_{\alpha})\}$ plus $rem^{\star}_{1,\mathfrak{l}}$ disjoint edges from $\mathfrak{R}'$ excluding $(\mathfrak{l}'_{\alpha}, \mathfrak{l}'_{\alpha+1})$ if it exists (that exist as $\alpha^{*} + rem^{\star}_{1,\mathfrak{l}} + rem^{\star}_{2,\mathfrak{l}} \leq s_1 + s_2 = s = S$). We have $\mathfrak{R}'_2 = \mathfrak{R}' \setminus \mathfrak{R}'_1$. Otherwise (i.e. $rem^{\star}_{1,\mathfrak{l}} + \alpha^{*} = s_1$), if $rem^{\star}_{2,\mathfrak{l}} < s_2$, then $\mathfrak{R}'_2$ is defined as a set with $rem^{\star}_{2,\mathfrak{l}}$ edges among

$$\{(\mathfrak{l}'_{\alpha}, \mathfrak{l}'_{\alpha+1}), \ldots, (\mathfrak{l}'_{s^{\star}-1}, \mathfrak{l}'_{s^{\star}})\} \cup \{(\mathfrak{n}'_1, \mathfrak{m}'_1), \ldots, (\mathfrak{n}'_T, \mathfrak{m}'_T)\}$$

and $\mathfrak{R}'_1 = \mathfrak{R}' \setminus \mathfrak{R}'_2$. It remains to deal with the case $rem^{\star}_{1,\mathfrak{l}} + \alpha^{*} = s_1$ and $rem^{\star}_{2,\mathfrak{l}} = s_2$. So, $rem^{\star}_{1,\mathfrak{l}} + rem^{\star}_{2,\mathfrak{l}} = s - \alpha^{*}$ and $(s^{\star} - \alpha^{*}) + T = s - \alpha^{*}$ and therefore $rem^{\star}_{1,\mathfrak{l}} + rem^{\star}_{2,\mathfrak{l}} = (s^{\star} - \alpha^{*}) + T$ (as $s = s^{\star} + T$). It is therefore possible to define a partition $\{\mathfrak{R}'^{r}_1, \mathfrak{R}'_2\}$ of the set below

$$\{(\mathfrak{l}'_{\alpha}, \mathfrak{l}'_{\alpha+1}), \ldots, (\mathfrak{l}'_{s^{\star}-1}, \mathfrak{l}'_{s^{\star}})\} \cup \{(\mathfrak{n}'_1, \mathfrak{m}'_1), \ldots, (\mathfrak{n}'_T, \mathfrak{m}'_T)\}$$

so that $\mathsf{card}(\mathfrak{R}'^{r}_1) = rem^{\star}_{1,\mathfrak{l}}$, $\mathsf{card}(\mathfrak{R}'_2) = rem^{\star}_{2,\mathfrak{l}}$ and $(\mathfrak{l}'_{\alpha}, \mathfrak{l}'_{\alpha+1}) \in \mathfrak{R}'_2$ if $s^{\star} > \alpha^{\star}$. We pose $\mathfrak{R}'_1 = \mathfrak{R}' \setminus \mathfrak{R}'_2$. It is easy to check that

$$\mathfrak{R}'^{r}_1 \cup \{(\mathfrak{l}'_0, \mathfrak{l}'_1), \ldots, (\mathfrak{l}'_{\alpha-1}, \mathfrak{l}'_{\alpha})\} \subseteq \mathfrak{R}'_1.$$

So, $\mathfrak{M}_1, \mathfrak{l} \overset{s_1,n}{\sim} \mathfrak{M}'_1, \mathfrak{l}'$ 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 *memory size* (written msize(ϕ)):

- $\mathrm{msize}(p) \overset{\text{def}}{=} \mathrm{msize}(\mathtt{emp}) \overset{\text{def}}{=} 1$,
- $\mathrm{msize}(\neg\phi) \overset{\text{def}}{=} \mathrm{msize}(\phi)$; $\mathrm{msize}(\phi \wedge \psi) \overset{\text{def}}{=} \max(\mathrm{msize}(\phi), \mathrm{msize}(\psi))$,
- $\mathrm{msize}(\Diamond\phi) \overset{\text{def}}{=} 1 + \mathrm{msize}(\phi)$,
- $\mathrm{msize}(\phi * \psi) \overset{\text{def}}{=} \mathrm{msize}(\phi) + \mathrm{msize}(\psi)$.

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

Let us first present a property of $\overset{s,n}{\sim}$ that is used in the proof of forthcoming Lemma 11.

Lemma 10. *Let $\mathfrak{M}, \mathfrak{M}'$ be models, $\mathfrak{l}, \mathfrak{l}' \in \mathbb{N}$ and $s, n \geq 1$ such that $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$. If there are $\mathfrak{l}_{\dagger}$ and $\mathfrak{l}'_{\dagger}$ such that $(\mathfrak{l}, \mathfrak{l}_{\dagger}) \in \mathfrak{R}$ and $(\mathfrak{l}', \mathfrak{l}'_{\dagger}) \in \mathfrak{R}'$, then $\mathfrak{M}, \mathfrak{l}_{\dagger} \overset{s-1,n}{\sim} \mathfrak{M}', \mathfrak{l}'_{\dagger}$.*

Proof. Suppose $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$ and there are locations $\mathfrak{l}_{\dagger}$ and $\mathfrak{l}'_{\dagger}$ such that $(\mathfrak{l}, \mathfrak{l}_{\dagger}) \in \mathfrak{R}$ and $(\mathfrak{l}', \mathfrak{l}'_{\dagger}) \in \mathfrak{R}'$. In order to show $\mathfrak{M}, \mathfrak{l}_{\dagger} \overset{s-1,n}{\sim} \mathfrak{M}', \mathfrak{l}'_{\dagger}$, we need to verify four conditions. First, we need to check that $t_{\mathfrak{l}_{\dagger}} = t_{\mathfrak{l}'_{\dagger}}$. By hypothesis, we have $\mathfrak{W}_{\mathfrak{l},s} = \{(0, \mathfrak{l}_0), \ldots, (t, \mathfrak{l}_t)\}$ and $\mathfrak{W}'_{\mathfrak{l}',s} = \{(0, \mathfrak{l}'_0), \ldots, (t, \mathfrak{l}'_t)\}$. Then $\mathfrak{l}_{\dagger} = \mathfrak{l}_1$ and $\mathfrak{l}'_{\dagger} = \mathfrak{l}'_1$ (by functionality of $\mathfrak{R}$ and $\mathfrak{R}'$), so $\mathfrak{W}_{\mathfrak{l}_{\dagger},s-1} = \{(0, \mathfrak{l}_1), \ldots, (t-1, \mathfrak{l}_t)\}$ and $\mathfrak{W}'_{\mathfrak{l}'_{\dagger},s-1} = \{(0, \mathfrak{l}'_1), \ldots, (t-1, \mathfrak{l}'_t)\}$). Then $t_{\mathfrak{l}_{\dagger}} = t_{\mathfrak{l}'_{\dagger}}$. The second and third conditions also hold since $\mathfrak{W}_{\mathfrak{l}_{\dagger},s-1}$ and $\mathfrak{W}'_{\mathfrak{l}'_{\dagger},s-1}$ are restrictions of $\mathfrak{W}_{\mathfrak{l},s}$ and $\mathfrak{W}_{\mathfrak{l},s}$, respectively. It only remains to check $rem^{\star}_{\mathfrak{l}_{\dagger}} = rem^{\star}_{\mathfrak{l}'_{\dagger}}$. We have:

$$\begin{aligned}
rem^{\star}_{\mathfrak{l}_{\dagger}} &= \min(s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l}_{\dagger},s-1}), \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l}_{\dagger},s-1})) && \text{(Def. of } rem^{\star}_{\mathfrak{l}_{\dagger}}) \\
&= \min(s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) + 1, \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) + 1) && \text{(Def. of } \mathfrak{R}_{\mathfrak{l}_{\dagger},s-1}) \\
&= \min(s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}), \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s})) + 1 && \text{(Distributivity)}
\end{aligned}$$

Following the same reasoning we can also conclude

$$rem^{\star}_{\mathfrak{l}'_{\dagger}} = \min(s - 1 - \mathsf{card}(\mathfrak{R}'_{\mathfrak{l}',s}), \mathsf{card}(\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s})) + 1,$$

and since $\mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) = \mathsf{card}(\mathfrak{R}'_{\mathfrak{l}',s})$ we get

$$\mathit{rem}^{\star}_{\mathfrak{l}'_{\dagger}} = \min(s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}), \mathsf{card}(\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s})) + 1.$$

Remember also that $\mathit{rem}^{\star}_{\mathfrak{l}} = \mathit{rem}^{\star}_{\mathfrak{l}'} = \min(s - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}), \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}))$. Then we have two cases to consider:

Case $\mathit{rem}^{\star}_{\mathfrak{l}} = \mathit{rem}^{\star}_{\mathfrak{l}'} = s - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) \leq \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s})$**.** Then we have $\mathit{rem}^{\star}_{\mathfrak{l}'} = s - \mathsf{card}(\mathfrak{R}'_{\mathfrak{l}',s}) \leq \mathsf{card}(\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s})$. This implies that $\mathit{rem}^{\star}_{\mathfrak{l}_{\dagger}} = s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s}) + 1 = \mathit{rem}^{\star}_{\mathfrak{l}'_{\dagger}}$.

Case $\mathit{rem}^{\star}_{\mathfrak{l}} = \mathit{rem}^{\star}_{\mathfrak{l}'} = \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) < s - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s})$**.** It can be noticed that $\mathit{rem}^{\star}_{\mathfrak{l}'} = \mathsf{card}(\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s}) = \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s})$. Since $\mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) < s - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s})$, then $\mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) \leq s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s})$, hence $\mathit{rem}^{\star}_{\mathfrak{l}_{\dagger}} = \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) + 1$. Also we have $\mathsf{card}(\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s}) \leq s - 1 - \mathsf{card}(\mathfrak{R}_{\mathfrak{l},s})$, so $\mathit{rem}^{\star}_{\mathfrak{l}'_{\dagger}} = \mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) + 1 = \mathit{rem}^{\star}_{\mathfrak{l}_{\dagger}}$.

So, $\mathfrak{M}, \mathfrak{l}_{\dagger} \overset{s-1,n}{\sim} \mathfrak{M}', \mathfrak{l}'_{\dagger}$. □

This leads us to the following result.

Lemma 11. *Let $s, n \geq 1$. For all formulae ϕ in* $\mathrm{MSL}(*, \Diamond)$ *with* $\mathrm{msize}(\phi) \leq s$ *and built over* $p_1, \ldots, p_n$*, we have* $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$ *implies* $\mathfrak{M}, \mathfrak{l} \models \phi$ *iff* $\mathfrak{M}', \mathfrak{l}' \models \phi$*.*

Proof. The proof is by double induction, on s and then on the structure of ϕ. Let us start by the base case $s = 1$. So ϕ is a Boolean formula built over $\mathtt{emp}$, p_1, $\ldots$, p_n. Suppose $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$.

- Suppose that $\mathfrak{M}, \mathfrak{l} \models p_i$. As $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$, we get $(0, \mathfrak{l}) \in \mathfrak{W}_{\mathfrak{l},s}$, $(0, \mathfrak{l}') \in \mathfrak{W}'_{\mathfrak{l}',s}$ and, $\mathfrak{l}$ and $\mathfrak{l}'$ agree on the propositional variables $p_1, \ldots, p_n$. Consequently, $\mathfrak{M}', \mathfrak{l}' \models p_i$.

- Suppose that $\mathfrak{M}, \mathfrak{l} \models \mathtt{emp}$. So $\mathfrak{R}_{\mathfrak{l},s} = \emptyset$ and $\mathsf{card}(\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}) = 0$ (i.e., $s^{\star}_{\mathfrak{l}} = t_{\mathfrak{l}} = \mathit{rem}^{\star}_{\mathfrak{l}} = 0$). Hence, $\mathfrak{R}'_{\mathfrak{l}',s} = \emptyset$ and $\mathsf{card}(\mathfrak{R}' \setminus \mathfrak{R}'_{\mathfrak{l}',s}) = 0$ too as $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$, which implies that $\mathfrak{M}', \mathfrak{l}' \models \mathtt{emp}$.

- The cases with the Boolean connectives are by an easy verification.

In the induction step, we distinguish different cases depending on the outermost connectives. The induction hypothesis can be stated as follows: for all $s \in [1, k]$, for all formulae ϕ in $\mathrm{MSL}(*, \Diamond)$ with $\mathrm{msize}(\phi) \leq s$ and built over $p_1, \ldots, p_n$, we have $\mathfrak{M}, \mathfrak{l} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'$ implies, $\mathfrak{M}, \mathfrak{l} \models \phi$ iff $\mathfrak{M}', \mathfrak{l}' \models \phi$. Let $s = k + 1$ and let us perform an induction on the structure of ϕ. We omit the base cases (similar to what is done above) as well as the easy cases with Boolean connectives.

- Suppose that $\mathfrak{M}, \mathfrak{l} \models \Diamond\psi$. So, there is $\mathfrak{l}_{\dagger}$ such that $(\mathfrak{l}, \mathfrak{l}_{\dagger}) \in \mathfrak{R}$ and $\mathfrak{M}, \mathfrak{l}_{\dagger} \models \psi$. So $t_{\mathfrak{l}} \geq 1$, $t_{\mathfrak{l}'} \geq 1$ and there is $\mathfrak{l}'_{\dagger}$ such that $(\mathfrak{l}', \mathfrak{l}'_{\dagger}) \in \mathfrak{R}'$. Moreover, by Lemma 10, we have $\mathfrak{M}, \mathfrak{l}_{\dagger} \overset{s-1,n}{\sim} \mathfrak{M}', \mathfrak{l}'_{\dagger}$ (by contrast, $\mathfrak{M}, \mathfrak{l}_{\dagger} \overset{s,n}{\sim} \mathfrak{M}', \mathfrak{l}'_{\dagger}$ does not necessarily hold). As $\mathrm{msize}(\Diamond\psi) \geq 2$, we have $s - 1 \geq 1$. By the induction hypothesis, we get $\mathfrak{M}', \mathfrak{l}'_{\dagger} \models \psi$ (as $\mathrm{msize}(\psi) = \mathrm{msize}(\Diamond\psi) - 1 \leq s - 1$) and therefore $\mathfrak{M}', \mathfrak{l}' \models \Diamond\psi$.

- Suppose that $\mathfrak{M}, \mathfrak{l} \models \psi_1 * \psi_2$. There are $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathfrak{M}_1, \mathfrak{l} \models \psi_1$ and $\mathfrak{M}_2, \mathfrak{l} \models \psi_2$. By definition of $\mathrm{msize}(\cdot)$, there are $s_1, s_2 \geq 1$ such that $s = s_1 + s_2$, $\mathrm{msize}(\psi_1) \leq s_1$ and $\mathrm{msize}(\psi_2) \leq s_2$. By Lemma 9, 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 $\mathfrak{M}_2, \mathfrak{l} \overset{s_2,n}{\sim} \mathfrak{M}'_2, \mathfrak{l}'$. By the induction hypothesis, we get $\mathfrak{M}'_1, \mathfrak{l}' \models \psi_1$ and $\mathfrak{M}'_2, \mathfrak{l}' \models \psi_2$, whence $\mathfrak{M}', \mathfrak{l}' \models \psi_1 * \psi_2$. □

In Figure 3, we present two models $\mathfrak{M}_1$ and $\mathfrak{M}_2$, such that $\mathfrak{M}_1, 0 \overset{3,1}{\sim} \mathfrak{M}_2, 0$ and therefore these two pointed models agree on all formulae of *msize* less or equal to three by Lemma 11 (we have omitted the presentation of $\mathbb{N} \setminus \{0, 1, 2\}$ whose elements do not participate to any edge). However, one can check that not $\mathfrak{M}_1, 0 \overset{4,1}{\sim} \mathfrak{M}_2, 0$ and the distinction can be made with the formula $\psi = \Diamond(\neg\mathtt{emp} * \Diamond\Diamond\Diamond p_1)$ and $\mathrm{msize}(\psi) = 6$.

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

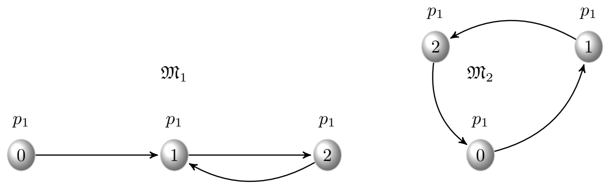


Figure 3: Two MSL$(*, \Diamond)$ models.

Lemma 12. *Let ϕ be a formula in* MSL$(*, \Diamond)$*. ϕ is satisfiable iff ϕ is satisfiable in a finite and functional model with* $\mathsf{card}(\mathfrak{R}) \leq \mathrm{msize}(\phi)$*.*

Proof. The proof is divided in two parts.

1. First, we show that ϕ is satisfiable iff ϕ is MSL$^f(*, \Diamond)$-satisfiable, in a way similar to the proof of Lemma 3.

   So, suppose $\mathfrak{M}, \mathfrak{l} \models \phi$ with $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$. Let us build a finite and functional model $\mathfrak{M}' = \langle \mathfrak{W}', \mathfrak{R}', \mathfrak{V}' \rangle$ such that $\mathfrak{W}' \subseteq \mathbb{N}$ (to be defined below), $\mathfrak{R}' \stackrel{\text{def}}{=} \mathfrak{R}$ and $\mathfrak{V}'$ is the restriction of $\mathfrak{V}$ to $\mathfrak{W}'$. The set $\mathfrak{W}'$ is defined as $\{\mathfrak{n}, \mathfrak{n}' \mid (\mathfrak{n}, \mathfrak{n}') \in \mathfrak{R}\} \cup \{\mathfrak{l}\}$.

   Here is the proof by induction. Actually, we show that given any finite $\mathbb{N} \supset \mathfrak{W}'' \supseteq \mathfrak{W}'$, we have $\mathfrak{M}'' = \langle \mathfrak{W}'', \mathfrak{R}', \mathfrak{V}'' \rangle, \mathfrak{l}' \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$ where $\mathfrak{V}''$ is the restriction of $\mathfrak{V}$ to $\mathfrak{W}''$ and $\mathfrak{l}' \in \mathfrak{W}''$. This slight generalisation is considered in order to handle the separating conjunction $*$. The base cases are by an easy verification as well as the cases in the induction step for the Boolean connectives.

   - Suppose that $\mathfrak{M}, \mathfrak{l}' \models \Diamond\psi$. So, there is $\mathfrak{l}'' \in \mathfrak{R}(\mathfrak{l}')$ such that $\mathfrak{M}, \mathfrak{l}'' \models \psi$. So, $\mathfrak{l}'' \in \mathfrak{W}' \subseteq \mathfrak{W}''$ and by the induction hypothesis $\mathfrak{M}'', \mathfrak{l}'' \models \psi$. As $\mathfrak{R}' = \mathfrak{R}$, we conclude that $\mathfrak{M}'', \mathfrak{l}' \models \Diamond\psi$.

     Conversely, suppose that $\mathfrak{M}'', \mathfrak{l}' \models \Diamond\psi$. So, there is $\mathfrak{l}'' \in \mathfrak{R}'(\mathfrak{l}')$ such that $\mathfrak{M}'', \mathfrak{l}'' \models \psi$. By the induction hypothesis, $\mathfrak{M}, \mathfrak{l}'' \models \psi$. As $\mathfrak{R}' = \mathfrak{R}$, we conclude that $\mathfrak{M}, \mathfrak{l}' \models \Diamond\psi$.
   - Suppose that $\mathfrak{M}, \mathfrak{l}' \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R}$ such that $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, we have $\langle \mathfrak{W}'', \mathfrak{R}_1, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}'', \mathfrak{R}_2, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_2$ and therefore $\mathfrak{M}'', \mathfrak{l}' \models \psi_1 * \psi_2$ (here it is essential to use the generalisation as $\{\mathfrak{n}, \mathfrak{n}' \mid (\mathfrak{n}, \mathfrak{n}') \in \mathfrak{R}_1\} \cup \{\mathfrak{l}'\} \subseteq \mathfrak{W}''$ and $\{\mathfrak{n}, \mathfrak{n}' \mid (\mathfrak{n}, \mathfrak{n}') \in \mathfrak{R}_2\} \cup \{\mathfrak{l}'\} \subseteq \mathfrak{W}''$).

     Conversely, suppose that $\mathfrak{M}'', \mathfrak{l}' \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R} = \mathfrak{R}'$ such that $\langle \mathfrak{W}'', \mathfrak{R}_1, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}'', \mathfrak{R}_2, \mathfrak{V}'' \rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l}' \models \psi_2$, whence $\mathfrak{M}, \mathfrak{l}' \models \psi_1 * \psi_2$.

   Now, suppose that $\mathfrak{M}, \mathfrak{l} \models \phi$ for some finite and functional model $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ with $\mathfrak{W} \subset \mathbb{N}$. Let us build the model $\mathfrak{M}' = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V}' \rangle$ such that $\mathfrak{V}$ is the restriction of $\mathfrak{V}'$ to $\mathfrak{W}$. For all $\mathfrak{l}' \in \mathbb{N} \setminus \mathfrak{W}$ and $i \in [1, n]$, we set $\mathfrak{l}' \in \mathfrak{V}(p_i)$ (arbitrary value). By structural induction, one can show that for all $\mathfrak{l}' \in \mathfrak{W}$, we have $\mathfrak{M}', \mathfrak{l}' \models \phi$ iff $\mathfrak{M}, \mathfrak{l}' \models \phi$ (similar to the second part of the proof of Lemma 3).

2. Now, let us conclude that ϕ is satisfiable iff ϕ is satisfiable in a finite and functional model with $\mathsf{card}(\mathfrak{R}) \leq \mathrm{msize}(\phi)$. Here, we use Lemma 11 in a simple way. Suppose that ϕ is satisfiable, i.e. there is a model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and $\mathfrak{l}$ such that $\mathfrak{M}, \mathfrak{l} \models \phi$. Let $s = \mathrm{msize}(\phi)$. Let us consider $\mathfrak{W}_{\mathfrak{l},s}$, $\mathfrak{R}_{\mathfrak{l},s}$, $rem^{\star}_{\mathfrak{l}} = T$, etc. as defined earlier. So, there are edges $(\mathfrak{n}_1, \mathfrak{m}_1), \cdots, (\mathfrak{n}_T, \mathfrak{m}_T) \in \mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}$. Let $\mathfrak{R}' = \mathfrak{R}_{\mathfrak{l},s} \uplus \{(\mathfrak{n}_1, \mathfrak{m}_1), \cdots, (\mathfrak{n}_T, \mathfrak{m}_T)\}$. We have $\mathfrak{M}, \mathfrak{l} \overset{s;n}{\sim} \mathfrak{M}', \mathfrak{l}$ with $\mathfrak{M}' = \langle \mathbb{N}, \mathfrak{R}', \mathfrak{V} \rangle$ and $\mathsf{card}(\mathfrak{R}') \leq \mathrm{msize}(\phi)$. By Lemma 11, we have $\mathfrak{M}', \mathfrak{l} \models \phi$. By the first part of the proof, we also get $\mathfrak{M}'', \mathfrak{l} \models \phi$ with $\mathfrak{M}'' = \langle \mathfrak{W}'', \mathfrak{R}', \mathfrak{V}' \rangle$ where $\mathfrak{V}''$ is the restriction of $\mathfrak{V}$ to $\mathfrak{W}''$ and $\mathfrak{W}'' = \{\mathfrak{l}\} \cup \{\mathfrak{l}_1, \ldots, \mathfrak{l}_{s^{\star}}, \mathfrak{n}_1, \mathfrak{m}_1, \ldots, \mathfrak{n}_T, \mathfrak{m}_T\}$. □

4.2 Model-checking for $\mathrm{MSL}(*, \Diamond)$ in P

It remains to show that the model-checking problem for $\mathrm{MSL}^f(*, \Diamond)$ is in P. The main difficulty to obtain a polynomial-time algorithm 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 12. 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 $\overset{s,n}{\sim}$ can be also defined on finite and functional pointed models leading to a natural variant of Lemma 11 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 $\mathsf{card}(\mathfrak{R}') \leq \mathrm{msize}(\phi)$ 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 $\mathrm{msize}(\phi)$. In forthcoming Algorithm 1, instead of working with finite and functional models, we shall 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 $r \geq 0$ (standing for the number of remaining edges up to s), $(\mathfrak{l}_0, \ldots, \mathfrak{l}_t) \in \mathbb{N}^+$ (standing for a sequence of 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 $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s)$ 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 $\mathrm{shrink}(\mathfrak{M}, \mathfrak{l}, s)$ be the finite and functional model $\mathfrak{M}' = \langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle$ such that $\mathfrak{R}' \overset{\text{def}}{=} \{(\mathfrak{l}_i, \mathfrak{l}_{i+1}) \mid i \in [0, t-1]\} \cup \{(\mathfrak{n}_1, \mathfrak{n}'_1), \ldots, (\mathfrak{n}_r, \mathfrak{n}'_r)\}$, where $\{(\mathfrak{n}_1, \mathfrak{n}'_1), \ldots, (\mathfrak{n}_r, \mathfrak{n}'_r)\}$ is a set of r edges in $\mathfrak{R} \setminus \mathfrak{R}_{\mathfrak{l},s}$ and the locations $\mathfrak{n}_1, \ldots, \mathfrak{n}_r$ have minimal values (minimality is used here to have a deterministic way to shrink). Lemma 13 below justifies the correctness of the abstraction.

Lemma 13. *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 $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathrm{abst}(\mathfrak{M}', \mathfrak{l}, s)$.*

The proof is by an easy verification. 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 conditions at first glance, the best way to understand what is really needed, is by proving forthcoming Lemma 14 and Lemma 15. Note also that in (v), $r^{3-i} = s_{3-i}$ is equivalent to $r^{3-i} = s_{3-i} - t^*_{3-i}$ with $t^*_{3-i} = 0$ (see (i)).

Let $\mathcal{F}_1 = ((\mathfrak{l}^1_0, \ldots, \mathfrak{l}^1_{t_1}), r^1)$ and $\mathcal{F}_2 = ((\mathfrak{l}^2_0, \ldots, \mathfrak{l}^2_{t_2}), r^2)$ be two abstract frames up to s_1 and s_2 respectively, 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, $r^1 + t^*_1 \leq r^2 + t^*_2$ (a factor is a contiguous sequence of locations).

Forthcoming Algorithm 1 shall operate with abstract frames and its correctness is partly based on forthcoming Lemma 14 and Lemma 15. For instance, Lemma 14 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 14. *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* $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1) \uplus \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$.

Proof. Let $s = s_1 + s_2$ with $s, s_1, s_2 \geq 1$. Let $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$, $\mathfrak{M}_1 = \langle \mathfrak{W}, \mathfrak{R}_1, \mathfrak{V} \rangle$ and $\mathfrak{M}_2 = \langle \mathfrak{W}, \mathfrak{R}_2, \mathfrak{V} \rangle$ be finite and functional models such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, and let $\mathfrak{l} \in \mathfrak{W}$. Let $\mathcal{F} = ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ be equal to $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s)$, and $\mathcal{F}_i = ((\mathfrak{l}^i_0, \ldots, \mathfrak{l}^i_{t_i}), r^i)$ be equal to $\mathrm{abst}(\mathfrak{M}_i, \mathfrak{l}, s_i)$ $(i = 1, 2)$. In order to show that $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1) \uplus \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$, we need to prove that the conditions (i)–(v) hold. When $t = 0$, $t_1 = t_2 = 0$ and therefore the conditions (i)-(v) trivially hold. In the sequel, we assume that $t > 0$. Moreover, for checking the satisfaction of the conditions (ii)-(v), we develop the case $i = 1$ only (as the case $i = 2$ can be deduced very easily).

(i) Since $s_1, s_2 \leq s$, we have $t_1, t_2 \leq t$. Therefore $\max(t_1, t_2) \leq t$. For the second property, since $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, then for all $(\mathfrak{n}, \mathfrak{n}') \in \mathfrak{R}$, we have $(\mathfrak{n}, \mathfrak{n}') \in \mathfrak{R}_1$ if and only if $(\mathfrak{n}, \mathfrak{n}') \notin \mathfrak{R}_2$. In particular, $(\mathfrak{l}_0, \mathfrak{l}_1) \in \mathfrak{R}_1$ if and only if $(\mathfrak{l}_0, \mathfrak{l}_1) \notin \mathfrak{R}_2$. So, either $t_1 = 0$ or $t_2 = 0$, then $t_1 \times t_2 = 0$. Finally, suppose $t > 0$, then $(\mathfrak{l}_0, \mathfrak{l}_1) \in \mathfrak{R}$. Again, either $(\mathfrak{l}_0, \mathfrak{l}_1) \in \mathfrak{R}_1$ or $(\mathfrak{l}_0, \mathfrak{l}_1) \in \mathfrak{R}_2$ (but not both). So, since $s_1, s_2 \geq 1$, then $(\mathfrak{l}_0, \mathfrak{l}_1)$ belongs to $\mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1)$ or $(\mathfrak{l}_0, \mathfrak{l}_1)$ belongs to $\mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$. Then we have $t_1 > 0$ or $t_2 > 0$, which implies $t_1 + t_2 > 0$ as wanted.

(ii) Direct, by definition of $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s)$ and $\mathfrak{R} = \mathfrak{R}_1 \uplus \mathfrak{R}_2$.

(iii) Suppose that $0 < t_1 < \min(s_1, t)$. As $\mathfrak{R} = \mathfrak{R}_1 \uplus \mathfrak{R}_2$, and $\mathfrak{l}_0 \neq \mathfrak{l}_{t_1}$, (since $t_1 < t$ and t_1 is maximal), we can conclude that $(\mathfrak{l}_{t_1}, \mathfrak{l}_{t_1+1}) \in \mathfrak{R}_2$. Hence, $\mathfrak{R}_2(\mathfrak{l}_0) = \emptyset$ and $\mathfrak{R}_2 \neq \emptyset$, which entails $r^2 > 0$.

(iv) Suppose that $0 < t_1$. Let us establish that $r^1 + r^2 \leq r + t^* - t^*_1$. Obviously, $\mathsf{card}(\mathfrak{R}) = \mathsf{card}(\mathfrak{R}_1) + \mathsf{card}(\mathfrak{R}_2)$. If $r + t^* < s$, then $\mathsf{card}(\mathfrak{R}) = r + t^*$. As $r^1 \leq \mathsf{card}(\mathfrak{R}_1) - t^*_1$ and $r^2 \leq \mathsf{card}(\mathfrak{R}_2)$, we get $r^1 + r^2 \leq \mathsf{card}(\mathfrak{R}_1) + \mathsf{card}(\mathfrak{R}_2) - t^*_1 = (r + t^*) - t^*_1$. Otherwise $(r + t^* = s)$, by definition of abstract frames, we have $t^*_1 + r^1 \leq s_1$ and $t^*_2 + r^2 \leq s_2$ with $t^*_2 = 0$. So, $r^1 + r^2 + t^*_1 \leq s_1 + s_2 = s = r + t^*$. Consequently, $r^1 + r^2 \leq (r + t^*) - t^*_1$.

(v) Suppose that $0 < t_1$ and $r^1 + r^2 < r + t^* - t^*_1$. Let us show that either $r^1 = s_1 - t^*_1$ or $r^2 = s_2$. Let $\mathfrak{R}' \subseteq \mathfrak{R}$ be the relation defined below:

$$\mathfrak{R}' = (\{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{t-1}, \mathfrak{l}_t)\} \setminus \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{t_1-1}, \mathfrak{l}_{t_1})\}) \cup \{(\mathfrak{n}_1, \mathfrak{n}'_1), \ldots, (\mathfrak{n}_r, \mathfrak{n}'_r)\},$$

where the set $\{(\mathfrak{n}_1, \mathfrak{n}'_1), \ldots, (\mathfrak{n}_r, \mathfrak{n}'_r)\}$ witnesses the value r in the abstract frame $\mathcal{F}$. We also have $\mathsf{card}(\mathfrak{R}') = r + t^* - t^*_1$. Every memory cell in $\mathfrak{R}'$ may contribute to the garbage part of either $\mathfrak{M}_1$ or $\mathfrak{M}_2$. As $r^1 + r^2 < r + t^* - t^*_1$, at least one memory cell in $\mathfrak{R}'$ does not need to be counted in r^1 or in r^2. Equivalently, for some $i \in \{1, 2\}$, $r^i = s_i - t^*_i$, i.e. the number of memory cells in the garbage part of $\mathfrak{M}_i$ is maximal with respect to the bound $s_i - t^*_i$. As $t^*_2 = 0$, we have $r^1 = s_1 - t^*_1$ or $r^2 = s_2$. □

By contrast, Lemma 15 below can be understood as a completeness result: the satisfaction of the conditions (i)-(v) can always be mimicked at the level of models.

Lemma 15. *Let $s = s_1 + s_2$ with $s, s_1, s_2 \geq 1$. Let $\mathfrak{M}$ be finite and functional, $\mathcal{F}_i = ((\mathfrak{l}^i_0, \ldots, \mathfrak{l}^i_{t_i}), r^i)$ 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 submodels $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$ and $\mathcal{F}_i = \mathrm{abst}(\mathfrak{M}_i, \mathfrak{l}, s_i)$ $(i \in \{1, 2\})$.*

Proof. Let $\mathfrak{M} = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ be a finite and functional model, $\mathcal{F}_i = ((\mathfrak{l}_0^i, \ldots, \mathfrak{l}_{t_i}^i), r^i)$ be an abstract frame up to s_i ($i \in \{1,2\}$) such that $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s) = \mathcal{F}_1 \uplus \mathcal{F}_2 = ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$. We will construct $\mathfrak{M}_1 = \langle \mathfrak{W}, \mathfrak{R}_1, \mathfrak{V} \rangle$ and $\mathfrak{M}_2 = \langle \mathfrak{W}, \mathfrak{R}_2, \mathfrak{V} \rangle$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathcal{F}_1 = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1)$ and $\mathcal{F}_2 = \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$.

When $t = 0$, we distinguish several cases. If $r < s$, then $r^1 + r^2 = r$. $\mathfrak{M}_1$ is made of r^1 edges from $\mathfrak{M}$ and $\mathfrak{M}_2$ is made of the other r^2 edges from $\mathfrak{M}$. If $r = s$ and $r^1 = s_1$, then $\mathfrak{M}_2$ is made of r^2 edges from $\mathfrak{M}$ and $\mathfrak{M}_1$ is defined with the remaining edges from $\mathfrak{M}$. If If $r = s$ and $r^1 < s_1$, then $\mathfrak{M}_1$ is made of r^1 edges from $\mathfrak{M}$ and $\mathfrak{M}_2$ is defined with the remaining edges from $\mathfrak{M}$.

Below, let us assume that $t > 0$ and $t_1 > 0$. Let $\mathfrak{R}' \subseteq \mathfrak{R}$ be the relation defined below:

$$\mathfrak{R}' = (\{(\mathfrak{l}_{t_1}, \mathfrak{l}_{t_1+1}), \ldots, (\mathfrak{l}_{t-1}, \mathfrak{l}_t)\} \setminus \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{t_1-1}, \mathfrak{l}_{t_1})\}) \cup \{(\mathfrak{n}_1, \mathfrak{n}_1'), \ldots, (\mathfrak{n}_r, \mathfrak{n}_r')\},$$

where the set $\{(\mathfrak{n}_1, \mathfrak{n}_1'), \ldots, (\mathfrak{n}_r, \mathfrak{n}_r')\}$ witnesses the value r in the abstract frame $\mathrm{abst}(\mathfrak{M}, \mathfrak{l}, s)$. We also have $\mathsf{card}(\mathfrak{R}') = r + t^* - t_1^*$ (similarly to the proof of Lemma 14).

Case 1: $t_1^* = s_1$. So $r^1 = 0$ as $t_1^* + r^1 \leq s_1$. By satisfaction of (iv), $r^1 + r^2 \leq r + t^* - t_1^*$ and therefore $r^2 \leq r + t^* - t_1^*$. Let $\mathfrak{R}_2$ be defined as a subset of r^2 elements of $\mathfrak{R}'$ such that if $0 < t_1 < \min(s_1, t)$, then we require that $(\mathfrak{l}_{t_1}, \mathfrak{l}_{t_1+1})$ belongs to $\mathfrak{R}_2$ (as $r^2 \geq 1$ by (iii)). We set $\mathfrak{R}_1 \overset{\text{def}}{=} \mathfrak{R} \setminus \mathfrak{R}_2$. It is easy to check that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathcal{F}_1 = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1)$ and $\mathcal{F}_2 = \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$. Note that we need to use the satisfaction of (ii), as in the rest of the proof.

Case 2: $t_1^* < s_1$. By (iv), we have $r^1 + r^2 \leq r + t^* - t_1^*$, which means that in the construction below, we will always have enough pairs to pick from the relation $\mathfrak{R}' \subseteq \mathfrak{R}$. Again, let us distinguish several cases.

Case 2.1: $r^1 = s_1 - t_1^*$. Let $\mathfrak{R}_2$ be defined as a subset of r^2 elements of $\mathfrak{R}'$ such that if $0 < t_1 < \min(s_1, t)$, then we require that $(\mathfrak{l}_{t_1}, \mathfrak{l}_{t_1+1})$ belongs to $\mathfrak{R}_2$ (as $r^2 \geq 1$ by (iii)). We set $\mathfrak{R}_1 \overset{\text{def}}{=} \mathfrak{R} \setminus \mathfrak{R}_2$. It is easy to check that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathcal{F}_1 = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1)$ and $\mathcal{F}_2 = \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$.

Case 2.2: $r^2 = s_2$ and $r^1 < s_1 - t_1^*$. Let $\mathfrak{R}_1$ be defined as the set $\{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{t_1-1}, \mathfrak{l}_{t_1})\}$ augmented with a subset of r^1 elements from $\mathfrak{R}'$. We set $\mathfrak{R}_2 \overset{\text{def}}{=} \mathfrak{R} \setminus \mathfrak{R}_1$. It is easy to check that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathcal{F}_1 = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1)$ and $\mathcal{F}_2 = \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$.

Case 2.3: $r^2 < s_2$ and $r^1 < s_1 - t_1^*$. By satisfaction of (v), $r^1 + r^2 = r + t^* - t_1^*$. So, let $\{X_1, X_2\}$ be a partition of $\mathfrak{R}'$ such that $\mathsf{card}(X_1) = r^1$, $\mathsf{card}(X_2) = r^2$ and if $0 < t_1 < \min(s_1, t)$, then we require that $(\mathfrak{l}_{t_1}, \mathfrak{l}_{t_1+1}) \in X_2$ (again, this is fine by satisfaction of (iii)). Note also that $r^2 < s_2$ and $r^1 < s_1 - t_1^*$ imply $r^1 + r^2 < s_2 + s_1 - t_1^*$ and therefore $r + t^* < s_1 + s_2 = s$. Hence $\mathsf{card}(\mathfrak{R}) = r + t^*$. Let $\mathfrak{R}_2 \overset{\text{def}}{=} X_2$ and $\mathfrak{R}_1 \overset{\text{def}}{=} X_1 \cup \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{t_1-1}, \mathfrak{l}_{t_1})\}$. It is easy to check that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathcal{F}_1 = \mathrm{abst}(\mathfrak{M}_1, \mathfrak{l}, s_1)$ and $\mathcal{F}_2 = \mathrm{abst}(\mathfrak{M}_2, \mathfrak{l}, s_2)$. □

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

Lemma 16. *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* $\mathsf{card}(\{(\mathcal{F}_1, \mathcal{F}_2) \mid \mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2,\ \mathcal{F}_i \text{ up to } s_i\}) \leq 2(s+1)(s_1+1)(s_2+1)$.

Proof. Let $\mathcal{F} = ((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ be an abstract frame up to s, $\mathcal{F}_1 = ((\mathfrak{l}_0^1, \ldots, \mathfrak{l}_{t_1}^1), r^1)$ be an abstract frame up to s_1 and $\mathcal{F}_2 = ((\mathfrak{l}_0^2, \ldots, \mathfrak{l}_{t_2}^2), r^2)$ be an abstract frame up to s_2 with $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$. It is obvious that $(r^1, r^2) \in [0, s_1] \times [0, s_2]$ and $((\mathfrak{l}_0^1, \ldots, \mathfrak{l}_{t_1}^1), (\mathfrak{l}_0^2, \ldots, \mathfrak{l}_{t_2}^2))$ belongs to the set

$$X = \{((\mathfrak{l}_0, \ldots, \mathfrak{l}_{t'}), (\mathfrak{l}_0)) \mid t' \leq t \leq s\} \cup \{((\mathfrak{l}_0), (\mathfrak{l}_0, \ldots, \mathfrak{l}_{t'})) \mid t' \leq t \leq s\},$$

where $\mathsf{card}(X) \leq 2(s+1)$. So, clearly the set of pairs of abstract frames in the statement has cardinal bounded by $2(s+1)(s_1+1)(s_2+1)$. □

Algorithm 1 below uses 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): $\mathrm{shrink}(((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r), s) \stackrel{\text{def}}{=} ((\mathfrak{l}_0, \ldots, \mathfrak{l}_{t'}), r')$ with

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

Note that the value $(\mathsf{card}(\{\mathfrak{l}_1, \ldots, \mathfrak{l}_t\}) - t'_*)$ corresponds to the number of edges removed from the first part of $\mathcal{F}$. It is easy to 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 (based on the fact that $((\mathfrak{l}_0, \ldots, \mathfrak{l}_t), r)$ is already an abstract frame).

Algorithm 1 only computes values for $T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_k)), k)$ as it would be time-consuming (and useless) to compute all the values $T(\mathcal{F}, k)$ (see how ψ_k is a subformula of the input formula). 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 $\mathrm{msize}(\cdot)$ 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 $\phi$
Out: Return 1 iff $\mathfrak{M}, \mathfrak{l} \models \phi$.
 1: function MC($\mathfrak{M}, \mathfrak{l}, \phi$)
 2:   $((\mathfrak{l}_0, \ldots, \mathfrak{l}_L), R) := \mathrm{abst}(\mathfrak{M}, \mathfrak{l}, \mathrm{msize}(\phi))$          ▷ $\mathsf{card}(\{\mathfrak{l}_1, \ldots, \mathfrak{l}_L\}) + R \leq \mathrm{msize}(\phi)$
 3:   $\psi_1, \ldots, \psi_M$ subformulae of $\phi$ 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 $\max\{len' \in [0, L - j] \mid \mathsf{card}(\{\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len'}\}) \leq \mathrm{msize}(\psi_k)\}$ do
 7:         for $r \leftarrow 0$ to $\max\{r' \in [0, R] \mid \mathsf{card}(\{\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}\}) + r' \leq \mathrm{msize}(\psi_k)\}$ do
 8:           $\mathcal{F} := ((\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}), r)$          ▷ $\mathcal{F} = \mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_k))$
 9:           case $\psi_k$ of
10:           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:             $T(\mathcal{F}, k) := \min(T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_1})), k_1), T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_2})), k_2))$
15:           $\Diamond\psi_{k'}$:  if ($len > 0$), $\mathcal{F}' := \mathrm{shrink}(((\mathfrak{l}_{j+1}, \ldots, \mathfrak{l}_{j+len}), r), \mathrm{msize}(\psi_{k'}))$          ▷ $k' < k$
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:             $s_1 := \mathrm{msize}(\psi_{k_1})$; $s_2 := \mathrm{msize}(\psi_{k_2})$          ▷ $\mathrm{msize}(\psi_k) = s_1 + s_2$
19:             $T(\mathcal{F}, k) := \max\{\min(T(\mathcal{F}_1, k_1), T(\mathcal{F}_2, k_2)) \mid \mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2,\ \mathcal{F}_i$ 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. Moreover, whenever a value $T(\mathcal{F}, k)$ is computed, the algorithm enforces that $\mathcal{F} = \mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_k))$. The algorithm runs in polynomial time thanks to Lemma 16. The following lemma establishes that the algorithm is correct and explains what is the rationale behind computing the values $T(\mathcal{F}, k)$.

Lemma 17. *For all $k \in [1, M]$, for all abstract frames $\mathcal{F} = ((\mathfrak{l}, \ldots), R')$ up to $\mathrm{msize}(\psi_k)$ 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 $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_k)) = \mathcal{F}$, we have $\mathfrak{M}', \mathfrak{l} \models \psi_k$.*

Proof. The proof is by structural induction on the formula ψ_k. The base cases for `emp` and for the propositional variables are by an easy verification. Below, we present the different cases in the induction step. Before doing so, note the equivalence between the statements below ($\psi_{k'}$ is a strict subformula of ψ_k).

- $T(\mathcal{F}, k') = 0$ (i.e. $T(\mathcal{F}, k') \neq 1$).
- There is $\mathfrak{M}''$ such that $\mathfrak{M}'' \subseteq \mathfrak{M}$, $\mathrm{abst}(\mathfrak{M}'', \mathfrak{l}, \mathrm{msize}(\psi_{k'})) = \mathcal{F}$, and $\mathfrak{M}'', \mathfrak{l} \not\models \psi_{k'}$ (by the induction hypothesis).
- For all $\mathfrak{M}' \subseteq \mathfrak{M}$ such that $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_{k'})) = \mathcal{F}$, we have $\mathfrak{M}', \mathfrak{l} \not\models \psi_{k'}$ (by the variant of Lemma 11 on finite models, as $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_{k'})) = \mathrm{abst}(\mathfrak{M}'', \mathfrak{l}, \mathrm{msize}(\psi_{k'}))$ implies $\mathfrak{M}'', \mathfrak{l} \overset{\mathrm{msize}(\psi_{k'}),n}{\sim} \mathfrak{M}', \mathfrak{l}$).

Consequently, as the induction hypothesis with $\psi_{k'}$ strict subformula of ψ_k, one can use also the following equivalence ($\sharp$) $T(\mathcal{F}, k') = 0$ iff for all finite and functional submodels $\mathfrak{M}' \subseteq \mathfrak{M}$ such that $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_{k'})) = \mathcal{F}$, we have $\mathfrak{M}', \mathfrak{l} \not\models \psi_{k'}$.

Case $\psi_k = \neg\psi_{k'}$. Suppose that $T(\mathcal{F}, k) = 1$ with $\mathcal{F} = ((\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}), r)$. By line 12, $T(\mathcal{F}, k') = 0$. As $\mathrm{msize}(\psi_k) = \mathrm{msize}(\psi_{k'})$, we have $\mathcal{F} = \mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k'}))$. Let $\mathfrak{M}' \subseteq \mathfrak{M}$ and $\mathfrak{l} \in \mathfrak{W}$ be such that $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_k)) = \mathcal{F}$. By the induction hypothesis with ($\sharp$), we get $\mathfrak{M}', \mathfrak{l} \not\models \psi_{k'}$ and therefore $\mathfrak{M}', \mathfrak{l} \models \psi_k$. Conversely, suppose that for all finite and functional submodels $\mathfrak{M}' \subseteq \mathfrak{M}$ and $\mathfrak{l} \in \mathfrak{W}$ such that $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_k)) = \mathcal{F}$, we have $\mathfrak{M}', \mathfrak{l} \models \psi_k$. Necessarily, $\mathfrak{l} = \mathfrak{l}_j$ and by definition of $\models$, we have $\mathfrak{M}', \mathfrak{l} \not\models \psi_{k'}$. Notice that $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_{k'})) = \mathcal{F}$, so by the induction hypothesis with ($\sharp$), $T(\mathcal{F}, k') = 0$. Therefore $T(\mathcal{F}, k) = 1$ by line 12.

Case $\psi_k = \psi_{k_1} \wedge \psi_{k_2}$. We recall that $\mathrm{msize}(\psi_k) = \max(\mathrm{msize}(\psi_{k_1}), \mathrm{msize}(\psi_{k_2}))$. Without loss of generality, we assume that $\mathrm{msize}(\psi_k) = \mathrm{msize}(\psi_{k_1})$. First suppose that $T(\mathcal{F}, k) = 1$ with $\mathcal{F} = ((\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}), r)$. So, by line 13, $T(\mathcal{F}, k_1) = 1$ and $T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_2})), k_2) = 1$. Let $\mathfrak{M}' \subseteq \mathfrak{M}$ and $\mathfrak{l} \in \mathfrak{W}$ be such that $\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_k)) = \mathcal{F}$ (necessarily $\mathfrak{l} = \mathfrak{l}_j$). By the induction hypothesis, we get $\mathfrak{M}', \mathfrak{l} \models \psi_{k_1}$.

In order to show that $\mathfrak{M}', \mathfrak{l} \models \psi_{k_2}$, first we need to state a technical property.

($\dagger$) Let $\mathfrak{M}^\star = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$ be a model, $\mathfrak{l} \in \mathfrak{W}$, $n \geq 1$ and $1 \leq s' \leq s$. There is a submodel $\mathfrak{M}' \subseteq \mathfrak{M}^\star$ such that

- $\mathrm{shrink}(\mathrm{abst}(\mathfrak{M}^\star, \mathfrak{l}, s), s') = \mathrm{abst}(\mathfrak{M}', \mathfrak{l}, s')$.
- $\mathfrak{M}^\star, \mathfrak{l} \overset{s',n}{\sim} \mathfrak{M}', \mathfrak{l}$.

Roughly speaking, shrinking an abstract frame obtained from a model $\mathfrak{M}^\star$ can be equally performed by abstracting a submodel of $\mathfrak{M}^\star$. So, by ($\dagger$), there is $\mathfrak{M}'' \subseteq \mathfrak{M}'$ such that $\mathrm{shrink}(\mathrm{abst}(\mathfrak{M}', \mathfrak{l}, \mathrm{msize}(\psi_k)), \mathrm{msize}(\psi_{k_2})) = \mathrm{abst}(\mathfrak{M}'', \mathfrak{l}, \mathrm{msize}(\psi_{k_2}))$ and $\mathfrak{M}', \mathfrak{l} \overset{\mathrm{msize}(\psi_{k_2}),n}{\sim} \mathfrak{M}'', \mathfrak{l}$ (all the propositional variables in ϕ are among p_1, ..., p_n). By the induction hypothesis, $\mathfrak{M}'', \mathfrak{l} \models \psi_{k_2}$. By the variant of Lemma 11 on finite models, we get $\mathfrak{M}', \mathfrak{l} \models \psi_{k_2}$.

For the other direction, suppose that for all $\mathfrak{M}' \subseteq \mathfrak{M}$ such that $\mathcal{F} = ((\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}), r) = \mathrm{abst}(\mathfrak{M}', \mathfrak{l}_j, \mathrm{msize}(\psi_k))$, we have $\mathfrak{M}', \mathfrak{l}_j \models \psi_k$. So obviously, $\mathfrak{M}', \mathfrak{l}_j \models \psi_{k_1}$ and $\mathfrak{M}', \mathfrak{l}_j \models \psi_{k_2}$. By the induction hypothesis, we have

$$T(\mathcal{F}, k_1) = T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_1})), k_1) = 1.$$

It remains to show that $T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_2})), k_2) = 1$ with $\mathrm{msize}(\psi_{k_2}) \leq \mathrm{msize}(\psi_{k_1}) = \mathrm{msize}(\psi_k)$. By assumption, we have that for all $\mathfrak{M}' \subseteq \mathfrak{M}$ such that $\mathcal{F}=\mathrm{abst}(\mathfrak{M}', \mathfrak{l}_j, \mathrm{msize}(\psi_k))$, we have $\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_2})) = \mathrm{abst}(\mathfrak{M}', \mathfrak{l}_j, \mathrm{msize}(\psi_{k_2}))$.

This implies that for all $\mathfrak{M}' \subseteq \mathfrak{M}$ such that $\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_2})) = \mathrm{abst}(\mathfrak{M}', \mathfrak{l}_j, \mathrm{msize}(\psi_{k_2}))$, we have $\mathfrak{M}', \mathfrak{l} \models \psi_{k_2}$. By the induction hypothesis, we get $T(\mathrm{shrink}(\mathcal{F}, \mathrm{msize}(\psi_{k_2})), k_2) = 1$. By line 14, we get $T(\mathcal{F}, k) = 1$.

Case $\psi_k = \Diamond\psi_{k'}$. First suppose that $T(\mathcal{F}, k) = 1$ with $\mathcal{F} = ((\mathfrak{l}_j, \ldots, \mathfrak{l}_{j+len}), r)$. We recall that $\mathrm{msize}(\Diamond\psi_{k'}) = 1 + \mathrm{msize}(\psi_{k'})$. So we have $len > 0$ and $T(((\mathfrak{l}_{j+1}, \ldots, \mathfrak{l}_{j+len}), r), k') = 1$.

Note that $\mathrm{shrink}(((\mathfrak{l}_{j+1},\ldots,\mathfrak{l}_{j+len}),r),\mathrm{msize}(\psi_{k'})) = ((\mathfrak{l}_{j+1},\ldots,\mathfrak{l}_{j+len}),r)$, since we have $\mathrm{shrink}(((\mathfrak{l}_{j},\ldots,\mathfrak{l}_{j+len}),r),\mathrm{msize}(\psi_{k'})+1) = ((\mathfrak{l}_{j},\ldots,\mathfrak{l}_{j+len}),r)$ by hypothesis. Let $\mathfrak{M}' \subseteq \mathfrak{M}$ and $\mathfrak{l} \in \mathfrak{W}$ be such that $\mathrm{abst}(\mathfrak{M}',\mathfrak{l},\mathrm{msize}(\psi_k)) = \mathcal{F}$. Observe that necessarily $\mathfrak{l} = \mathfrak{l}_j$, and as a consequence, $\mathrm{abst}(\mathfrak{M}',\mathfrak{l}_{j+1},\mathrm{msize}(\psi_{k'})) = ((\mathfrak{l}_{j+1},\ldots,\mathfrak{l}_{j+len}),r)$. By the induction hypothesis, $\mathfrak{M}',\mathfrak{l}_{j+1} \models \psi_{k'}$. Then, since $(\mathfrak{l}_j,\mathfrak{l}_{j+1}) \in \mathfrak{R}'$, we get $\mathfrak{M}',\mathfrak{l}_j \models \psi_k$. Conversely, suppose that for all finite and functional submodels $\mathfrak{M}' \subseteq \mathfrak{M}$ and $\mathfrak{l} \in \mathfrak{W}$ such that $\mathrm{abst}(\mathfrak{M}',\mathfrak{l},\mathrm{msize}(\psi_k)) = \mathcal{F}$, we have $\mathfrak{M}',\mathfrak{l} \models \psi_k$. Necessarily, $\mathfrak{l} = \mathfrak{l}_j$ and by definition of $\models$, we have $\mathfrak{M}',\mathfrak{l}_{j+1} \models \psi_{k'}$. So, for all finite and functional submodels $\mathfrak{M}'' \subseteq \mathfrak{M}$ such that $\mathrm{abst}(\mathfrak{M}'',\mathfrak{l}_{j+1},\mathrm{msize}(\psi_{k'})) = ((\mathfrak{l}_{j+1},\ldots,\mathfrak{l}_{j+len}),r)$, we have $\mathfrak{M}'',\mathfrak{l}_{j+1} \models \psi_{k'}$. Here, we use the fact that two $\overset{\mathrm{msize}(\psi_{k'}),n}{\sim}$-equivalent submodels with the same frame abstraction satisfy the same formulae ψ with $\mathrm{msize}(\psi) \leq \mathrm{msize}(\psi_{k'})$ by the variant of Lemma 11 on finite models. Consequently, $T(((\mathfrak{l}_{j+1},\ldots,\mathfrak{l}_{j+len}),r),k') = 1$ and therefore $T(\mathcal{F},k) = 1$.

Case $\psi_k = \psi_{k_1} * \psi_{k_2}$. Suppose that $T(\mathcal{F},k) = 1$ with $\mathcal{F} = ((\mathfrak{l}_j,\ldots,\mathfrak{l}_{j+len}),r)$. By line 19, there exist $\mathcal{F}_1$ and $\mathcal{F}_2$ abstract frames up to s_1 and s_2 respectively, such that $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$ and $T(\mathcal{F}_1,k_1) = T(\mathcal{F}_2,k_2) = 1$. It is easy to check that $\mathcal{F}_1 \subseteq ((\mathfrak{l}_0,\ldots,\mathfrak{l}_L),R)$ and $\mathcal{F}_2 \subseteq ((\mathfrak{l}_0,\ldots,\mathfrak{l}_L),R)$, which allows us to apply below the induction hypothesis. Let $\mathfrak{M}' \subseteq \mathfrak{M}$ be such that $\mathrm{abst}(\mathfrak{M}',\mathfrak{l},\mathrm{msize}(\psi_k)) = \mathcal{F}$. By Lemma 15, there are submodels $\mathfrak{M}'_1$ and $\mathfrak{M}'_2$ such that $\mathfrak{M}' = \mathfrak{M}'_1 \uplus \mathfrak{M}'_2$, $\mathcal{F}_1 = \mathrm{abst}(\mathfrak{M}'_1,\mathfrak{l},s_1)$ and $\mathcal{F}_2 = \mathrm{abst}(\mathfrak{M}'_2,\mathfrak{l},s_2)$. As $\mathfrak{M}'_1$ and $\mathfrak{M}'_2$ are also submodels of $\mathfrak{M}$ and each $s_i = \mathrm{msize}(\psi_{k_i})$, by the induction hypothesis we get $\mathfrak{M}'_1,\mathfrak{l} \models \psi_{k_1}$ and $\mathfrak{M}'_2,\mathfrak{l} \models \psi_{k_2}$. Consequently, $\mathfrak{M}',\mathfrak{l} \models \psi_k$.

Now suppose that for all finite and functional submodels $\mathfrak{M}' \subseteq \mathfrak{M}$ and $\mathfrak{l} \in \mathfrak{W}$ such that $\mathrm{abst}(\mathfrak{M}',\mathfrak{l},\mathrm{msize}(\psi_k)) = \mathcal{F}$, we have $\mathfrak{M}',\mathfrak{l} \models \psi_k$. As $\mathcal{F} \subseteq ((\mathfrak{l}_0,\ldots,\mathfrak{l}_L),R)$, there is at least one model $\mathfrak{M}'$ satisfying the condition. By definition of $\models$, there are $\mathfrak{M}'_1$ and $\mathfrak{M}'_2$ such that $\mathfrak{M}' = \mathfrak{M}'_1 \uplus \mathfrak{M}'_2$, $\mathfrak{M}'_1,\mathfrak{l} \models \psi_{k_1}$ and $\mathfrak{M}'_2,\mathfrak{l} \models \psi_{k_2}$. As $\mathrm{msize}(\psi_k) = \mathrm{msize}(\psi_{k_1}) + \mathrm{msize}(\psi_{k_2})$, by Lemma 14, we have

$$\mathrm{abst}(\mathfrak{M}',\mathfrak{l},\mathrm{msize}(\psi_k)) = \mathrm{abst}(\mathfrak{M}'_1,\mathfrak{l},\mathrm{msize}(\psi_{k_1})) \uplus \mathrm{abst}(\mathfrak{M}'_2,\mathfrak{l},\mathrm{msize}(\psi_{k_2})).$$

For all finite and functional submodels $\mathfrak{M}' \subseteq \mathfrak{M}$ such that

$$\mathrm{abst}(\mathfrak{M}',\mathfrak{l},\mathrm{msize}(\psi_{k_i})) = \mathcal{F}_i,$$

we have $\mathfrak{M}',\mathfrak{l} \models \psi_{k_i}$. Here, we use the fact that two $\overset{\mathrm{msize}(\psi_{k_i}),n}{\sim}$-equivalent submodels with the same frame abstraction satisfy the same formulae ψ with $\mathrm{msize}(\psi) \leq \mathrm{msize}(\psi_{k_i})$ by the variant of Lemma 11 on finite models. By the induction hypothesis, we get $T(\mathcal{F}_1,k_1) = 1$ and $T(\mathcal{F}_2,k_2) = 1$, which entails that $T(\mathcal{F},k) = 1$. □

So, by taking $k = M$ and $\mathcal{F} = ((\mathfrak{l}_0,\ldots,\mathfrak{l}_L),R)$, we get that $T(\mathcal{F},k) = 1$ iff $\mathfrak{M},\mathfrak{l} \models \phi$. Again, here we use the fact that two $\overset{\mathrm{msize}(\phi),n}{\sim}$-equivalent submodels with the same frame abstraction satisfy the same formulae ψ with $\mathrm{msize}(\psi) \leq \mathrm{msize}(\phi)$ by the variant Lemma 11 on finite models.

So, we can characterise the complexity of the model-checking problem for $\mathrm{MSL}^f(*,\Diamond)$ by taking advantage of Lemma 16 (especially to handle the lines 17-19 efficiently).

Lemma 18. *The model checking problem for* $\mathrm{MSL}^f(*,\Diamond)$ *is in* P.

Then we can conclude.

Theorem 19. *The satisfiability problem for* $\mathrm{MSL}(*,\Diamond)$ *is* NP*-complete.*

Proof. NP-hardness follows from the NP-completeness of the satisfiability problem for propositional calculus. By Lemma 12, in order to get the NP upper bound, guess a finite and functional model $\mathfrak{M}$ with $\mathsf{card}(\mathfrak{W}) \leq 1 + 2 \times \mathrm{msize}(\phi)$ (polynomial value in the size of ϕ) and $\mathfrak{l} \in \mathfrak{W}$, and then check whether $\mathfrak{M},\mathfrak{l} \models \phi$, which can be done in polynomial-time by Lemma 18. □

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 19, we obtain the result below.

Corollary 20. *The satisfiability problem of* $\mathrm{MSL}(\Diamond, \langle \mathsf{gsb} \rangle)$ *is* NP*-complete.*

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

In this section, we establish that the satisfiability problem for $\mathrm{MSL}(*, \langle \neq \rangle)$ is NP-complete and its model-checking problem is in P. In order 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*, written $w_*(\phi)$, as follows:

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

Lemma 21. *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 $\mathsf{card}(\mathfrak{R}) = \mathsf{card}(\mathfrak{R}')$ in the statement of Lemma 21, then $\mathfrak{M}, \mathfrak{l}$ and $\mathfrak{M}', \mathfrak{l}$ satisfy exactly the same formulae in $\mathrm{MSL}(*, \langle \neq \rangle)$.

Proof. The proof is by structural induction. The base cases are by an easy verification ($\mathfrak{M}$ and $\mathfrak{M}'$ are built from the same valuation $\mathfrak{V}$) as well as the cases in the induction step for the Boolean connectives.

- Suppose that $\mathfrak{M}, \mathfrak{l} \models \langle \neq \rangle \psi$. There is $\mathfrak{l}' \neq \mathfrak{l}$ such that $\mathfrak{M}, \mathfrak{l}' \models \psi$. As $w_*(\langle \neq \rangle \psi) = w_*(\psi)$ and $w_*(\psi) \leq \alpha$, by the induction hypothesis, we have $\mathfrak{M}', \mathfrak{l}' \models \psi$. Hence, $\mathfrak{M}', \mathfrak{l} \models \langle \neq \rangle \psi$.
- Suppose that $\mathfrak{M}, \mathfrak{l} \models \psi_1 * \psi_2$. There exists a partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R}$ such that $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_2$. Let us make a case analysis.

  Case $\mathsf{card}(\mathfrak{R}) \leq \alpha$. So, $\mathsf{card}(\mathfrak{R}') = \mathsf{card}(\mathfrak{R})$, $w_*(\psi_1) \leq \alpha$ and $w_*(\psi_2) \leq \alpha$. There exists a partition $\{\mathfrak{R}'_1, \mathfrak{R}'_2\}$ of $\mathfrak{R}'$ such that $\mathsf{card}(\mathfrak{R}'_1) = \mathsf{card}(\mathfrak{R}_1)$ and $\mathsf{card}(\mathfrak{R}'_2) = \mathsf{card}(\mathfrak{R}_2)$. So, for all $j \in \{1, 2\}$, $\min(\mathsf{card}(\mathfrak{R}_j), \alpha) = \min(\mathsf{card}(\mathfrak{R}'_j), \alpha)$. By the induction hypothesis, $\langle \mathbb{N}, \mathfrak{R}'_1, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}'_2, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_2$. Hence, $\mathfrak{M}', \mathfrak{l} \models \psi_1 * \psi_2$.

  Case $\mathsf{card}(\mathfrak{R}) > \alpha$, $\mathsf{card}(\mathfrak{R}_1) \leq w_*(\psi_1)$ and $\mathsf{card}(\mathfrak{R}_2) > w_*(\psi_2)$. Then we have there exists a partition $\{\mathfrak{R}'_1, \mathfrak{R}'_2\}$ of $\mathfrak{R}'$ such that $\mathsf{card}(\mathfrak{R}'_1) = \mathsf{card}(\mathfrak{R}_1)$ and $\min(\mathsf{card}(\mathfrak{R}'_2), w_*(\psi_2)) = \min(\mathsf{card}(\mathfrak{R}_2), w_*(\psi_2))$. Then we have $\min(\mathsf{card}(\mathfrak{R}_1), w_*(\psi_1)) = \min(\mathsf{card}(\mathfrak{R}'_1), w_*(\psi_1))$ and also $\min(\mathsf{card}(\mathfrak{R}_2), w_*(\psi_2)) = \min(\mathsf{card}(\mathfrak{R}'_2), w_*(\psi_2))$. By the induction hypothesis, $\langle \mathbb{N}, \mathfrak{R}'_1, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}'_2, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_2$. Hence, $\mathfrak{M}', \mathfrak{l} \models \psi_1 * \psi_2$.

  Case $\mathsf{card}(\mathfrak{R}) > \alpha$, $\mathsf{card}(\mathfrak{R}_1) > w_*(\psi_1)$ and $\mathsf{card}(\mathfrak{R}_2) \leq w_*(\psi_2)$. The proof is similar to the previous case.

  Case $\mathsf{card}(\mathfrak{R}) > \alpha$, $\mathsf{card}(\mathfrak{R}_1) > w_*(\psi_1)$ and $\mathsf{card}(\mathfrak{R}_2) > w_*(\psi_2)$. Then we have there exists a partition $\{\mathfrak{R}'_1, \mathfrak{R}'_2\}$ of $\mathfrak{R}'$ such that $\mathsf{card}(\mathfrak{R}'_1) = w_*(\psi_1)$ and $\min(\mathsf{card}(\mathfrak{R}'_2), w_*(\psi_2)) = \min(\mathsf{card}(\mathfrak{R}_2), w_*(\psi_2))$. Then $\min(\mathsf{card}(\mathfrak{R}_1), w_*(\psi_1)) = \min(\mathsf{card}(\mathfrak{R}'_1), w_*(\psi_1))$ and also $\min(\mathsf{card}(\mathfrak{R}_2), w_*(\psi_2)) = \min(\mathsf{card}(\mathfrak{R}'_2), w_*(\psi_2))$. By the induction hypothesis, we obtain $\langle \mathbb{N}, \mathfrak{R}'_1, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}'_2, \mathfrak{V} \rangle, \mathfrak{l} \models \psi_2$. Hence, $\mathfrak{M}', \mathfrak{l} \models \psi_1 * \psi_2$. □

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 in order to deal with submodels in full generality. 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 21, this definition makes sense as two models with the same valuation and with the same cardinal of the accessibility 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 $\mathfrak{W} \stackrel{\text{def}}{=} \{\mathfrak{l}\} \cup \{\mathfrak{l}_j^{i,\beta'} \mid j \in \{1,2\},\ i \in [1, N],\ \beta' \in [0, \beta]\} \cup \{\mathfrak{l}, \mathfrak{l}' \mid (\mathfrak{l}, \mathfrak{l}') \in \mathfrak{R}\}$.

Lemma 22. *We have* $\langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V}\rangle, \mathfrak{l} \models \phi$.

Proof. By induction, we show that for all $\mathfrak{l}' \in \mathfrak{W}$, for all $\mathfrak{R}' \subseteq \mathfrak{R}$, for all subformulae ψ of ϕ, we have $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \psi$ iff $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \psi$. Again, the base cases are by an easy verification as well as the cases in the induction step for the Boolean connectives.

- Suppose that $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \langle\neq\rangle\psi_j$ for some $j \in [1, N]$. Let $\mathsf{card}(\mathfrak{R}') = \beta'$. There is necessarily $\mathfrak{l}_k^{j,\beta'} \neq \mathfrak{l}'$ for some $k \in \{1, 2\}$ such that $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}_k^{j,\beta'} \models \psi_j$. Indeed, by Lemma 21, two models with the same valuation and with the same cardinal of the relation satisfy the same formulae. By the induction hypothesis, $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}_k^{j,\beta'} \models \psi_j$ (as $\mathfrak{l}_k^{j,\beta'} \in \mathfrak{W}$) and therefore $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \langle\neq\rangle\psi_j$.

  Conversely, suppose that $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \langle\neq\rangle\psi_j$. There is some $\mathfrak{l}'' \in \mathfrak{W} \setminus \{\mathfrak{l}'\}$ such that $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}'' \models \psi_j$. By the induction hypothesis, we have $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}'' \models \psi_j$ and therefore $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \langle\neq\rangle\psi_j$.
- Suppose that $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1 * \psi_2$. So, there is a partition $\{\mathfrak{R}'_1, \mathfrak{R}'_2\}$ of $\mathfrak{R}'$ such that $\langle \mathbb{N}, \mathfrak{R}'_1, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}'_2, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, $\langle \mathfrak{W}, \mathfrak{R}'_1, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}, \mathfrak{R}'_2, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_2$. Consequently, $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1 * \psi_2$.

  Conversely, suppose that $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1 * \psi_2$. So, there is a partition $\{\mathfrak{R}'_1, \mathfrak{R}'_2\}$ of $\mathfrak{R}'$ such that $\langle \mathfrak{W}, \mathfrak{R}'_1, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathfrak{W}, \mathfrak{R}'_2, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_2$. By the induction hypothesis, we have $\langle \mathbb{N}, \mathfrak{R}'_1, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1$ and $\langle \mathbb{N}, \mathfrak{R}'_2, \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_2$. Consequently, $\langle \mathbb{N}, \mathfrak{R}', \mathfrak{V}\rangle, \mathfrak{l}' \models \psi_1 * \psi_2$. □

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

Corollary 23. *Let ϕ be a formula in* MSL$(*, \langle\neq\rangle)$. *ϕ is satisfiable iff ϕ is* MSL$^f(*, \langle\neq\rangle)$ *satisfiable in a model with* $\mathsf{card}(\mathfrak{W}) \leq 1 + 2|\phi| \times w_*(\phi)$.

In the expression $\mathsf{card}(\mathfrak{W}) \leq 1 + 2|\phi| \times w_*(\phi)$, the value $|\phi|$ can be replaced by the number of distinct subformulae of ϕ of the form $\langle\neq\rangle\psi$.

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

Lemma 24. *The model-checking problem for* 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 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 $\mathsf{card}(\mathfrak{R}) = \beta$. To determine whether $\mathfrak{M}, \mathfrak{l} \models \phi$, we use a labelling algorithm (see Algorithm 2) 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 22) 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 $\mathrm{MSL}^f(*, \langle\neq\rangle)$

```
In: A finite and functional model $\mathfrak{M} = \langle [0,K], \mathfrak{R}, \mathfrak{V} \rangle$, $K \geq 0$, an $\mathrm{MSL}(*, \langle\neq\rangle)$ formula $\phi$
Out: Return 1 iff $\mathfrak{M}, 0 \models \phi$.
 1: function MC($\mathfrak{M}, \mathfrak{l}, \phi$)
 2:   $\psi_1, \ldots, \psi_M$ subformulae of $\phi$ in increasing size          ▷ $\psi_M = \phi$
 3:   $\beta := \mathsf{card}(\mathfrak{R})$
 4:   for $j \leftarrow 0$ to $\beta$ do
 5:     for $k \leftarrow 1$ to $M$ do
 6:       for $i \leftarrow 0$ to $K$ do
 7:         case $\psi_k$ of
 8:         emp:  $T(i,j,k) := 1$ if $(j = 0)$, otherwise 0
 9:         $p$:  $T(i,j,k) := 1$ if $i \in \mathfrak{V}(p)$, otherwise 0
10:         $\neg\psi_{k'}$:  $T(i,j,k) := 1 - T(i,j,k')$                        ▷ $k' < k$
11:         $\psi_{k_1} \wedge \psi_{k_2}$:  $T(i,j,k) := \min(T(i,j,k_1), T(i,j,k_2))$   ▷ $k_1, k_2 < k$
12:         $\langle\neq\rangle\psi_{k'}$:                                        ▷ $k' < k$
13:           $T(i,j,k) := \max(T(1,j,k'), \ldots, T(i-1,j,k'), T(i+1,j,k'), \ldots, T(K,j,k'))$
14:         $\psi_{k_1} * \psi_{k_2}$:                                            ▷ $k_1, k_2 < k$
15:           $T(i,j,k) := \max\{\min(T(i,I,k_1), T(i,J,k_2)) \mid I + J = j$ and $I, J \geq 0\}$
16:         end case
17:   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 shown below, which entails that the satisfaction of $\mathfrak{M}, \mathfrak{l} \models \phi$ is equivalent to $T(0, \beta, M) = 1$.

So, by structural induction, let us show that $T(i,j,k) = 1$ iff $\langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle, i \models \psi_k$ with $\mathsf{card}(\mathfrak{R}') = j$. The model $\mathfrak{M}' = \langle \mathfrak{W}, \mathfrak{R}', \mathfrak{V} \rangle$ is a submodel of $\mathfrak{M}$ with $\mathsf{card}(\mathfrak{R}') = j$.

Base case: $\psi_k = p$**.** $T(i,j,k) = 1$ iff $i \in \mathfrak{V}(p)$ (see Line 9), iff (by definition of $\models$) $\mathfrak{M}', i \models p$.

Base case: $\psi_k = \mathtt{emp}$**.** $T(i,j,k) = 1$, iff $j = 0$ (see Line 8), that by assumption on $\mathfrak{R}'$, means $\mathfrak{R}' = \emptyset$, iff (by definition of $\models$), $\mathfrak{M}', i \models \mathtt{emp}$.

Case: $\psi_k = \psi_{k_1} \wedge \psi_{k_2}$**.** $T(i,j,k) = 1$ iff $T(i,j,k_1) = T(i,j,k_2) = 1$ (see Line 11). This is the case iff (by IH) $\mathfrak{M}', i \models \psi_{k_1}$ and $\mathfrak{M}', i \models \psi_{k_2}$, which is equivalent to $\mathfrak{M}', i \models \psi_{k_1} \wedge \psi_{k_2}$.

The case $\psi_k = \neg\psi_{k'}$ in the induction step is shown similarly.

Case: $\psi_k = \langle\neq\rangle\psi_{k'}$**.** $T(i,j,k) = 1$, there is $i' \neq i$ in $[0,K]$ such that $T(i',j,k') = 1$ (see Line 13). This is the case iff (by IH) there is $i' \neq i$ in $[0,K]$ such that $\mathfrak{M}', i' \models \psi_{k'}$, which is equivalent to $\mathfrak{M}', i \models \langle\neq\rangle\psi_{k'}$ (by definition of $\models$).

Case: $\psi_k = \psi_{k_1} * \psi_{k_2}$**.** $T(i,j,k) = 1$, iff there are I and J such that $j = I + J$ and $T(i,I,k_1) = T(i,J,k_2) = 1$ (see Line 15), iff (by IH) there are I and J such that $j = I + J$ and for all submodels $\mathfrak{M}'_1 = \langle \mathfrak{W}, \mathfrak{R}'_1, \mathfrak{V} \rangle$ and $\mathfrak{M}'_2 = \langle \mathfrak{W}, \mathfrak{R}'_2, \mathfrak{V} \rangle$ with $\mathsf{card}(\mathfrak{R}'_1) = I$ and $\mathsf{card}(\mathfrak{R}'_2) = J$, we have $\mathfrak{M}'_1, i \models \psi_{k_1}$ and $\mathfrak{M}'_2, i \models \psi_{k_2}$. Equivalently, by Lemma 21 there are I, J and submodels $\mathfrak{M}'_1 = \langle \mathfrak{W}, \mathfrak{R}'_1, \mathfrak{V} \rangle$ and $\mathfrak{M}'_2 = \langle \mathfrak{W}, \mathfrak{R}'_2, \mathfrak{V} \rangle$ with $\mathsf{card}(\mathfrak{R}'_1) = I$, $\mathsf{card}(\mathfrak{R}'_2) = J$ and $\mathfrak{M}' = \mathfrak{M}'_1 \uplus \mathfrak{M}'_2$, such that $\mathfrak{M}'_1, i \models \psi_{k_1}$ and $\mathfrak{M}'_2, i \models \psi_{k_2}$, iff $\mathfrak{M}', i \models \psi_{k_1} * \psi_{k_2}$. □

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

Theorem 25. *The satisfiability problem for* $\mathrm{MSL}(*, \langle\neq\rangle)$ *is* NP*-complete.*

Proof. As $\mathrm{MSL}(*, \langle\neq\rangle)$ contains the propositional calculus, NP-hardness is immediate. In order to get the NP upper bound, guess a finite and functional model $\mathfrak{M}$ with $\mathsf{card}(\mathfrak{W}) \leq 1 + 2|\phi| \times w_*(\phi)$ (polynomial value in the size of ϕ) and $\mathfrak{l} \in \mathfrak{W}$, and then check whether $\mathfrak{M}, \mathfrak{l} \models \phi$, which can be done in polynomial-time by Lemma 24. □

5 $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$: a Tower-complete fragment of MSL

In this section, we show that the satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ is Tower-complete though both $\mathrm{MSL}(*, \Diamond)$ and $\mathrm{MSL}(*, \langle\neq\rangle)$ admit NP-complete such problems (see Theorem 19 and Theorem 25). The upper bound is from Section 3 whereas the proof for Tower-hardness consists of two parts. First, we show that there is a (global) formula in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$ that characterises the linear structures. This is of interest for its own sake. Then, we reduce the nonemptiness problem for star-free expressions (precisely interpreted by finite words, i.e. linear structures) into the satisfiability problem.

5.1 Encoding linear structures

The goal of this section is to design a global formula in $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$, namely $\phi_{\exists \mathtt{ls}}$, such that for all models $\mathfrak{M}$, we have $\mathfrak{M} \models \phi_{\exists \mathtt{ls}}$ 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 $X \subseteq \mathrm{PROP}$, the relation $\{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ roughly 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 ($(\mathfrak{s}, \mathfrak{h})$ is a memory state with a store $\mathfrak{s}$ and a heap $\mathfrak{h}$, see Section 2.2):

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

The notation $\{\mathfrak{l}_0 \mapsto \mathfrak{l}_1, \mathfrak{l}_1 \mapsto \mathfrak{l}_2, \ldots, \mathfrak{l}_{n-1} \mapsto \mathfrak{l}_n\}$ refers to a heap $\mathfrak{h}$ with $\mathrm{dom}(\mathfrak{h}) = \{\mathfrak{l}_0, \ldots, \mathfrak{l}_{n-1}\}$ and $\mathfrak{h}(\mathfrak{l}_i) = \mathfrak{l}_{i+1}$ for all $i \in [0, n-1]$. 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 the reflexive and transitive closure modality $\langle\star\rangle$ (unlike MLH [31]), 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 4, we illustrate these concepts. This terminology making reference to trees is best understood if we think the definitions with respect to $\mathfrak{R}^{-1}$.

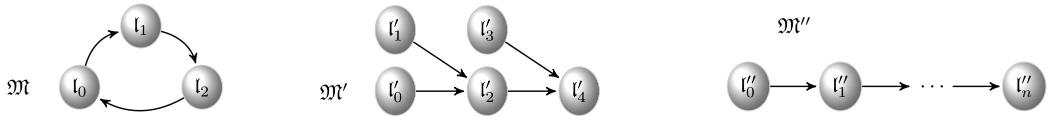


Figure 4: $\mathfrak{M}$ is a MCC and a loop, with no leaves and no pre-roots; $\mathfrak{M}'$ is a MCC with three leaves ($\mathfrak{l}'_0$, $\mathfrak{l}'_1$ and $\mathfrak{l}'_3$) and two pre-roots ($\mathfrak{l}'_2$ and $\mathfrak{l}'_3$); $\mathfrak{M}''$ is linear.

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 26. *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.*

Proof. First, suppose that $\mathfrak{R} = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ for some $n \geq 1$ such that for all $i \neq j \in [0, n]$, we have $\mathfrak{l}_i \neq \mathfrak{l}_j$. As for all $i \neq j$, we have $\mathfrak{l}_i \neq \mathfrak{l}_j$, we can conclude that $\mathfrak{M}$ is loop-free. Moreover, $\mathfrak{M}$ has a unique leaf, namely $\mathfrak{l}_0$.
Now suppose that $\mathfrak{M}$ is loop-free and has a unique leaf. As $\mathfrak{M}$ is loop-free, $\mathfrak{R}^{-1}$ can be seen as a finite collection of non-empty finite trees. If there are at least two finite trees, this contradicts the uniqueness property on leaves. Similarly, if the unique tree is not a word (i.e. has a linear structure), this also contradicts the uniqueness property. Consequently, $\mathfrak{R}^{-1}$ represents a finite word and therefore $\mathfrak{M}$ is linear. □

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

Lemma 27. *Let $\mathfrak{M}$ be an* MSL *model. $\mathfrak{M} \models \exists\mathsf{Loop}$ iff $\mathfrak{M}$ has at least one loop.*

Proof. First suppose that $\mathfrak{M}$ contains the loop $(\mathfrak{l}_0, \ldots, \mathfrak{l}_n)$ for some $n \geq 1$. Let $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ be a partition of $\mathfrak{R}$ such that $\mathfrak{R}_1 \overset{\text{def}}{=} \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ and $\mathfrak{R}_2 \overset{\text{def}}{=} \mathfrak{R} \setminus \mathfrak{R}_1$. Given $\mathfrak{l} \in \mathbb{N}$, we have $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l} \models \top$ (obviously), $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l} \models \neg\mathtt{emp}$ ($n \geq 1$, so $\mathfrak{R}_1$ is non-empty) and for all $\mathfrak{l}' \in \mathbb{N}$, $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \Box\Diamond\top$. Indeed, either $\mathfrak{l}' \not\in \{\mathfrak{l}_0, \ldots, \mathfrak{l}_{n-1}\}$ and therefore trivially $\mathfrak{R}_1(\mathfrak{l}') = \emptyset$ and $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \Box\Diamond\top$ or $\mathfrak{l}' = \mathfrak{l}_i$ for some $i \in [0, n-1]$ and $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l}' \models \Box\Diamond\top$ as there are unique $\mathfrak{l}^{\dagger}$ and $\mathfrak{l}^{\dagger\dagger}$ (possibly equal) such that $\mathfrak{R}_1(\mathfrak{l}') = \{\mathfrak{l}^{\dagger}\}$ and $\mathfrak{R}_1(\mathfrak{l}^{\dagger}) = \{\mathfrak{l}^{\dagger\dagger}\}$. Consequently, $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l} \models [\mathrm{U}]\Box\Diamond\top \wedge \neg\mathtt{emp}$. Hence, $\mathfrak{M} \models \exists\mathsf{Loop}$.

Now, suppose that $\mathfrak{M} \models \exists\mathsf{Loop}$, i.e. for all $\mathfrak{l} \in \mathbb{N}$, we have $\mathfrak{M}, \mathfrak{l} \models \exists\mathsf{Loop}$. Let $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ be a partition of $\mathfrak{R}$ such that $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l} \models [\mathrm{U}]\Box\Diamond\top \wedge \neg\mathtt{emp}$. So, $\mathfrak{R}_1$ is non-empty and for all $\mathfrak{l}' \in \mathbb{N}$, if $\mathfrak{R}_1(\mathfrak{l}') = \{\mathfrak{l}^{\dagger}\}$ for some location $\mathfrak{l}^{\dagger}$, then $\mathfrak{R}_1(\mathfrak{l}^{\dagger})$ is non-empty too. As $\mathfrak{R}_1$ is finite (and therefore cannot contain an infinite linear subrelation), $\mathfrak{R}_1$ is a finite collection of lassos ending by a non-empty loop and therefore contains at least one loop. As $\mathfrak{R}_1 \subseteq \mathfrak{R}$, any loop in $\mathfrak{R}_1$ is also a loop in $\mathfrak{R}$ and therefore, $\mathfrak{R}$ contains at least one loop (but it may also contain other parts that cannot be part of some loop). □

Let us consider the formulae below (whose semantics is given in Lemma 28).

$$\begin{array}{lcl}
\mathsf{PRoot} & \overset{\text{def}}{=} & \Diamond\Box\bot \\
\mathsf{UniqTreePRoot} & \overset{\text{def}}{=} & \neg\exists\mathsf{Loop} \wedge ((\neg(\neg\mathtt{emp} * \neg\mathtt{emp})) \vee \langle\, !\, \rangle\mathsf{PRoot}) \\
\mathsf{Leaf} & \overset{\text{def}}{=} & (\Diamond\top \wedge \mathtt{size} = 1) \vee \\
 & & (\Diamond\top \wedge \neg\mathsf{PRoot} \wedge ((\mathtt{size} = 1 \wedge \Diamond\top) * \mathsf{UniqTreePRoot})).
\end{array}$$

Lemma 28. *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.*

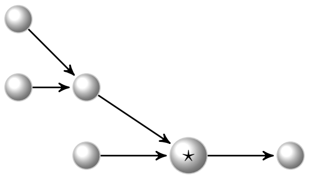


Figure 5: A tree whose root has a unique child

In Lemma 28(II), $\mathfrak{R}$ is a tree for which the root has a unique child (pre-root) as shown in Figure 5. However, note that the structure is not necessarily linear. The proof of Lemma 28 is rather tedious and is intrinsically related to the definition of the formulae.

Proof. (I) Obvious.
(II) First, suppose that $\mathfrak{M}, \mathfrak{l} \models \mathsf{UniqTreePRoot}$, i.e. for all locations $\mathfrak{l} \in \mathbb{N}$, we have $\mathfrak{M}, \mathfrak{l} \models \neg\exists\mathsf{Loop} \wedge ((\neg(\neg\mathtt{emp} * \neg\mathtt{emp})) \vee \langle\,!\,\rangle\mathsf{PRoot})$. Indeed, $\mathsf{UniqTreePRoot}$ is obviously a global formula. Consequently, $\mathfrak{M} \models \neg\exists\mathsf{Loop}$ and by Lemma 27, $\mathfrak{M}$ is loop-free. Now suppose that $\mathfrak{R}$ has at least two edges (otherwise it is obvious that either $\mathfrak{R}$ is empty or ($\mathfrak{M}$ has at most one MCC and a unique pre-root)). As a consequence, $\mathfrak{M} \models \langle\,!\,\rangle\mathsf{PRoot}$ and by (I), $\mathfrak{M}$ has a unique pre-root. *Ad absurdum*, suppose that $\mathfrak{M}$ has not at most one MCC. This means that there are $\mathfrak{l}_1$ and $\mathfrak{l}_2$ such that $\mathfrak{R}(\mathfrak{l}_1)$ and $\mathfrak{R}(\mathfrak{l}_2)$ are non-empty and $\mathfrak{R}^+(\mathfrak{l}_1) \cap \mathfrak{R}^+(\mathfrak{l}_2) = \emptyset$, where $\mathfrak{R}^+$ is the transitive closure of $\mathfrak{R}$. As $\mathfrak{M}$ is loop-free, there are distinct locations $\mathfrak{l}_1^+$ and $\mathfrak{l}_2^+$ such that $\mathfrak{l}_1^+ \in \mathfrak{R}^+(\mathfrak{l}_1)$, $\mathfrak{R}(\mathfrak{l}_1^+) = \emptyset$, $\mathfrak{l}_2^+ \in \mathfrak{R}^+(\mathfrak{l}_2)$ and $\mathfrak{R}(\mathfrak{l}_2^+) = \emptyset$. As $\mathfrak{l}_1^+$ and $\mathfrak{l}_2^+$ are distinct and $\mathfrak{R}$ is functional, we have that there are distinct locations $\mathfrak{l}_1'$ and $\mathfrak{l}_2'$ such that $(\mathfrak{l}_1', \mathfrak{l}_1^+), (\mathfrak{l}_2', \mathfrak{l}_2^+) \in \mathfrak{R}$. So, $\mathfrak{l}_1'$ and $\mathfrak{l}_2'$ are distinct pre-roots, which leads to a contradiction.
Now suppose that $\mathfrak{M}$ is loop-free and either $\mathfrak{R}$ is empty or ($\mathfrak{M}$ has at most one MCC and a unique pre-root). By Lemma 27, $\mathfrak{M} \models \neg\exists\mathsf{Loop}$. If $\mathfrak{R}$ has at most one edge, then $\mathfrak{M} \models ((\neg(\neg\mathtt{emp} * \neg\mathtt{emp})) \vee \langle\,!\,\rangle\mathsf{PRoot})$ because the first disjunct holds. Now, suppose that $\mathfrak{R}$ has at least two edges. By assumption, $\mathfrak{M}$ has a unique pre-root and therefore $\mathfrak{M} \models \langle\,!\,\rangle\mathsf{PRoot}$. Consequently, we conclude that $\mathfrak{M} \models \mathsf{UniqTreePRoot}$.
(III) Let us assume that $\mathfrak{M} \models \mathsf{UniqTreePRoot}$. By (II), $\mathfrak{M}$ is loop-free. First, we suppose that $\mathfrak{M}, \mathfrak{l} \models \mathsf{Leaf}$, i.e.

$$\mathfrak{M}, \mathfrak{l} \models (\Diamond\top \wedge \mathtt{size} = 1) \vee (\Diamond\top \wedge \neg\mathsf{PRoot} \wedge ((\mathtt{size} = 1 \wedge \Diamond\top) * \mathsf{UniqTreePRoot})).$$

If $\mathfrak{M}, \mathfrak{l} \models (\Diamond\top \wedge \mathtt{size} = 1)$, then $\mathfrak{R}$ contains a unique edge and $\mathfrak{R}(\mathfrak{l})$ is non-empty, which entails that $\mathfrak{l}$ is a leaf (as $\mathfrak{M}$ is loop-free). Otherwise, suppose that $\mathfrak{M}, \mathfrak{l} \models (\Diamond\top \wedge \neg\mathsf{PRoot} \wedge ((\mathtt{size} = 1 \wedge \Diamond\top) * \mathsf{UniqTreePRoot}))$ and let $\mathfrak{R}(\mathfrak{l}) = \{\mathfrak{l}'\}$ for some location $\mathfrak{l}'$ (different from $\mathfrak{l}$ as $\mathfrak{M}$ is loop-free). By (I), $\mathfrak{l}$ is not a pre-root, $\mathfrak{R}$ contains at least two edges and $\{\{(\mathfrak{l}, \mathfrak{l}')\}, \mathfrak{R} \setminus \{(\mathfrak{l}, \mathfrak{l}')\}\}$ is the unique partition of $\mathfrak{R}$ such that $\langle \mathbb{N}, \{(\mathfrak{l}, \mathfrak{l}')\}, \mathfrak{V}\rangle, \mathfrak{l} \models (\mathtt{size} = 1 \wedge \Diamond\top)$. Consequently,

$$\mathfrak{M}' = \langle \mathbb{N}, \mathfrak{R} \setminus \{(\mathfrak{l}, \mathfrak{l}')\}, \mathfrak{V}\rangle, \mathfrak{l} \models \mathsf{UniqTreePRoot}.$$

By (II), $\mathfrak{M}'$ is loop-free (obviously since $\mathfrak{M}$ were already loop-free by assumption) and either $\mathfrak{R}' = \mathfrak{R} \setminus \{(\mathfrak{l}, \mathfrak{l}')\}$ is empty or ($\mathfrak{M}'$ has at most one MCC and a unique pre-root). As $\mathfrak{R} \setminus \{(\mathfrak{l}, \mathfrak{l}')\}$ is non-empty, we have that $\mathfrak{M}'$ has at most one MCC and a unique pre-root. *Ad absurdum*, suppose that $\mathfrak{l}$ is not a leaf in $\mathfrak{M}$. This implies that there is a location $\mathfrak{l}^\dagger \in \mathbb{N}$ such that $(\mathfrak{l}^\dagger, \mathfrak{l}) \in \mathfrak{R}$. Since $\mathfrak{l}$ is not a pre-root in $\mathfrak{M}$, there is also a location $\mathfrak{l}^{\dagger\dagger}$ such that $(\mathfrak{l}', \mathfrak{l}^{\dagger\dagger}) \in \mathfrak{R}$. Moreover, since $\mathfrak{M}$ is loop-free, $\mathfrak{l} \notin \mathfrak{R}^*(\mathfrak{l}')$ and a fortiori, $\mathfrak{l} \notin \mathfrak{R}'^*(\mathfrak{l}')$. Similarly, since $\mathfrak{M}$ is loop-free, all locations among $\mathfrak{l}^\dagger$, $\mathfrak{l}$, $\mathfrak{l}'$ and $\mathfrak{l}^{\dagger\dagger}$ are distinct. So, $\mathfrak{R}'(\mathfrak{l}^\dagger) = \{\mathfrak{l}\}$, $\mathfrak{R}'(\mathfrak{l})$ is empty and $\mathfrak{R}'^*(\mathfrak{l}') \cap \{\mathfrak{l}^\dagger, \mathfrak{l}\} = \emptyset$, so $\mathfrak{M}'$ has not at most one MCC, which leads to a contradiction.
Now suppose that $\mathfrak{l}$ is leaf. Let $\mathfrak{R}(\mathfrak{l}) = \{\mathfrak{l}'\}$ for some location $\mathfrak{l}'$. If $\mathfrak{R}$ has a unique edge, then $\mathfrak{M}, \mathfrak{l} \models (\Diamond\top \wedge \mathtt{size} = 1)$ and therefore $\mathfrak{M}, \mathfrak{l} \models \mathsf{Leaf}$. Otherwise, $\mathfrak{R}$ has at least two edges and remember that $\mathfrak{M} \models \mathsf{UniqTreePRoot}$ by assumption. Since $\mathfrak{M}$ has at most one MCC and a unique pre-root, $\mathfrak{l}$ cannot be a pre-root (otherwise, the existence of at least two edges implies the existence of another pre-root). So, $\mathfrak{M}, \mathfrak{l} \models \Diamond\top \wedge \neg\mathsf{PRoot}$. Let $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ be the unique partition of $\mathfrak{R}$ with $\mathfrak{R}_1 = \{(\mathfrak{l}, \mathfrak{l}')\}$. Obviously, $\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V}\rangle, \mathfrak{l} \models (\mathtt{size} = 1 \wedge \Diamond\top)$ and because $\mathfrak{l}$ is a leaf of $\mathfrak{M}$, the model $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V}\rangle$ is loop-free (inherited from $\mathfrak{M}$), has at most one MCC and has a unique pre-root (actually the same pre-root of $\mathfrak{M}$). By (II), we get $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V}\rangle, \mathfrak{l} \models \mathsf{UniqTreePRoot}$. So, $\mathfrak{M}, \mathfrak{l} \models (\mathtt{size} = 1 \wedge \Diamond\top) * \mathsf{UniqTreePRoot}$, which is all what we need to conclude that $\mathfrak{M}, \mathfrak{l} \models \mathsf{Leaf}$. □

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 29. *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.*

Proof. First, suppose that $\mathfrak{M}$ is a linear model. If $\mathfrak{M}$ has an empty relation, $\mathfrak{M}, \mathfrak{l} \models \mathtt{emp}$ for all locations $\mathfrak{l}$, and therefore $\mathfrak{M} \models \phi_{\exists\mathtt{ls}}$. Otherwise, by Lemma 26, $\mathfrak{M}$ is loop-free and has a unique leaf. So, $\mathfrak{M}$ has at most one MCC and a unique pre-root and therefore by Lemma 28(II), we have $\mathfrak{M}, \mathfrak{l} \models \mathsf{UniqTreePRoot}$ for all locations $\mathfrak{l}$. By Lemma 28(III), for all locations $\mathfrak{l}$, we have $\mathfrak{M}, \mathfrak{l} \models \langle\,!\,\rangle\mathsf{Leaf}$. Consequently, $\mathfrak{M} \models \phi_{\exists\mathtt{ls}}$.

Conversely, suppose that $\mathfrak{M} \models \phi_{\exists\mathtt{ls}}$. If $\mathfrak{M} \models \mathtt{emp}$, then obviously, $\mathfrak{M}$ is linear. Otherwise, suppose that $\mathfrak{M} \models \mathsf{UniqTreePRoot} \wedge \langle\,!\,\rangle\mathsf{Leaf}$. By Lemma 28(II), $\mathfrak{M}$ has at most one MCC and a unique pre-root. As $\mathfrak{M} \models \langle\,!\,\rangle\mathsf{Leaf}$, $\mathfrak{M}$ has a unique leaf by Lemma 28(III), and therefore $\mathfrak{M}$ is a non-empty linear structure. □

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 MSL$(*, \Diamond, \langle\neq\rangle)$ is Tower-hard by reduction from the nonemptiness problem for star-free expressions [54, 62]. The proof takes advantage of Theorem 29 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 [31, Section 3] as it is essential to be able to encode finite words. Instead of reducing the satisfiability problem for Propositional Interval Temporal Logic [57] as done in [31, Section 3], we define a reduction from the nonemptiness problem for star-free expressions.

A *star-free expression* e over some finite 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 [54, 64] and refined to Tower-completeness in [62].

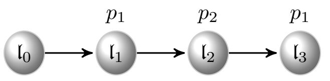
Given a finite alphabet $\Sigma = \{a_1, \ldots, a_\alpha\}$, we use the models encoding finite words thanks to the formula $\phi_{\exists\mathtt{ls}}$ and furthermore, we require that $[\mathrm{U}] \bigvee_i \mathsf{a}_i$ where

$$\mathsf{a}_i \stackrel{\text{def}}{=} p_i \wedge \bigwedge_{j \in \{1,\ldots,\alpha\} \setminus \{i\}} \neg p_j.$$

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 $k \in [1, \alpha]$, $\mathfrak{V}(p_k) = \{\mathfrak{l}_j \mid j \in [1, n],\ i_j = k\}$.

By way of example,

$a_1\ a_2\ a_1 \ \rhd$  $, \mathfrak{l}_0$

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 30. *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 $\mathfrak{w}_2 \rhd \mathfrak{M}_2, \mathfrak{l}'$.*

Proof. Suppose that $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V}\rangle$ with $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$. We perform a case analysis depending whether $\mathfrak{w}_i = \varepsilon$.

- If $\mathfrak{w} = \varepsilon$, then $\mathfrak{M}_1 = \mathfrak{M}$, $\mathfrak{M}_2 = \mathfrak{M}$ and $\mathfrak{l}' = \mathfrak{l}$.
- If $\mathfrak{w}_1 = \varepsilon$ and $\mathfrak{w}_2 \neq \varepsilon$, then $\mathfrak{M}_1 = \mathfrak{M}_\emptyset$, $\mathfrak{M}_2 = \mathfrak{M}$ and $\mathfrak{l}' = \mathfrak{l}$ with $\mathfrak{M}_\emptyset = \langle \mathbb{N}, \emptyset, \mathfrak{V}\rangle$.
- If $\mathfrak{w}_2 = \varepsilon$ and $\mathfrak{w}_1 \neq \varepsilon$, then $\mathfrak{M}_1 = \mathfrak{M}$, $\mathfrak{M}_2 = \mathfrak{M}_\emptyset$ and $\mathfrak{l}' = \mathfrak{l}$.
- Suppose that $\mathfrak{w}_1 \neq \varepsilon$ and $\mathfrak{w}_2 \neq \varepsilon$. So, $\mathfrak{w} = b_1 \cdots b_n$ with $n \geq 2$ and as $\mathfrak{M}$ is linear, we have $\mathfrak{R} = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ and for all $i \neq j \in [0, n]$, we have $\mathfrak{l}_i \neq \mathfrak{l}_j$. Suppose that $\mathfrak{w}_1 = b_1 \cdots b_{n'}$ for some $n' < n$. Let $\mathfrak{R}_1 = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n'-1}, \mathfrak{l}_{n'})\}$ and $\mathfrak{R}_2 = \{(\mathfrak{l}_{n'}, \mathfrak{l}_{n'+1}), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$. We set $\mathfrak{M}_1 = \langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V}\rangle$, $\mathfrak{M}_2 = \langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V}\rangle$ and $\mathfrak{l}' = \mathfrak{l}_{n'}$.

For the four cases, it is easy to check 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 32. It states that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$ and those models are linear entail that there is necessarily a "clean cut".

Lemma 31. *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 $\mathfrak{w}_2 \rhd \mathfrak{M}_2, \mathfrak{l}'$.*

Note that in $\mathfrak{w}_2 \rhd \mathfrak{M}_2, \mathfrak{l}'$, the location $\mathfrak{l}'$ is involved.

Proof. Suppose that $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$ with $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$ and both $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are linear. Let $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V}\rangle$, $\mathfrak{M}_1 = \langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V}\rangle$, $\mathfrak{M}_2 = \langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V}\rangle$ and $\mathfrak{R} = \mathfrak{R}_1 \uplus \mathfrak{R}_2$. We perform a case analysis depending whether $\mathfrak{M}_i$ has an empty relation. When $\mathfrak{R}_1 = \emptyset$ or $\mathfrak{R}_2 = \emptyset$, the definition of $\mathfrak{w}_1$ and $\mathfrak{w}_2$ is easy. Let us consider the case $\mathfrak{R}_1 \neq \emptyset$ and $\mathfrak{R}_2 \neq \emptyset$. Suppose that $\mathfrak{R} = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ for some $n \geq 2$ and for all $i \neq j \in [0, n]$, we have $\mathfrak{l}_i \neq \mathfrak{l}_j$. Moreover, suppose that the word $\mathfrak{w}$ is equal to $b_1 \cdots b_n$. Let $\{X_1, X_2\}$ be a partition of $[0, n-1]$ such that both X_1 and X_2 are non-empty and for all $i \in \{1, 2\}$, we have $\mathfrak{R}_i = \{(\mathfrak{l}_j, \mathfrak{l}_{j+1}) \mid j \in X_i\}$. The only case for which $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$ and both $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are linear is when there is $n' \in [0, n-2]$ such that $X_1 = [0, n']$ and $X_2 = [n'+1, n-1]$. Indeed, if X_i is made of at least two disjoint nonempty intervals, then $\mathfrak{M}_i$ cannot be linear. We set $\mathfrak{w}_1 = b_1 \cdots b_{n'+1}$, $\mathfrak{w}_2 = b_{n'+2} \cdots b_n$ and $\mathfrak{l}' = \mathfrak{l}_{n'+1}$. It is easy to check 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 star-free expression e is translated as

$$T(e) \stackrel{\text{def}}{=} ([\mathrm{U}] \textstyle\bigvee_i \mathtt{a}_i) \wedge \phi_{\exists 1 \mathtt{s}} \wedge (\mathtt{emp} \wedge t(e)) \vee (\neg \mathtt{emp} \wedge \mathsf{Leaf} \wedge t(e)),$$

where $t(\cdot)$ is recursively defined (the evaluation is performed at leaf locations, if any). The four disjuncts in $t(e_1e_2)$ below correspond to cases depending on the emptiness of subwords.

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

$$\psi_1 \stackrel{\text{def}}{=} \mathtt{emp} \wedge t(e_1) \wedge t(e_2) \quad \psi_2 \stackrel{\text{def}}{=} (t(e_1) \wedge \mathtt{emp}) * t(e_2) \quad \psi_3 \stackrel{\text{def}}{=} t(e_1) * (t(e_2) \wedge \mathtt{emp})$$
$$\psi_4 \stackrel{\text{def}}{=} (\phi_{\exists 1 \mathtt{s}} \wedge \neg \mathtt{emp} \wedge t(e_1)) * (\phi_{\exists 1 \mathtt{s}} \wedge \neg \mathtt{emp} \wedge \langle \mathrm{U}\rangle(\mathsf{Leaf} \wedge t(e_2))).$$

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

Lemma 32. *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 $\mathfrak{M}, \mathfrak{l} \models t(e)$.*

Proof. The proof is by structural induction on e. Let us start by treating the base cases.

$e = \varepsilon$. Suppose that $\mathfrak{w} \in \mathrm{L}(e)$. Obviously, $\mathfrak{w} = \varepsilon$ and $\mathfrak{M}$ has an empty accessibility relation by definition of $\triangleright$. So, $\mathfrak{M}, \mathfrak{l} \models \texttt{emp}$ and therefore $\mathfrak{M}, \mathfrak{l} \models t(e)$ by definition of t.

Now suppose that $\mathfrak{M}, \mathfrak{l} \models t(e)$. As $t(\varepsilon) = \texttt{emp}$, this implies that $\mathfrak{M}$ has an empty accessibility relation and therefore $\mathfrak{w} = \varepsilon$ and $\mathfrak{w} \in \mathrm{L}(e)$.

$e = a_i$. Suppose that $\mathfrak{w} \in \mathrm{L}(e)$. Obviously, $\mathfrak{w} = a_i$ and by definition of $\triangleright$, $\mathfrak{M}$ is of the form $\langle \mathbb{N}, \{(\mathfrak{l}, \mathfrak{l}')\}, \mathfrak{V}\rangle$ such that $\mathfrak{l}' \in \mathfrak{V}(p_i)$ and for all $j \neq i \in [1, \alpha]$, we have $\mathfrak{l}' \notin \mathfrak{V}(p_j)$. So, obviously, $\mathfrak{M}, \mathfrak{l} \models (\Diamond \mathtt{a}_i) \wedge \texttt{size} = 1$ with $\mathtt{a}_i = (p_i \wedge \bigwedge_{j \neq i} \neg p_j)$. By definition of t, we get $\mathfrak{M}, \mathfrak{l} \models t(e)$.

Now suppose that $\mathfrak{M}, \mathfrak{l} \models t(e)$. By definition of t, this means that $\mathfrak{M}, \mathfrak{l} \models (\Diamond \mathtt{a}_i) \wedge \texttt{size} = 1$. So, $\mathfrak{M}$ is of the form $\langle \mathbb{N}, \{(\mathfrak{l}, \mathfrak{l}')\}, \mathfrak{V}\rangle$ such that $\mathfrak{l}' \in \mathfrak{V}(p_i)$ and for all $j \neq i \in [1, \alpha]$, we have $\mathfrak{l}' \notin \mathfrak{V}(p_j)$. By definition of $\triangleright$, this means that $\mathfrak{w} = a_i$ and therefore $\mathfrak{w} \in \mathrm{L}(e)$.

In the induction step, let us treat the following cases.

$e = e_1 \cup e_2$. Suppose that $\mathfrak{w} \in \mathrm{L}(e)$. So, there is $i \in \{1, 2\}$ such that $\mathfrak{w} \in \mathrm{L}(e_i)$. By the induction hypothesis, we have $\mathfrak{M}, \mathfrak{l} \models t(e_i)$ and therefore $\mathfrak{M}, \mathfrak{l} \models t(e_1) \vee t(e_2)$. Hence, $\mathfrak{M}, \mathfrak{l} \models t(e)$ by definition of t.

Now suppose that $\mathfrak{M}, \mathfrak{l} \models t(e)$. By definition of t, there is $i \in \{1, 2\}$ such that $\mathfrak{M}, \mathfrak{l} \models t(e_i)$. By the induction hypothesis, we have $\mathfrak{w} \in \mathrm{L}(e_i)$. By definition of $\mathrm{L}(e)$, we get $\mathfrak{w} \in \mathrm{L}(e)$.

$e = \sim e'$. $\mathfrak{w} \in \mathrm{L}(e)$, iff (by definition of $\mathrm{L}(e)$), $\mathfrak{w} \notin \mathrm{L}(e')$ iff (by the induction hypothesis) $\mathfrak{M}, \mathfrak{l} \not\models t(e')$, iff (by definition of $\models$) $\mathfrak{M}, \mathfrak{l} \models \neg t(e')$, iff $\mathfrak{M}, \mathfrak{l} \models t(e)$ (by definition of t).

$e = e_1 e_2$. Suppose that $\mathfrak{w} \in \mathrm{L}(e)$. There are $\mathfrak{w}_1 \in \mathrm{L}(e_1)$ and $\mathfrak{w}_2 \in \mathrm{L}(e_2)$ such that $\mathfrak{w} = \mathfrak{w}_1 \mathfrak{w}_2$. By Lemma 30, there exist linear models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ and $\mathfrak{l}'$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathfrak{w}_1 \triangleright \mathfrak{M}_1, \mathfrak{l}$ and $\mathfrak{w}_2 \triangleright \mathfrak{M}_2, \mathfrak{l}'$.

- If $\mathfrak{w} = \varepsilon$ (and therefore $\mathfrak{M}_1 = \mathfrak{M}_2 = \mathfrak{M}$ and $\mathfrak{w}_1 = \mathfrak{w}_2 = \varepsilon$), by the induction hypothesis, $\mathfrak{M}_1, \mathfrak{l} \models t(e_1)$ and $\mathfrak{M}_2, \mathfrak{l} \models t(e_2)$. Consequently, $\mathfrak{M}, \mathfrak{l} \models \texttt{emp} \wedge t(e_1) \wedge t(e_2)$ (corresponding to the satisfaction of ψ_1) and therefore $\mathfrak{M}, \mathfrak{l} \models t(e)$.
- If $\mathfrak{w}_1 = \varepsilon$ and $\mathfrak{w}_2 \neq \varepsilon$ (and therefore $\mathfrak{M}_2 = \mathfrak{M}$ and $\mathfrak{M}_1 = \langle \mathbb{N}, \emptyset, \mathfrak{V}\rangle$), by the induction hypothesis $\mathfrak{M}_2, \mathfrak{l} \models t(e_2)$ and $\mathfrak{M}_1, \mathfrak{l} \models t(e_1)$ as $\mathfrak{w}_1 \triangleright \mathfrak{M}_1, \mathfrak{l}$. So $\mathfrak{M}, \mathfrak{l} \models (t(e_1) \wedge \texttt{emp}) * t(e_2)$ (corresponding to the satisfaction of ψ_2) and therefore $\mathfrak{M}, \mathfrak{l} \models t(e)$.
- The case $\mathfrak{w}_1 \neq \varepsilon$ and $\mathfrak{w}_2 = \varepsilon$ is dealt with similarly.
- Suppose that $\mathfrak{w}_1 \neq \varepsilon$ and $\mathfrak{w}_2 \neq \varepsilon$. By the induction hypothesis, we have $\mathfrak{M}_1, \mathfrak{l} \models t(e_1)$, $\mathfrak{M}_1, \mathfrak{l} \models \phi_{\exists \mathtt{ls}} \wedge \neg\texttt{emp}$ as $\mathfrak{M}_1$ is linear and non-empty. Similarly, $\mathfrak{M}_2, \mathfrak{l}' \models \phi_{\exists \mathtt{ls}} \wedge \neg\texttt{emp}$. As $\mathfrak{w}_2 \triangleright \mathfrak{M}_2, \mathfrak{l}'$, by the induction hypothesis, we get $\mathfrak{M}_2, \mathfrak{l}' \models t(e_2)$. Moreover, by definition of $\triangleright$, $\mathfrak{l}'$ is the unique leaf of $\mathfrak{R}_2$ and therefore $\mathfrak{M}_2, \mathfrak{l}' \models \texttt{Leaf}$. Consequently, $\mathfrak{M}, \mathfrak{l} \models (\phi_{\exists \mathtt{ls}} \wedge \neg\texttt{emp} \wedge t(e_1)) * (\phi_{\exists \mathtt{ls}} \wedge \neg\texttt{emp} \wedge \langle \mathrm{U}\rangle(\texttt{Leaf} \wedge t(e_2)))$ (corresponding to the satisfaction of ψ_4) and therefore $\mathfrak{M}, \mathfrak{l} \models t(e)$.

Now suppose that $\mathfrak{M}, \mathfrak{l} \models t(e)$. Let us make a case analysis depending on which ψ_i holds true.

$\mathfrak{M}, \mathfrak{l} \models \psi_1$. By definition of t, we have $\mathfrak{M}, \mathfrak{l} \models \texttt{emp} \wedge t(e_1) \wedge t(e_2)$ and therefore $\mathfrak{w} = \varepsilon$. So $\mathfrak{M}, \mathfrak{l} \models t(e_1)$ and $\mathfrak{M}, \mathfrak{l} \models t(e_2)$. By the induction hypothesis, $\varepsilon \in \mathrm{L}(e_1)$ and $\varepsilon \in \mathrm{L}(e_2)$, so $\mathfrak{w} \in \mathrm{L}(e)$.

$\mathfrak{M}, \mathfrak{l} \models \psi_2$. By definition of t, we have $\mathfrak{M}, \mathfrak{l} \models (t(e_1) \wedge \texttt{emp}) * t(e_2)$. Let $\mathfrak{M}_2 = \mathfrak{M}$ and $\mathfrak{M}_1 = \langle \mathbb{N}, \emptyset, \mathfrak{V}\rangle$. So, $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathfrak{M}_2, \mathfrak{l} \models t(e_2)$, $\mathfrak{w} \triangleright \mathfrak{M}_2, \mathfrak{l}$, and $\mathfrak{M}_1, \mathfrak{l} \models t(e_1) \wedge \texttt{emp}$ and $\varepsilon \triangleright \mathfrak{M}_1, \mathfrak{l}$. By the induction hypothesis, $\mathfrak{w} \in \mathrm{L}(e_2)$ and $\varepsilon \in \mathrm{L}(e_1)$. So, $\mathfrak{w} \in \mathrm{L}(e)$.

$\mathfrak{M}, \mathfrak{l} \models \psi_3$. Similar to the previous case.

$\mathfrak{M}, \mathfrak{l} \models \psi_4$. By definition of t, we have $\mathfrak{M}, \mathfrak{l} \models (\phi_{\exists \texttt{ls}} \wedge \neg \texttt{emp} \wedge t(e_1)) * (\phi_{\exists \texttt{ls}} \wedge \neg \texttt{emp} \wedge \langle \mathrm{U} \rangle (\mathsf{Leaf} \wedge t(e_2)))$. By definition of $\models$, there are linear non-empty models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that $\mathfrak{M} = \mathfrak{M}_1 \uplus \mathfrak{M}_2$, $\mathfrak{M}_1, \mathfrak{l} \models t(e_1)$ and $\mathfrak{M}_2, \mathfrak{l}' \models \mathsf{Leaf} \wedge t(e_2)$ for some location $\mathfrak{l}'$. By Lemma 31, there are $\mathfrak{w}_1, \mathfrak{w}_2 \in \Sigma^*$ such that $\mathfrak{w} = \mathfrak{w}_1 \mathfrak{w}_2$, $\mathfrak{w}_1 \rhd \mathfrak{M}_1, \mathfrak{l}$ and $\mathfrak{w}_2 \rhd \mathfrak{M}_2, \mathfrak{l}'$. By the induction hypothesis, $\mathfrak{w}_1 \in \mathrm{L}(e_1)$ and $\mathfrak{w}_2 \in \mathrm{L}(e_2)$. So, $\mathfrak{w} \in \mathrm{L}(e)$. □

As a consequence,

Lemma 33. *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.*

Proof. ($\Rightarrow$) Suppose that $\mathfrak{w} \in \mathrm{L}(e)$. One can easily build a linear model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and a location $\mathfrak{l}$ such that $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$. It is also easy to enforce that $\mathfrak{M} \models [\mathrm{U}] \bigvee_i \mathsf{a}_i$. So, $\mathfrak{M}, \mathfrak{l} \models ([\mathrm{U}] \bigvee_i \mathsf{a}_i) \wedge \phi_{\exists \texttt{ls}}$. If $\mathfrak{w} = \varepsilon$, then necessarily, $\mathfrak{R} = \emptyset$ and therefore $\mathfrak{M}, \mathfrak{l} \models \texttt{emp}$. By Lemma 32, we also get that $\mathfrak{M}, \mathfrak{l} \models t(e)$. This concludes that $\mathfrak{M}, \mathfrak{l} \models T(e)$. If $\mathfrak{w} \neq \varepsilon$, then necessarily, $\mathfrak{R} \neq \emptyset$ and therefore $\mathfrak{M}, \mathfrak{l} \models \neg\texttt{emp}$. Again, by Lemma 32, we also get that $\mathfrak{M}, \mathfrak{l} \models t(e)$. This concludes that $\mathfrak{M}, \mathfrak{l} \models T(e)$.

($\Leftarrow$) Suppose that $\mathfrak{M}, \mathfrak{l} \models T(e)$ for some model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and $\mathfrak{l} \in \mathbb{N}$. Note that by definition of $T(e)$, $\mathfrak{M}$ is linear. If $\mathfrak{R} = \emptyset$, then $\varepsilon \rhd \mathfrak{M}, \mathfrak{l}$ and necessarily $\mathfrak{M}, \mathfrak{l} \models \texttt{emp} \wedge t(e)$. By Lemma 32, we get that $\varepsilon \in \mathrm{L}(e)$. If $\mathfrak{R} = \{(\mathfrak{l}_0, \mathfrak{l}_1), \ldots, (\mathfrak{l}_{n-1}, \mathfrak{l}_n)\}$ for some $n \geq 1$ ($\mathfrak{l}_0 = \mathfrak{l}$), then $\mathfrak{w} \rhd \mathfrak{M}, \mathfrak{l}$ for some word $\mathfrak{w} = b_1 \cdots b_n$. Necessarily $\mathfrak{M}, \mathfrak{l} \models \neg\texttt{emp} \wedge \mathsf{Leaf} \wedge t(e)$. By Lemma 32, we get that $b_1 \cdots b_n \in \mathrm{L}(e)$. Consequently, $\mathrm{L}(e)$ is non-empty. □

Finally, we get the TOWER-completeness.

Theorem 34. *The satisfiability problem for $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ is TOWER-complete.*

6 When the magic wand strikes back

In this section, we show that the satisfiability problem for MSL is actually undecidable by taking advantage of previous results. Moreover, we conclude the section by showing other undecidability results, such as for $\mathrm{MSL}^g(*, \Diamond)$ by designing a reduction from the global sabotage modal logic.

All the previous complexity results, involving classes such as NP, PSPACE and TOWER, deal with fragments of MSL that are $-\!\!*$-free and MSL is precisely equal to $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ augmented with the separating implication $-\!\!*$. It is well-known that adding the separating connective $-\!\!*$ can dramatically augment the expressive power or the complexity, see e.g. [15]. Below, the expressive strength of $-\!\!*$ is again illustrated, via a reduction from propositional separation logic augmented with the list segment predicate `ls` [35]. By contrast, it is known that the modal logic for heaps MLH restricted to $*$ is decidable [31], but it is open whether the addition of $-\!\!*$ leads to undecidability.

6.1 Undecidability of MSL

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 [47] 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 $\phi_{\exists \texttt{ls}}$ can enforce a linear structure 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 $\mathrm{SL}(*, -\!\!*, \texttt{ls})$, recently shown undecidable in [35] (based on [15]).

Notice that for the translation of the formulae from $\mathrm{SL}(*, -\!\!*, \texttt{ls})$, the most complex part is the encoding of the atomic formulae $\texttt{ls}(\mathtt{x}, \mathtt{y})$. 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 separating 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 35. *Let ϕ be an $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ formula. ϕ is satisfiable iff $t(\phi)$ is satisfiable in* MSL.

Proof. Let $\mathtt{N}$ be a finite set of propositional variables playing the role of nominals (also understood as program variables). Given a model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and a memory state $(\mathfrak{s}, \mathfrak{h})$, we write $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$ iff the conditions below hold:

1. $\mathfrak{R} \stackrel{\text{def}}{=} \{(\mathfrak{l}, \mathfrak{h}(\mathfrak{l})) \mid \mathfrak{l} \in \mathrm{dom}(\mathfrak{h})\}$,
2. for all $\mathtt{x} \in \mathtt{N}$, $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{s}(\mathtt{x})\}$.

Notice that every variable $\mathtt{x} \in \mathrm{PVAR}$ is a nominal in MSL.

Below, we show that for all formulae ϕ in $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ built over program variables in $\mathtt{N}$, and $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, we have $(\mathfrak{s}, \mathfrak{h}) \models \phi$ iff $\mathfrak{M} \models t(\phi)$. In that way, assuming that ϕ is satisfiable (say $(\mathfrak{s}, \mathfrak{h}) \models \phi$), the MSL model $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$ and $\mathfrak{l} \in \mathbb{N}$ such that $\mathfrak{R} \stackrel{\text{def}}{=} \{(\mathfrak{l}, \mathfrak{h}(\mathfrak{l})) \mid \mathfrak{l} \in \mathrm{dom}(\mathfrak{h})\}$ and for all $\mathtt{x} \in \mathrm{PVAR}$, $\mathfrak{V}(\mathtt{x}) \stackrel{\text{def}}{=} \{\mathfrak{s}(\mathtt{x})\}$, verify that $\mathfrak{M}, \mathfrak{l} \models t(\phi)$. Conversely, assuming that $\mathfrak{M}, \mathfrak{l} \models t(\phi)$, one can easily build $(\mathfrak{s}, \mathfrak{h})$ such that $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$ and $(\mathfrak{s}, \mathfrak{h}) \models \phi$.

Below, we assume that $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$. By structural induction, we show that $(\mathfrak{s}, \mathfrak{h}) \models \phi$ iff $\mathfrak{M} \models t(\phi)$ for all formulae ϕ in $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ built over program variables in $\mathtt{N}$.

$\phi = \mathtt{emp}$. $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{emp}$ iff (by $\models$) $\mathrm{dom}(\mathfrak{h}) = \emptyset$. As $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, $\mathfrak{R} = \emptyset$, then $\mathfrak{M} \models t(\mathtt{emp})$. Conversely, $\mathfrak{M}, \mathfrak{l} \models t(\mathtt{emp})$ iff (by definition of t) $\mathfrak{M}, \mathfrak{l} \models \mathtt{emp}$ iff (by $\models$) $\mathfrak{R} = \emptyset$. As $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, $\mathrm{dom}(\mathfrak{h}) = \emptyset$, and consequently $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{emp}$.

$\phi = (\mathtt{x} = \mathtt{y})$. $(\mathfrak{s}, \mathfrak{h}) \models (\mathtt{x} = \mathtt{y})$ iff (by $\models$) $\mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y})$. As $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, there is some $\mathfrak{l} \in \mathbb{N}$ such that $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{l}\} = \mathfrak{V}(\mathtt{y})$. Then we have $\mathfrak{M}, \mathfrak{l} \models (\mathtt{x} \wedge \mathtt{y})$, and by $\models$ we get $\mathfrak{M}, \mathfrak{l}' \models \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y})$ for all $\mathfrak{l}' \in \mathbb{N}$, whence $\mathfrak{M} \models t(\mathtt{x} = \mathtt{y})$. Conversely, $\mathfrak{M}, \mathfrak{l} \models \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y})$ iff there is $\mathfrak{l}' \in \mathbb{N}$ such that $\mathfrak{M}, \mathfrak{l}' \models \mathtt{x} \wedge \mathtt{y}$, i.e., $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{l}'\} = \mathfrak{V}(\mathtt{y})$ (it is a singleton since $\mathtt{x}, \mathtt{y} \in \mathrm{PVAR}$, therefore they are nominals). As $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, this implies that $\mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y}) = \mathfrak{l}'$, and therefore $(\mathfrak{s}, \mathfrak{h}) \models (\mathtt{x} = \mathtt{y})$.

$\phi = (\mathtt{x} \hookrightarrow \mathtt{y})$. $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{x} \hookrightarrow \mathtt{y}$ iff (by $\models$) $\mathfrak{s}(\mathtt{x}) \in \mathrm{dom}(\mathfrak{h})$ and $\mathfrak{h}(\mathfrak{s}(\mathtt{x})) = \mathfrak{s}(\mathtt{y})$. As $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, there exist $\mathfrak{l}', \mathfrak{l}'' \in \mathbb{N}$ such that $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{l}'\}$, $\mathfrak{V}(\mathtt{y}) = \{\mathfrak{l}''\}$, and $(\mathfrak{l}', \mathfrak{l}'') \in \mathfrak{R}$. By definition of $\models$, we have $\mathfrak{M}, \mathfrak{l}' \models \mathtt{x} \wedge \Diamond \mathtt{y}$, and by globality of $\langle \mathrm{U} \rangle$ we get $\mathfrak{M}, \mathfrak{l} \models \langle \mathrm{U} \rangle (\mathtt{x} \wedge \Diamond \mathtt{y})$ for all $\mathfrak{l} \in \mathbb{N}$, whence $\mathfrak{M} \models t(\mathtt{x} \hookrightarrow \mathtt{y})$. Conversely, $\mathfrak{M}, \mathfrak{l} \models \langle \mathrm{U} \rangle (\mathtt{x} \wedge \Diamond \mathtt{y})$ iff there exists $\mathfrak{l}' \in \mathbb{N}$ such that $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{l}'\}$, and there exists $\mathfrak{l}'' \in \mathbb{N}$ such that $(\mathfrak{l}', \mathfrak{l}'') \in \mathfrak{R}$, and $\mathfrak{V}(\mathtt{y}) = \{\mathfrak{l}''\}$. Then, as $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, we have $\mathfrak{s}(\mathtt{x}) = \mathfrak{l}'$, $\mathfrak{s}(\mathtt{y}) = \mathfrak{l}''$ and $\mathfrak{h}(\mathfrak{s}(\mathtt{x})) = \mathfrak{s}(\mathtt{y})$. Therefore, $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{x} \hookrightarrow \mathtt{y}$.

$\phi = \mathtt{ls}(\mathtt{x}, \mathtt{y})$. $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{ls}(\mathtt{x}, \mathtt{y})$ iff either

(i) $\mathrm{dom}(\mathfrak{h}) = \emptyset$ and $\mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y})$, or

(ii) $\mathfrak{h} = \{\mathfrak{l}_0 \mapsto \mathfrak{l}_1, \mathfrak{l}_1 \mapsto \mathfrak{l}_2, \ldots, \mathfrak{l}_{n-1} \mapsto \mathfrak{l}_n\}$ with $n \geq 1$ ($\mathfrak{h}$ is made of n memory cells with the obvious values), $\mathfrak{l}_0 = \mathfrak{s}(\mathtt{x})$, $\mathfrak{l}_n = \mathfrak{s}(\mathtt{y})$ and for all $i \neq j \in [0, n]$, $\mathfrak{l}_i \neq \mathfrak{l}_j$.

First suppose (i). Since $\mathrm{dom}(\mathfrak{h}) = \emptyset$, $\mathfrak{R} = \emptyset$ by definition. On the other hand, as $\mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y})$, by $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, there exists some $\mathfrak{l} \in \mathbb{N}$ such that $\mathfrak{M}, \mathfrak{l} \models \mathtt{x} \wedge \mathtt{y}$. Then, since $\mathfrak{M} \models \mathtt{emp}$, we have $\mathfrak{M} \models \phi_{\exists \mathtt{ls}} \wedge \mathtt{emp} \wedge \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y})$.

Now suppose (ii). First notice that $\mathfrak{M}$ is linear by definition, so by Theorem 29 we have $\mathfrak{M} \models \phi_{\exists \mathtt{ls}}$, and since $\mathfrak{R} \neq \emptyset$ by (ii), then $\mathfrak{M} \models \mathsf{UniqTreePRoot} \wedge \langle \, ! \, \rangle \mathsf{Leaf}$. Also by (ii) and definition of $\mathfrak{M}$, we know $\mathfrak{M}, \mathfrak{l}_0 \models \mathtt{x}$. But $\mathfrak{l}_0$ is a leaf, then since $\mathfrak{M} \models \mathsf{UniqTreePRoot}$ we get (by Lemma 28 (III)) $\mathfrak{M}, \mathfrak{l}_0 \models \mathtt{x} \wedge \mathsf{Leaf}$.

On the other hand, since $\mathfrak{l}_n \notin \mathrm{dom}(\mathfrak{h})$, $\mathfrak{l}_{n-1}$ is a pre-root and by Lemma 28 (I), we get $\mathfrak{M}, \mathfrak{l}_{n-1} \models \mathsf{PRoot}$. Moreover, since $\mathfrak{s}(\mathtt{y}) = \mathfrak{l}_n$, by definition of $\mathfrak{R}$ we have $\mathfrak{M}, \mathfrak{l}_{n-1} \models \Diamond \mathtt{y}$. Hence $\mathfrak{M}, \mathfrak{l}_{n-1} \models \mathsf{PRoot} \wedge \Diamond \mathtt{y}$.

Therefore we get $\mathfrak{M} \models \phi_{\exists \mathtt{ls}} \wedge \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathsf{Leaf}) \wedge \langle \mathrm{U} \rangle (\mathsf{PRoot} \wedge \Diamond \mathtt{y})$.

From the two cases above we get $\mathfrak{M} \models t(\mathtt{ls}(\mathtt{x}, \mathtt{y}))$.

Conversely, $\mathfrak{M}, \mathfrak{l} \models t(\mathtt{ls}(\mathtt{x}, \mathtt{y}))$ implies either:

(i) $\mathfrak{M}, \mathfrak{l} \models \phi_{\exists \mathtt{ls}} \wedge \mathtt{emp} \wedge \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathtt{y})$, or

(ii) $\mathfrak{M}, \mathfrak{l} \models \phi_{\exists \mathtt{ls}} \wedge \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathsf{Leaf}) \wedge \langle \mathrm{U} \rangle (\mathsf{PRoot} \wedge \Diamond \mathtt{y})$.

First, let us suppose (i). By $\models$ we have $\mathfrak{R} = \emptyset$ and there exists $\mathfrak{l}' \in \mathbb{N}$ such that $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{l}'\} = \mathfrak{V}(\mathtt{y})$. Then $\mathrm{dom}(\mathfrak{h}) = \emptyset$, $\mathfrak{s}(\mathtt{x}) = \mathfrak{s}(\mathtt{y})$, as $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$. Hence, $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{ls}(\mathtt{x}, \mathtt{y})$.

Now suppose (ii). First, notice that $\mathfrak{M}$ is linear (by Theorem 29), then there exist $\mathfrak{l}_0, \mathfrak{l}_1, \ldots, \mathfrak{l}_n$ such that $\mathfrak{l}_0 \mathfrak{R} \mathfrak{l}_1 \mathfrak{R} \ldots \mathfrak{R} \mathfrak{l}_n$, which gives us $\mathfrak{h} = \{\mathfrak{l}_0 \mapsto \mathfrak{l}_1, \mathfrak{l}_1 \mapsto \mathfrak{l}_2, \ldots, \mathfrak{l}_{n-1} \mapsto \mathfrak{l}_n\}$ with $n \geq 1$. Also, as $\mathfrak{M}, \mathfrak{l} \models \phi_{\exists \mathtt{ls}}$ and $\mathfrak{R} \neq \emptyset$, $\mathfrak{M} \models \mathsf{UniqTreePRoot} \wedge \langle ! \rangle \mathsf{Leaf}$. Since $\mathfrak{l}_0$ is the only leaf of the model, and the fact that $\mathfrak{M}, \mathfrak{l} \models \langle \mathrm{U} \rangle (\mathtt{x} \wedge \mathsf{Leaf})$, we obtain $\mathfrak{V}(\mathtt{x}) = \{\mathfrak{l}_0\}$. Then, as $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, we get $\mathfrak{s}(\mathtt{x}) = \mathfrak{l}_0$.

On the other hand, $\mathfrak{M}, \mathfrak{l} \models \langle \mathrm{U} \rangle (\mathsf{PRoot} \wedge \Diamond \mathtt{y})$, then there exists $(\mathfrak{l}', \mathfrak{l}'') \in \mathfrak{R}$ such that $\mathfrak{V}(\mathtt{y}) = \{\mathfrak{l}''\}$ and there is no $\mathfrak{l}'''$ such that $(\mathfrak{l}'', \mathfrak{l}''') \in \mathfrak{R}$. Then (as $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$) $\mathfrak{s}(\mathtt{y}) = \mathfrak{l}''$, $\mathfrak{h}(\mathfrak{l}') = \mathfrak{s}(\mathtt{y})$ and $\mathfrak{s}(\mathtt{y}) \notin \mathrm{dom}(\mathfrak{h})$, so $\mathfrak{s}(\mathtt{y}) = \mathfrak{l}_n$.

Since $\mathfrak{M}$ is linear, as $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, this gives us that for all $i \neq j \in [0, n]$, $\mathfrak{l}_i \neq \mathfrak{l}_j$. Hence, $(\mathfrak{s}, \mathfrak{h}) \models \mathtt{ls}(\mathtt{x}, \mathtt{y})$. □

For the Boolean cases, $*$ and $-\!\!*$ we only need to apply the induction hypothesis and to use the semantical correspondence $\mathfrak{M} \approx_{\mathtt{N}} (\mathfrak{s}, \mathfrak{h})$, as well as the fact that separation in $\mathfrak{M}$ can be mimicked in $(\mathfrak{s}, \mathfrak{h})$ (and reciprocally) as well as composition of $\mathfrak{M}$ with another model can be also mimicked from $(\mathfrak{s}, \mathfrak{h})$ (and reciprocally).

As the satisfiability problem for $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ is recently shown undecidable [35], we get the following result.

Theorem 36. *The satisfiability problem for* MSL *is undecidable.*

Another consequence is the non-finite axiomatisability of MSL, which is a feature inherited from $\mathrm{SL}(*, -\!\!*, \mathtt{ls})$ (itself inherited from [15]). A bounded number of propositional variables can lead to undecidability too, thanks to [36, Corollary 3.16(IV)] and to [32]. As a corollary, the modal logic for heaps MLH (including $-\!\!*$) augmented with propositional variables is undecidable [31] as MSL is one of its fragments. Moreover, a recent refinement of the main result from [35] allows to conclude that $\mathrm{MSL}(*, -\!\!*, \Diamond)$ is undecidable too [52].

6.2 MSL over general models

We already showed that the global sabotage operation can be translated into $\mathrm{MSL}(*, \Diamond)$ as $t(\langle \mathsf{gsb} \rangle \phi) \stackrel{\text{def}}{=} (\mathtt{size} = 1) * t(\phi)$. This gives us a proof that $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$ is NP-complete over the class of models with finite and functional models. On the other hand, it is known that the satisfiability problem of $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$ (over arbitrary models) is undecidable [4]. Therefore, the minimal modal separation logic over arbitrary models $\mathrm{MSL}^g(*, \Diamond)$ is also undecidable.

Corollary 37. *The satisfiability problem of* $\mathrm{MSL}^g(*, \Diamond)$ *is undecidable.*

On the other hand, it is shown in [2, Theorem 11] and [3, Theorem 2] that the model checking problem for $\mathrm{MSL}^g(\Diamond, \langle \mathsf{gsb} \rangle)$ is PSPACE-complete, which implies the same result for $\mathrm{MSL}^g(*, \Diamond)$.

Corollary 38. *The model checking problem for* $\mathrm{MSL}^g(*, \Diamond)$ *is* PSPACE*-complete.*

7 Conclusion

We have introduced the logic MSL whose models are Kripke-style structures and it can be viewed either as a genuine modal logic or as a genuine separation logic, depending on the connectives considered in the fragments. For the minimal modal separation logic MSL$(*, \Diamond)$, we proved that the satisfiability problem is NP-complete by showing on the one hand the logic satisfies a small model property, and on the other hand that the model-checking problem is in P. A similar complexity characterisation is provided for MSL$(*, \langle \neq \rangle)$, by using a similar approach. Surprisingly, we have shown that the satisfiability problem for the logic MSL$(*, \Diamond, \langle \neq \rangle)$, i.e., the combination of the two logics we mentioned, is Tower-complete. Satisfiability for MSL$(*, \Diamond, \langle \neq \rangle)$ can be checked in Tower since it can be translated into weak monadic second order logic (by simply internalising the semantics of its operators), while Tower-hardness is established by reduction from the nonemptiness problem for star-free expressions. 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 within MSL$(*, \Diamond, \langle \neq \rangle)$, which is far from obvious. 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 MSL$(\Diamond, \langle \mathsf{gsb} \rangle)$ is (only) NP-complete, since it is a fragment of MSL$(*, \Diamond)$. The same translation also gives us undecidability for MSL$^g(*, \Diamond)$, i.e., for the minimal separation logic over the class of general models.

Most of the results are summarised in the table below.

	Model checking (with finite models)	Satisfiability
MSL$(*, \Diamond)$, MSL$(*, \langle \neq \rangle)$	P **(Lemmas 18 and 24)**	NP-complete **(Th. 19 and 25)**
MSL$(*, \Diamond, \langle \neq \rangle)$	PSpace-complete **(Cor. 5)**	Tower-complete **(Th. 34)**
MSL	PSpace-complete	Undecidable **(Th. 36)**
MSL$(*, \Diamond^{-1})$	PSpace-complete **(Th. 6 and Cor. 5)**	PSpace-hard, in Tower **(Th. 6 and 7)**
MSL$(*, \Diamond, \langle \neq \rangle, \Diamond^{-1})$	PSpace-complete **(Cor. 8)**	Tower-complete **(Th. 34 and 7)**
MSL$(\Diamond, \langle \mathsf{gsb} \rangle)$	P **(Lemma 18)**	NP-complete **(Cor. 20)**
MSL$^g(*, \Diamond)$	PSpace-complete **(Cor. 38)**	Undecidable **(Cor. 37)**

As a latest news, in [53], it is shown that the satisfiability problem for MSL$(\Diamond, \langle \mathrm{U} \rangle, *)$ with unique atomic formula $\top$ (i.e. with no propositional variable) is already Tower-hard. In general, understanding the effects of the interactions between modal operators and separating connectives is still to be strengthened and many interesting problems are left open. For instance, it would be worth investigating whether our results could be extended, adapted or related for combinations of modal/temporal/epistemic logics with abstract separation logics, along the lines of the logics investigated in [22, 25, 24, 23, 26]. Similarly, we have shown that the satisfiability problem for MSL$(*, \Diamond^{-1})$ is PSpace-hard and in Tower but a complexity characterisation is not yet known. The decidability status of MSL^f and MLH [31] is also open. Furthermore, even though the modalities we include in our language are not completely arbitrary but chosen in terms of their expressivity, many other posibilities remain to be explored, e.g. $\langle \star \rangle$ or $\Diamond^{-1}$. Finally, the design of proof systems for modal separation logics remains a challenging question and preliminary recent results can be found in [34].

Acknowledgements. We would like to thank the reviewers for their helpful comments, and Bartosz Bednarczyk, Alessio Mansutti and Andrés Saravia for helpful suggestions and enlightning discussions. This work was partially supported by ANPCyT-PICTs-2017-1130 and 2016-0215, the Laboratoire International Associé INFINIS and the Centre National de la Recherche Scientifique (CNRS).

References

[1] K. Apt. Ten Years of Hoare's Logic. *ACM Transactions on Programming Languages and Systems*, 3(4):431–483, 1981.

[2] C. Areces, R. Fervari, and G. Hoffmann. Moving arrows and four model checking results. In *WoLLIC'12*, volume 7456 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2012.

[3] C. Areces, R. Fervari, and G. Hoffmann. Relation-Changing Modal Operators. *Logic Journal of the IGPL*, 23(4):601–627, 2015.

[4] C. Areces, R. Fervari, G. Hoffmann, and M. Martel. Satisfiability for relation-changing logics. *Journal of Logic and Computation*, 28(7):1443–1470, 2018.

[5] G. Aucher, Ph. Balbiani, L. Fariñas del Cerro, and A. Herzig. Global and local graph modifiers. *ENTCS*, 231:293–307, 2009.

[6] F. Baader, I. Horrocks, C. Lutz, and U. Sattler. *An Introduction to Description Logic.* Cambridge University Press, 2017.

[7] B. Bednarczyk and S. Demri. Why propositional quantification makes modal logics on trees robustly hard? In *LICS'19*, 2019.

[8] J. Berdine, C. Calcagno, and P. O'Hearn. A decidable fragment of separation logic. In *FST&TCS'04*, volume 3328 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 2004.

[9] P. Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL*, 8(3):339–365, 2000.

[10] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2001.

[11] I. Boneva, J.-M. Talbot, and S. Tison. Expressiveness of a spatial logic for trees. In *LICS'05*, pages 280–289. IEEE Computer Society, 2005.

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

[13] J. Boudou. Decidable logics with associative binary modalities. In *CSL'17*, volume 82 of *LIPIcs*, pages 1–15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[14] R. Brochenin, S. Demri, and E. Lozes. Reasoning about sequences of memory states. *Annals of Pure and Applied Logic*, 161(3):305–323, 2009.

[15] R. Brochenin, S. Demri, and E. Lozes. On the almighty wand. *Information and Computation*, 211:106–137, 2012.

[16] J. Brotherston and M. Kanovich. Undecidability of propositional separation logic and its neighbours. *Journal of the ACM*, 61(2), 2014.

[17] J. Brotherston and J. Villard. Parametric completeness for separation theories. In *POPL'14*, pages 453–464. ACM, 2014.

[18] C. Calcagno, L. Cardelli, and A.D. Gordon. Deciding validity in a spatial logic for trees. In *TLDI'03*, pages 62–73. ACM, 2003.

[19] C. Calcagno, P. O'Hearn, and H. Yang. Computability and complexity results for a spatial assertion language for data structures. In *FSTTCS'01*, volume 2245 of *Lecture Notes in Computer Science*, pages 108–119. Springer, 2001.

[20] D. Calvanese, 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*, volume 8739 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2014.

[21] B. Cook, C. Haase, J. Ouaknine, M. Parkinson, and J. Worrell. Tractable reasoning in a fragment of separation logic. In *CONCUR'11*, volume 6901 of *Lecture Notes in Computer Science*, pages 235–249. Springer, 2011.

[22] J.-R. Courtault and D. Galmiche. A modal BI logic for dynamic resource properties. In *LFCS'13*, volume 7734 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2013.

[23] J.-R. Courtault and D. Galmiche. A modal separation logic for resource dynamics. *Journal of Logic and Computation*, 28(4):733–778, 2018.

[24] J.-R. Courtault, D. Galmiche, and D. Pym. A logic of separating modalities. *Theoretical Computer Science*, 637:30–58, 2016.

[25] J.-R. Courtault, H. van Ditmarsch, and D. Galmiche. An epistemic separation logic. In *WoLLIC 2015*, pages 156–173, 2015.

[26] J.-R. Courtault, H. van Ditmarsch, and D. Galmiche. A public announcement separation logic. *Mathematical Structures in Computer Science*, 29(6):828–871, 2019.

[27] A. Dawar, Ph. Gardner, and G. Ghelli. Expressiveness and complexity of graph logic. *Information and Computation*, 205(3):263–310, 2007.

[28] M. de Rijke. The modal logic of inequality. *The Journal of Symbolic Logic*, 57(2):566–584, 1992.

[29] S. Demri. A simple tableau system for the logic of elsewhere. In *TABLEAUX'96*, volume 1071 of *Lecture Notes in Artificial Intelligence*, pages 177–192. Springer, 1996.

[30] S. Demri and M. Deters. Separation logics and modalities: A survey. *Journal of Applied Non-Classical Logics*, 25(1):50–99, 2015.

[31] S. Demri and M. Deters. Two-variable separation logic and its inner circle. *ACM Transactions on Computational Logic*, 2(16), 2015.

[32] S. Demri and M. Deters. Expressive completeness of separation logic with two variables and no separating conjunction. *ACM Transactions on Computational Logic*, 17(2):12, 2016.

[33] S. Demri and R. Fervari. On the complexity of modal separation logics. In *AiML'18*, pages 179–198. College Publications, 2018.

[34] S. Demri, R. Fervari, and A. Mansutti. Axiomatising logics with separating conjunctions and modalities. In *JELIA'19*, volume 11468 of *Lecture Notes in Artificial Intelligence*, pages 692–708. Springer, 2019.

[35] S. Demri, E. Lozes, and A. Mansutti. The effects of adding reachability predicates in propositional separation logic. In *FoSSaCS'18*, volume 10803 of *Lecture Notes in Computer Science*, pages 476–493. Springer, 2018.

[36] S. Demri, E. Lozes, and A. Mansutti. The effects of adding reachability predicates in propositional separation logic. arXiv:1810.05410, October 2018. 44 pages. Under submission.

[37] K. Fine. *For some proposition and so many possible worlds.* PhD thesis, University of Warwick, 1969.

[38] K. Fine. Some connections between elementary and modal logic. In *3rd Scandinavian Logic Symposium*, pages 15–31. North Holland, 1975.

[39] D. Gabbay. Decidability results in non-classical logics. *Annals of Mathematical Logic*, 8:237–295, 1975.

[40] D. Gabbay. *Reactive Kripke Semantics.* Springer, 2013.

[41] D. Galmiche, P. Kimmel, and D. Pym. A substructural epistemic resource logic. In *ICLA'17*, volume 10119 of *Lecture Notes in Computer Science*, pages 106–122. Springer, 2017.

[42] L. Georgieva and P. Maier. Description logics for shape analysis. In *SEFM'05*, pages 321–331. IEEE Computer Society, 2005.

[43] M. Hannula, J. Kontinen, J. Virtema, and H. Vollmer. Complexity of propositional logics in team semantic. *ACM Transactions on Computational Logic*, 19(1):2:1–2:14, 2018.

[44] E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37(2):173–203, 1996.

[45] A. Herzig. A simple separation logic. In *WoLLIC'13*, volume 8071 of *Lecture Notes in Computer Science*, pages 168–178. Springer, 2013.

[46] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[47] I. Hodkinson, A. Montanari, and G. Sciavicco. Non-finite axiomatizability and undecidability of interval temporal logics with C, D, and T. In *CSL'08*, volume 5213 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2008.

[48] S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *POPL'01*, pages 14–26. ACM, 2001.

[49] D. Larchey-Wendling and D. Galmiche. Nondeterministic phase semantics and the undecidability of Boolean BI. *ACM Transactions on Computational Logic*, 14(1), 2013.

[50] Ch. Löding and Ph. Rohde. Model checking and satisfiability for sabotage modal logic. In *FST&TCS'03*, volume 2914 of *Lecture Notes in Computer Science*, pages 302–313. Springer, 2003.

[51] C. Lutz. Complexity and succinctness of public announcement logic. In *AAMAS'06*, pages 137–143. ACM, 2006.

[52] A. Mansutti. Private communication, October 2018.

[53] A. Mansutti. An auxiliary logic on trees: on the tower-hardness of logics featuring reachability and submodel reasoning, October 2019. Under submission.

[54] A. Meyer and L. Stockmeyer. Word problems requiring exponential time. In *STOC'73*, pages 1–9. ACM, 1973.

[55] R.C. Moore. Reasoning about knowledge and action. In *IJCAI-5*, pages 223–227, 1977.

[56] Ch. Morgan. Methods for automated theorem proving in non classical logics. *IEEE Transactions on Computers*, 25(8):852–862, 1976.

[57] B. Moszkowski. Reasoning about digital circuits. Technical Report STAN-CS-83-970, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.

[58] P.W. O'Hearn, J.C. Reynolds, and H. Yang. Local reasoning about programs that alter data structures. In *CSL'01*, volume 2142 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2001.

[59] D. Pym, J. Spring, and P.W. O'Hearn. Why separation logic works. *Philosophy & Technology*, pages 1–34, 2018.

[60] M. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 41:1–35, 1969.

[61] J.C. Reynolds. Separation logic: a logic for shared mutable data structures. In *LICS'02*, pages 55–74. IEEE, 2002.

[62] S. Schmitz. Complexity hierarchies beyond Elementary. *ACM Transactions on Computation Theory*, 8(1):3:1–3:36, 2016.

[63] K. Segerberg. A note on the logic of elsewhere. *Theoria*, 47:183–187, 1981.

[64] L. Stockmeyer. *The complexity of decision problems in automata theory and logic.* PhD thesis, Department of Electrical Engineering, MIT, 1974.

[65] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–21, 1977.

[66] J. van Benthem. *Correspondence Theory.* PhD thesis, Mathematical Institute, University of Amsterdam, 1976.

[67] J. van Benthem. An Essay on Sabotage and Obstruction. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg Siekmann on the Occasion of his 69th Birthday*, pages 268–276. Springer Verlag, 2005.