Prenex Separation Logic with One Selector Field

Mnacho Echenim[1], Radu Iosif[2] and Nicolas Peltier[1]

[1] Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble France
[2] Univ. Grenoble Alpes, CNRS, VERIMAG, F-38000 Grenoble France

Abstract. We show that infinite satisfiability can be reduced to finite satisfiability for all prenex formulas of Separation Logic with $k \geq 1$ selector fields (SL^k). This fact entails the decidability of the finite and infinite satisfiability problems for the class of prenex formulas of SL^1, by reduction to the first-order theory of a single unary function symbol and an arbitrary number of unary predicate symbols. We also prove that the complexity of this fragment is not elementary recursive, by reduction from the first-order theory of one unary function symbol. Finally, we prove that the Bernays-Schönfinkel-Ramsey fragment of prenex SL^1 formulas with quantifier prefix in the language $\exists^*\forall^*$ is PSPACE-complete.

1 Introduction

Separation Logic [8,11] (SL) is a logical framework used to describe properties of the heap memory, such as the placement of pointer variables within the topology of complex data structures (lists, trees, etc.). The features that make SL attractive for program verification are the ability of defining (i) weakest pre- and post-condition calculi that capture the semantics of programs with pointers, and (ii) compositional verification methods, based on inferring local specifications of methods and threads independently of the context in which they evolve. The search for automated push-button program verification methods motivates the understanding of the decidability, complexity and expressive power of various dialects of SL, used as assertion languages in Hoare-style proofs [8], or logic-based abstract domains in static analysis [3].

Formal definitions are provided later, but essentially, SL can be viewed as the first order theory of one partial finite function from $\mathfrak{U} \rightarrow \mathfrak{U}^k$, called a *heap*, where $\mathfrak{U}$ denotes the universe of memory locations (i.e., addresses), to which two non-classical connectives are added: (i) the *separating conjunction* $\phi_1 * \phi_2$, that asserts a split of the heap into disjoint heaps satisfying ϕ_1 and ϕ_2 respectively, and (ii) the *separating implication* or *magic wand* $\phi_1 \mathrel{-\!\!*} \phi_2$, stating that each extension of the heap by a disjoint heap satisfying ϕ_1 must satisfy ϕ_2. The number k denotes the number of selector fields and we use the notation SL^k to make this number explicit. Quantification over elements of $\mathfrak{U}$ is allowed. A fragment of separation logic that is practically relevant in verification is when $k = 1$, i.e., every allocated cell points to a unique cell. This fragment allows, e.g., to describe simply linked lists.

As a simple example of application, let us consider the following Hoare triple with left-hand side that is the weakest precondition of an arbitrary formula ϕ with respect to a selector update in a program handling lists:

$$\{\exists x \,.\, \mathsf{i} \mapsto x * (\mathsf{i} \mapsto \mathsf{j} \mathrel{-\!\!*} \phi)\} \quad \mathsf{i.next} = \mathsf{j} \quad \{\phi\}$$

Informally, the formula $\exists x . \mathsf{i} \mapsto x * (\mathsf{i} \mapsto \mathsf{j} \mathrel{-\!\!*} \phi)$ holds when the heap can be separated into disjoint parts, one in which cell i is allocated (the formula $\mathsf{i} \mapsto x$ states that the heap maps i to x), and one that, when extended by allocating cell i to j, satisfies ϕ. In other words, the formula states that cell i is allocated and that ϕ holds after i is redirected to j. A typical verification condition checks whether this formula is entailed by another precondition ψ, generated by a program verifier or supplied by the user. The entailment $\psi \models \exists x . \mathsf{i} \mapsto x * (\mathsf{i} \mapsto \mathsf{j} \mathrel{-\!\!*} \phi)$ is valid if and only if the formula $\theta \stackrel{\mathsf{def}}{=} \psi \wedge \forall x . \neg(\mathsf{i} \mapsto x * (\mathsf{i} \mapsto \mathsf{j} \mathrel{-\!\!*} \phi))$ is unsatisfiable. In addition, if ϕ and ψ are formulas in prenex form[3] then, because the assertions $\mathsf{i} \mapsto x$ and $\mathsf{i} \mapsto \mathsf{j}$ unambiguously define a specific part of the heap (the cell corresponding to i), the quantifiers of ϕ can be hoisted outside of the separating conjunction and implication, and the formula θ can be written in prenex form.

Deciding the satisfiability of (prenex) SL formulas is thus an important ingredient for push-button program verification. Unlike first order logic, some SL formulas do not have a prenex form (see Example 2 on Page 7). Moreover, satisfiability is decidable (and PSPACE-complete) for quantifier-free SL-formulas, but it is undecidable for first-order SL-formulas, even when $k = 1$. In fact SL^1 is as expressive as second-order logic in the presence of $*$ and $\mathrel{-\!\!*}$ whereas the fragment of SL^1 without $\mathrel{-\!\!*}$ is decidable but not elementary recursive [2]. In [6], we investigated the Bernays-Schönfinkel-Ramsey fragment of SL^k, i.e., the fragment containing formulas of the form $\exists x_1, \ldots, x_n \forall y_1, \ldots, y_m . \phi$ where ϕ is a quantifier-free formula of SL^k. We proved that for $k > 1$, satisfiability is undecidable in general and decidable if $\mathrel{-\!\!*}$ only occurs in the scope of an odd number of negations. However, nothing is known concerning the prenex fragment of SL^1. In this paper we fill in this gap and show that:

1. the prenex fragment of SL^1 is decidable but not elementary recursive, and
2. the Bernays-Schönfinkel-Ramsey fragment of SL^1 is PSPACE-complete.

The results are established using reductions to and from the fragment of first order logic with one monadic function symbol [1]. The decidability of this fragment is a consequence of the celebrated Rabin Tree Theorem [10], which established the decidability of monadic second order logic of infinite binary tree (S2S). As in our previous work [6] and unlike most existing approaches, we consider both the finite and infinite satisfiability problems (other approaches usually assume that the universe is infinite). Essential to our reductions to and from this fragment is a result (proven in [6]) stating that each quantifier-free SL^k formula, for $k \geq 1$, is equivalent to a boolean combination of formulas of some specific forms, called *test formulas*. Similar translations exist for quantifier-free SL^1 [9,2] and for SL^1 with one quantified variable [5]. In addition we show in the present paper that the infinite satisfiability reduces to the finite satisfiability for quantified boolean quantifications of test formulas.

2 Preliminaries

In this section, we briefly review some usual definitions and notations (missing definitions can be found in, e.g., [7] or [1]). We denote by $\mathbb{Z}$ the set of integers and by $\mathbb{N}$ the set of positive integers including zero. We define $\mathbb{Z}_\infty = \mathbb{Z} \cup \{\infty\}$ and $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$,

[3] $Q_1 x_1 \ldots Q_n x_n . \varphi$, where $Q_1, \ldots, Q_n$ are the first order quantifiers $\exists$ or $\forall$ and φ is quantifier-free.

where for each $n \in \mathbb{Z}$ we have $n + \infty = \infty$ and $n < \infty$. For two positive integers $m \leq n$, we denote by $[\![m \mathinner{..} n]\!]$ the set $\{m, m+1, \ldots, n\}$. For a countable set S we denote by $||S|| \in \mathbb{N}_\infty$ the cardinality of S. A decision problem is in $\mathsf{(N)SPACE}(n)$ if it can be decided by a (nondeterministic) Turing machine in space $O(n)$ and in PSPACE if it is in $\mathsf{SPACE}(n^c)$ for some input-independent integer $c \geq 1$.

2.1 First Order Logic

Syntax Let Var be a countable set of *variables*, denoted by x, y, z and B and U be distinct *sorts*, where B denotes booleans and U denotes memory locations. A *function symbol* f has $\#(f) \geq 0$ arguments of sort U and a sort $\sigma(f)$, which is either B or U. If $\#(f) = 0$, we call f a *constant*. We use $\bot$ and $\top$ for the boolean constants false and true, respectively. First-order (FO) *terms* t and *formulas* φ are defined by the following grammar:

$$t := x \mid f(\underbrace{t, \ldots, t}_{\#(f)}) \qquad \varphi := \bot \mid \top \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists x \,.\, \varphi \mid t \approx t \mid p(\underbrace{t, \ldots, t}_{\#(p)})$$

where $x \in \mathsf{Var}$, f and p are function symbols, $\sigma(f) = U$ and $\sigma(p) = B$. We write $\varphi_1 \vee \varphi_2$ for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2$ for $\neg\varphi_1 \vee \varphi_2$, $\varphi_1 \leftrightarrow \varphi_2$ for $\varphi_1 \rightarrow \varphi_2 \wedge \varphi_2 \rightarrow \varphi_1$ and $\forall x \,.\, \varphi$ for $\neg\exists x \,.\, \neg\varphi$. The *size* of a formula φ, denoted by $\mathsf{size}(\varphi)$, is the number of occurrences of symbols in φ. A variable is *free* in φ if it occurs in φ but not in the scope of a quantifier. We denote by $\mathrm{fv}(\varphi)$ the set of variables that are free in φ. A *sentence* is a formula φ such that $\mathrm{fv}(\varphi) = \emptyset$. The *Bernays-Schönfinkel-Ramsey fragment* of FO [$\mathsf{BSR(FO)}$] is the set of sentences of the form $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \,.\, \varphi$, where φ is a quantifier-free formula in which all function symbols f of arity $\#(f) > 0$ have sort $\sigma(f) = B$. We denote by FO^1 the set of formulas built on a signature containing only one function symbol of arity 1, the equality predicate and an arbitrary number of unary predicate symbols[4].

Semantics First-order formulas are interpreted over FO-*structures* (called structures, when no confusion arises) $\mathcal{S} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{i})$, where $\mathfrak{U}$ is a nonempty countable set, called the *universe*, the elements of which are called *locations*, $\mathfrak{s} : \mathsf{Var} \rightarrow \mathfrak{U}$ is a function mapping variables to locations called a *store*, and $\mathfrak{i}$ interprets each function symbol f by a function $f^{\mathfrak{i}} : \mathfrak{U}^{\#(f)} \rightarrow \mathfrak{U}$, if $\sigma(f) = U$ and $f^{\mathfrak{i}} : \mathfrak{U}^{\#(f)} \rightarrow \{\bot, \top\}$ if $\sigma(f) = B$. A structure $(\mathfrak{U}, \mathfrak{s}, \mathfrak{i})$ is *finite* when $||\mathfrak{U}|| \in \mathbb{N}$ and *infinite* otherwise. We write $\mathcal{S} \models \varphi$ iff φ is true when interpreted in $\mathcal{S}$. This relation is defined recursively on the structure of φ, as usual. When $\mathcal{S} \models \varphi$, we say that $\mathcal{S}$ is a *model* of φ. A formula is *satisfiable* when it has a model. We write $\varphi_1 \models \varphi_2$ when every model of φ_1 is also a model of φ_2 and by $\varphi_1 \equiv \varphi_2$ we mean $\varphi_1 \models \varphi_2$ and $\varphi_1 \models \varphi_2$. The *(in)finite satisfiability problem* asks, given a formula φ, whether a (in)finite model exists for this formula.

We now recall and refine an essential known result concerning the satisfiability problem for formulas in FO^1:

Theorem 1. *The finite satisfiability problem is decidable for first-order formulas in* FO^1. *Furthermore, the problem is nonelementary even if the formula contains no unary predicate symbols.*

[4] The fragment FO^1 is denoted by $[all, (\omega), (1)]_=$ in [1].

Proof. The decidability result is proven in [1, Corollary 7.2.12, page 341]. The complexity lower bound is established in [1, Theorem 7.2.15, page 342] for arbitrary domains, however a careful analysis of the proof reveals that it also holds for finite domains. Indeed, the proof goes by showing that a domino problem of nonelementary complexity can be polynomially reduced to the satisfiability problem for a first-order formula φ satisfying the conditions of the lemma. The initial domino problem is not important here and its definition is omitted. To establish the desired result, we only have to prove that satisfiability is actually equivalent to finite satisfiability for the obtained formula φ. The formula φ output of the reduction is of the following form (see [1, Page 345]): $\varphi = \alpha \wedge \gamma \wedge \eta'[D(x)/\delta(x), P_i(x,y)/\pi_i(x,y)$, where:

- $\alpha = \exists x \forall y \,.\, f(x) \approx x \wedge f^{n+1}(y) \approx x$. This formula states that the domain can be viewed as a tree of height at most $n+1$, where the (necessarily unique) element corresponding to the variable x is the root of the tree, and where f maps every other node to its parent.
- The formula δ is based on an equivalence relation E_{n-1} on nodes in a (possibly infinite) tree, which is inductively defined as follows:
  - All nodes are E_0-equivalent.
  - For $m > 1$, two nodes are E_m-equivalent if for every E_{m-1}-equivalence class K, either both nodes have no child in K or both nodes have a child in K.

  The formula $\delta(x)$ states that x is a child of the root with at most one child in each E_{n-1}-equivalence class. We also denote by E the intersection $\bigcap_{i=1}^{n} E_i$.
- $\gamma = \forall x, y \,.\, \delta(x) \wedge \delta(y) \wedge \beta_n(x,y) \rightarrow x \approx y$, where $\beta_n(x,y)$ is a formula stating that x and y have height at most n and are E_n-equivalent.
- For $i = 0, \dots, r$, $\pi_i(x,y)$ is a formula stating that there exists a z satisfying the following property denoted by $\mathcal{P}(i,a,b)$: z is a child of the root and for every E_{n-1}-equivalence class K and for all $j,k \in \{0,1\}$, if x and y have exactly j and k children in K respectively, then z has exactly $2+4i+2j+k$ children in K.
- η' is equivalent to a closed formula defined over a signature containing a unary predicate symbol D and $r+1$ binary predicate symbols $P_0, \dots, P_r$, in which every quantification ranges over elements x satisfying $D(x)$. It is thus of the form $\exists x \,.\, D(x) \wedge \psi$ or $\forall x \,.\, D(x) \rightarrow \psi$.
- $\eta'[D(x)/\delta(x), P_i(x,y)/\pi_i(x,y)]$ denotes the formula η' in which every occurrence of a formula $D(x)$ (resp. $P_i(x,y)$) is replaced by $\delta(x)$ (resp. $\pi_i(x,y)$). Thus it is equivalent to a formula in which every quantification ranges over elements x satisfying $\delta(x)$.

The formal definitions of η', $\delta(x)$ and $\pi_i(x,y)$ are unimportant and omitted.

Let $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{i})$ be a model of φ, with $\mathfrak{f} = f^{\mathfrak{i}}$. We denote by $\mathfrak{r}$ the root of the tree, i.e., the unique element of $\mathfrak{U}$ with $(\mathfrak{U}, \mathfrak{s}[x \mapsto \mathfrak{r}], \mathfrak{i}) \models \forall y \,.\, f(x) \approx x \wedge f^{n+1}(y) \approx x$. Given $i \in [0, r]$ and $a, b \in \mathfrak{U}$, if $(\mathfrak{U}, \mathfrak{s}[x \mapsto a, y \mapsto b], \mathfrak{i}) \models \pi_i(x,y)$, then we denote by $\mu(i,a,b)$ a set containing an arbitrarily chosen element z satisfying $\mathcal{P}(i,a,b)$ in the definition of $\pi_i(x,y)$ along with all the children of z, otherwise $\mu(i,a,b)$ is empty. Observe that $\mu(i,a,b)$ is always finite because the number of children of z in each equivalence class is bounded by $2+4 \times i+2+1 \leq 2+4 \times r+2+1$, moreover the number of E-equivalence classes is finite [1, bottom of Page 343].

We show that φ admits a finite model $\mathcal{I}'$. The set B of elements b such that $(\mathfrak{U},\mathfrak{s}[x \mapsto b],\mathfrak{i}) \models \delta(x)$ is finite [1, Page 344, Lines 21-22]. Let Π be the set: $\Pi = \bigcup\{\mu(i,a,b) \mid a,b \in B, i \in [0,r]\}$. Since B is finite and every set $\mu(i,a,b)$ is finite, Π is also finite. With each element $a \in \mathfrak{U}$ and each E-equivalence class K, we associate a set $\nu(a,K)$ containing exactly one child of a in K if such a child exists, otherwise $\nu(a,K)$ is empty. We now consider the subset $\mathfrak{U}'$ of $\mathfrak{U}$ defined as the set of elements a such that for every $m \in \mathbb{N}$, $\mathfrak{f}^m(a)$ occurs either in $\{\mathfrak{r}\} \cup B \cup \Pi$ or in a set $\nu(b,K)$, where $b \in \mathfrak{U}$ and K is an E-equivalence class. Note that $\mathfrak{r} \in \mathfrak{U}'$ and that if $a \in \mathfrak{U}'$ then necessarily $\mathfrak{f}(a) \in \mathfrak{U}'$. Furthermore, if $\mathfrak{f}(b) \in \mathfrak{U}'$ and $b \in \nu(\mathfrak{f}(b),K)$ then $b \in \mathfrak{U}'$.

It is easy to check that $\mathfrak{U}'$ is finite. Indeed, since $(\mathfrak{U},\mathfrak{s},\mathfrak{i}) \models \alpha$ and no new node or edge is added, all nodes are of height less or equal to $n+1$. Furthermore, all nodes have at most $||B|| + ||\Pi|| + \#K$ children in $\mathfrak{U}'$, where $\#K$ denotes the number of E-equivalence classes.

We denote by $\mathcal{I}' = (\mathfrak{U}',\mathfrak{s},\mathfrak{i}')$ the restriction of $\mathcal{I}$ to the elements of $\mathfrak{U}'$ (we may assume that $\mathfrak{s}$ is a store on $\mathfrak{U}'$ since φ is closed). We prove that $\mathcal{I}' \models \varphi$.

- Since $\mathfrak{U}'$ contains the root, and $\mathcal{I} \models \alpha$, we must have $\mathcal{I}' \models \alpha$.
- Observe that $\mathfrak{U}'$ necessarily contains $\nu(b,K)$, for every $b \in \mathfrak{U}'$, since by definition the parent of the (unique) element of $\nu(b,K)$ is b. Thus at least one child of b is kept in each equivalence class. Thus the relations E_m on elements of $\mathfrak{U}'$ are preserved in the transformation: for every $a,b \in \mathfrak{U}'$, a,b are E_m-equivalent in the structure $\mathcal{I}$ iff they are equivalent in the structure $\mathcal{I}'$. Further, the height of the nodes cannot change. Therefore, for every $a,a' \in U'$:

  $$(\mathfrak{U}',\mathfrak{s}[x \mapsto a, y \mapsto a'],\mathfrak{i}') \models \beta_n(x,y) \text{ iff } (\mathfrak{U},\mathfrak{s}[x \mapsto a, y \mapsto a'],\mathfrak{i}) \models \beta_n(x,y)$$

  By definition, for every $a \in B$ and $m \in \mathbb{N}$, $\mathfrak{f}^m(a) \in \{a,\mathfrak{r}\}$, thus $B \subseteq \mathfrak{U}'$. Because no new edges are added, we deduce:

  $$(\mathfrak{U}',\mathfrak{s}[x \mapsto a],\mathfrak{i}') \models \delta(x) \Leftrightarrow (\mathfrak{U},\mathfrak{s}[x \mapsto a],\mathfrak{i}) \models \delta(x) \Leftrightarrow a \in B$$

  Consequently, since $\mathcal{I} \models \gamma$, we have $\mathcal{I}' \models \gamma$.
- All elements in $\mu(i,a,a')$ with $a,a' \in B$ occur in $\mathfrak{U}'$ (because if $b \in \mu(i,a,a')$ and $m \in \mathbb{N}$ then $\mathfrak{f}^m(b) \in \{\mathfrak{r}\} \cup B \cup \mu(i,a,a')$), thus, for all $a,a' \in B$:

  $$(\mathfrak{U}',\mathfrak{s}[x \mapsto a, y \mapsto a'],\mathfrak{i}') \models \pi_i(x,y) \Leftrightarrow (\mathfrak{U},\mathfrak{s}[x \mapsto a, y \mapsto a'],\mathfrak{i}) \models \pi_i(x,y)$$

  Since all quantifications in η' range over elements in B, we deduce, by a straightforward induction on the formula, that $\mathcal{I}$ and $\mathcal{I}'$ necessarily agree on the formula $\eta'[D(x)/\delta(x), P_i(x,y)/\pi_i(x,y)]$. Consequently, $\mathcal{I}' \models \eta'[D(x)/\delta(x), P_i(x,y)/\pi_i(x,y)]$.

□

2.2 Separation Logic

Syntax Let $k \in \mathbb{N}$ be a strictly positive integer. The logic SL^k is the set of formulas generated by the grammar:

$$\varphi := \bot \mid \top \mid \mathsf{emp} \mid x \approx y \mid x \mapsto (y_1,\ldots,y_k) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi * \varphi \mid \varphi \mathrel{-\!\!*} \varphi \mid \exists x \,.\, \varphi$$

where $x, y, y_1, \dots, y_k \in \mathsf{Var}$. The connectives $*$ and $-\!\!*$ are respectively called the *separating conjunction* and *separating implication* (*magic wand*). The symbols $\vee$, $\rightarrow$, $\leftrightarrow$ and $\forall$ are defined as in first-order logic, and in addition, we write $\varphi_1 \multimap \varphi_2$ for $\neg(\varphi_1 -\!\!* \neg\varphi_2)$ ($\multimap$ is called *septraction*).

A tuple $(y_1, \dots, y_k) \in \mathsf{Var}^k$ is sometimes denoted by $\mathbf{y}$. The *size* and *free variables* of an SL^k formula φ are defined as for first-order formulas. The *prenex fragment* of SL^k (denoted by $\mathsf{PRE}(\mathsf{SL}^k)$) is the set of sentences $Q_1 x_1 \dots Q_n x_n \, . \, \phi$, where $Q_1, \dots, Q_n \in \{\exists, \forall\}$ and ϕ is a quantifier-free SL^k formula. The *Bernays-Schönfinkel-Ramsey fragment* of SL^k [$\mathsf{BSR}(\mathsf{SL}^k)$] is the set of sentences $\exists x_1 \dots \exists x_n \forall y_1 \dots \forall y_m \, . \, \phi$, where ϕ is a quantifier-free SL^k formula. Since there are no function symbols of arity greater than zero in SL^k, there are no restrictions, other than the form of the quantifier prefix, defining $\mathsf{BSR}(\mathsf{SL}^k)$.

Semantics SL^k formulas are interpreted over SL-*structures* (called structures when no confusion arises) $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$, where $\mathfrak{U}$ and $\mathfrak{s}$ are defined as for first-order formulas[5] and $\mathfrak{h} : \mathfrak{U} \rightharpoonup_{fin} \mathfrak{U}^k$ is a finite partial mapping of locations to k-tuples of locations, called a *heap*. A structure $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ is finite when $||\mathfrak{U}|| \in \mathbb{N}$ and infinite otherwise (note that the heap is always finite, but that the universe may be finite or infinite).

Given a heap $\mathfrak{h}$, we denote by $\mathsf{dom}(\mathfrak{h})$ the domain of the heap, by $\mathsf{img}(\mathfrak{h}) \stackrel{\mathsf{def}}{=} \{\ell_i \mid \exists \ell \in \mathsf{dom}(\mathfrak{h}), \mathfrak{h}(\ell) = (\ell_1, \dots, \ell_k), i \in \llbracket 1 \mathinner{..} k \rrbracket\}$ its range and we let $\mathrm{elems}(\mathfrak{h}) \stackrel{\mathsf{def}}{=} \mathsf{dom}(\mathfrak{h}) \cup \mathsf{img}(\mathfrak{h})$. A element x is *allocated* in $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ if it belongs to $\mathsf{dom}(\mathfrak{h})$. For a store $\mathfrak{s}$, we define its range $\mathsf{img}(\mathfrak{s}) \stackrel{\mathsf{def}}{=} \{\ell \mid x \in \mathsf{Var},\ \mathfrak{s}(x) = \ell\}$. If $\mathbf{x} = (x_1, \dots, x_n)$ is a vector of pairwise distinct variables and $\mathbf{e} = (e_1, \dots, e_n)$ is a vector of elements of $\mathfrak{U}$ of the same length as $\mathbf{x}$, then $\mathfrak{s}[\mathbf{x} \mapsto \mathbf{e}]$ denotes the store that maps x_i to e_i (for all $i \in \llbracket 1 \mathinner{..} n \rrbracket$) and coincides with $\mathfrak{s}$ on every variable distinct from $x_1, \dots, x_n$. Two heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are *disjoint* if and only if $\mathsf{dom}(\mathfrak{h}_1) \cap \mathsf{dom}(\mathfrak{h}_2) = \emptyset$, in which case $\mathfrak{h}_1 \uplus \mathfrak{h}_2$ denotes their union ($\mathfrak{h}_1 \uplus \mathfrak{h}_2$ is undefined if $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are not disjoint). The relation $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \varphi$ is defined inductively, as follows:

$$\begin{array}{ll}
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \mathsf{emp} & \Leftrightarrow \mathfrak{h} = \emptyset \\
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models x \approx y & \Leftrightarrow \mathfrak{s}(x) = \mathfrak{s}(y) \\
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models x \mapsto (y_1, \dots, y_k) & \Leftrightarrow \mathfrak{h}(\mathfrak{s}(x)) = (\mathfrak{s}(y_1), \dots, \mathfrak{s}(y_k)) \wedge \mathsf{dom}(\mathfrak{h}) = \{\mathfrak{s}(x)\} \\
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \varphi_1 \text{ and } (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \varphi_2 \\
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \neg\varphi & \Leftrightarrow (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \not\models \varphi \\
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \exists x \, . \, \varphi & \Leftrightarrow \text{there exists } e \in \mathfrak{U} \text{ s.t. } (\mathfrak{U}, \mathfrak{s}[x \mapsto e], \mathfrak{h}) \models \varphi
\end{array}$$

$$\begin{array}{ll}
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \varphi_1 * \varphi_2 & \Leftrightarrow \text{there exist disjoint heaps } \mathfrak{h}_1, \mathfrak{h}_2 \text{ such that } \mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2 \\
& \quad\ \text{and } (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}_i) \models \varphi_i, \text{ for } i = 1, 2 \\
(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \varphi_1 -\!\!* \varphi_2 & \Leftrightarrow \text{for all heaps } \mathfrak{h}' \text{ disjoint from } \mathfrak{h} \text{ such that } (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}') \models \varphi_1, \\
& \quad\ \text{we have } (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}' \uplus \mathfrak{h}) \models \varphi_2
\end{array}$$

Satisfiability, entailment and equivalence are defined for SL^k as for FO formulas. The finite [resp. infinite] satisfiability problem for SL^k asks whether a finite [resp. an infinite] model exists for a given formula. We write $\phi \equiv^{fin} \psi$ [$\phi \equiv^{inf} \psi$] whenever $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \phi \Leftrightarrow (\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \psi$ for every finite [infinite] structure $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$.

[5] In contrast to most existing work in Separation Logic, we do not assume that $\mathfrak{U}$ is infinite.

As stated in the introduction, SL formulas do not admit prenex forms in general, because the quantifiers cannot be shifted outside of separating connectives. This is an essential difference with FO, where each formula is equivalent to a linear-size formula in prenex form. In particular, the equivalences $\phi * \forall x . \psi(x) \Leftrightarrow \forall x . \phi * \psi(x)$ and $\phi \mathrel{-\!\!*} \exists x . \psi(x) \Leftrightarrow \exists x . \phi \mathrel{-\!\!*} \psi(x)$ do not always hold.

Example 2. For instance, the formula $(\forall x . x \mapsto x) * \top$ is satisfiable only on universes of cardinality 1 (because $\forall x . x \mapsto x$ entails that the domain of the heap is of size 1 and contains all locations), but the formula $\forall x . (x \mapsto x * \top)$ is satisfiable if and only if the universe is finite and each location points to itself. ∎

2.3 Test formulas for SL^k

This section presents the definitions and results from [6], needed for self-containment.

Definition 3. *The following patterns are called* test formulas *of* SL^k*, for any* $k \geq 1$*:*

$$
\begin{array}{ll}
x \hookrightarrow \mathbf{y} \overset{\mathsf{def}}{=} x \mapsto \mathbf{y} * \top & |U| \geq n \overset{\mathsf{def}}{=} \top \multimap |h| \geq n,\ n \in \mathbb{N} \\
\mathsf{alloc}(x) \overset{\mathsf{def}}{=} x \mapsto \underbrace{(x, \ldots, x)}_{k\ times} \mathrel{-\!\!*} \bot & |h| \geq |U| - n \overset{\mathsf{def}}{=} |h| \geq n + 1 \mathrel{-\!\!*} \bot, n \in \mathbb{N} \\
x \approx y & |h| \geq n \overset{\mathsf{def}}{=} \begin{cases} |h| \geq n - 1 * \neg \mathsf{emp}, & if\ n > 0 \\ \top, & if\ n = 0 \end{cases}
\end{array}
$$

where $x, y \in \mathsf{Var}$*,* $\mathbf{y} \in \mathsf{Var}^k$ *is a k-tuple of variables and* $n \in \mathbb{N}$ *is a positive integer. A* literal *is a test formula or its negation and a* minterm *is any conjunction of literals.*

The semantics of test formulas is intuitive: $x \hookrightarrow \mathbf{y}$ holds when x denotes a location and $\mathbf{y}$ is the image of that location in the heap, $\mathsf{alloc}(x)$ holds when x denotes a location in the domain of the heap (allocated), $|h| \geq n$, $|U| \geq n$ and $|h| \geq |U| - n$ are cardinality constraints involving the size of the heap, denoted by $|h|$ and that of the universe, denoted by $|U|$. We recall that $|h|$ ranges over $\mathbb{N}$, whereas $|U|$ is always interpreted as a number larger than $|h|$ and possibly infinite. The truth value of the test formulas of the form $|U| \geq n$ and $|h| \geq |U| - n$ depend on the universe $\mathfrak{U}$, hence such test formulas are called *universe-dependent*. The truth value of the other test formulas depend only on the store and heap, thus they are called *universe-independent*. Clearly, all universe-dependent test formulas are trivially equivalent to true (for $|U| \geq n$) or false (for $|h| \geq |U| - n$) when interpreted over an infinite universe. Observe that not all atoms of SL^k are test formulas, for instance $x \mapsto \mathbf{y}$ and emp are not test formulas. However, it is easy to check that any atom may be written as a boolean combination of test formulas, for instance $x \mapsto \mathbf{y}$ is equivalent to $x \hookrightarrow \mathbf{y} \wedge \neg |h| \geq 2$ and emp is equivalent to $\neg |h| \geq 1$.

The following result establishes a translation of quantifier-free SL^k formulas into boolean combinations of test formulas. A *literal* is a test formula or its negation and a *minterm* is any conjunction of literals.

Lemma 4. *Given a quantifier-free* SL^k *formula* ϕ*, there exist finite sets of minterms* $\mu^{fin}(\phi)$ *and* $\mu^{inf}(\phi)$ *such that* $\phi \equiv^{fin} \bigvee_{M \in \mu^{fin}(\phi)} M$ *and* $\phi \equiv^{inf} \bigvee_{M \in \mu^{inf}(\phi)} M$*. Furthermore, the size of every* $M \in \mu^{fin}(\phi) \cup \mu^{inf}(\phi)$ *is polynomial w.r.t.* $\mathsf{size}(\phi)$*, and given a minterm* M*, the problem of checking whether* $M \in \mu^{fin}(\phi)$ *[resp.* $M \in \mu^{inf}(\phi)$*] is in* PSPACE*.*

Proof. See [6]. □

Given a quantifier-free SL^k formula ϕ, the number of minterms in $\mu^{\mathit{fin}}(\phi)$ [resp. in $\mu^{\mathit{inf}}(\phi)$] is exponential in the size of ϕ, in the worst case. An optimal decision procedure does not generate and store these sets explicitly, but rather enumerate minterms lazily.

Example 5. The formula $x \mapsto y * y \mapsto x * \neg\mathsf{emp}$ is equivalent to the minterm: $x \hookrightarrow y \wedge y \hookrightarrow x \wedge x \not\approx y \wedge |h| \geq 3$. Indeed, because the atoms $x \mapsto y$, $y \mapsto x$ and $\neg\mathsf{emp}$ must be satisfied on disjoint heaps, the initial formula entails that x, y are distinct and that the heap contains at least 3 allocated elements (x, y and an additional element distinct from x and y). The formula $x \mapsto y \mathrel{-\!\!*} x \mapsto z$ is equivalent to the disjunction of minterms $\mathsf{alloc}(x) \vee (\neg|h| \geq 1 \wedge y \approx z)$. Indeed, if x is allocated then the heap cannot be extended by a disjoint heap satisfying $x \mapsto y$ hence the separating implication trivially holds, otherwise the implication holds iff the heap is empty and $y \approx z$. ■

3 From Infinite to Finite Satisfiability

We begin by showing that for prenex SL-formulas, the infinite satisfiability problem can be reduced to the finite satisfiability problem. The intuition is that two SL-structures defined on the same heap and store can be considered as equivalent if both have enough locations outside of the heap.

Definition 6. *Let X be a set of variables and let $n \in \mathbb{N}$. Two SL-structures $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ and $\mathcal{I}' = (\mathfrak{U}', \mathfrak{s}', \mathfrak{h}')$ are* (X,n)-similar *(written $\mathcal{I} \sim^n_X \mathcal{I}'$) iff the following conditions hold:*

1. $\mathfrak{h} = \mathfrak{h}'$.
2. *For every $x \in X$, if $\mathfrak{s}(x) \in \mathrm{elems}(\mathfrak{h})$ or $\mathfrak{s}'(x) \in \mathrm{elems}(\mathfrak{h}')$ then $\mathfrak{s}(x) = \mathfrak{s}'(x)$.*
3. $\|\mathfrak{U} \setminus \mathrm{elems}(\mathfrak{h})\| \geq n + \|X\|$ *and* $\|\mathfrak{U}' \setminus \mathrm{elems}(\mathfrak{h})\| \geq n + \|X\|$.
4. *For all $x, y \in X$, $\mathcal{I} \models x \approx y$ iff $\mathcal{I}' \models x \approx y$.*

Condition 1 entails that $\mathrm{elems}(\mathfrak{h}) \subseteq \mathfrak{U} \cap \mathfrak{U}'$. We prove that any two SL-structures that are $(\mathrm{fv}(\phi), m)$-similar are indistinguishable by any formula ϕ prefixed by m quantifiers.

Proposition 7. *Let $\phi = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_m x_m \,.\, \psi$ be a prenex SL^k formula, with $\mathsf{Q}_i \in \{\forall, \exists\}$ for all $i = 1, \ldots, m$, where ψ is a quantifier-free boolean combination of universe-independent test formulas. If $\mathcal{I} \sim^m_{\mathrm{fv}(\phi)} \mathcal{I}'$ and $\mathcal{I} \models \phi$ then $\mathcal{I}' \models \phi$.*

Proof. Let $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ and $\mathcal{I}' = (\mathfrak{U}', \mathfrak{s}', \mathfrak{h}')$. Assume that $\mathcal{I} \sim^m_{\mathrm{fv}(\phi)} \mathcal{I}'$ and $\mathcal{I} \models \phi$. By Condition 1 in Definition 6 we have $\mathfrak{h} = \mathfrak{h}'$. We prove that $\mathcal{I}' \models \phi$ by induction on m.

- If $m = 0$, then we have $\phi = \psi$, we show that $\mathcal{I}$ and $\mathcal{I}'$ agree on every atomic formula in ϕ, which entails by an immediate induction that they agree on ϕ. By Condition 4 in Definition 6, we already have that $\mathcal{I}$ and $\mathcal{I}'$ agree on every atom $x \approx x'$ with $x, x' \in \mathrm{fv}(\phi)$. By Condition 1, $\mathcal{I}$ and $\mathcal{I}'$ agree on all atoms $|h| \geq n$. Consider an atom $\ell \in \{y_0 \hookrightarrow (y_1, \ldots, y_k),\ \mathsf{alloc}(y_0)\}$, with $y_0, \ldots, y_k \in \mathrm{fv}(\phi)$. If for every $i \in [\![0 \mathrel{..} k]\!]$ we have $\mathfrak{s}(y_i) \in \mathrm{elems}(\mathfrak{h})$ then by Condition 2 we deduce that $\mathfrak{s}'$ and $\mathfrak{s}$ coincide on $y_0, \ldots, y_k$ hence $\mathcal{I}$ and $\mathcal{I}'$ agree on ℓ because they share the same heap. The same holds if $\mathfrak{s}'(y_i) \in \mathrm{elems}(\mathfrak{h})$, $\forall i \in [\![0 \mathrel{..} k]\!]$. If both conditions are false, then we must have $\mathcal{I} \not\models \ell$ and $\mathcal{I}' \not\models \ell$, by definition of $\mathrm{elems}(\mathfrak{h})$, thus $\mathcal{I}$ and $\mathcal{I}'$ also agree on ℓ in this case.

- Assume that $m \geq 1$ and $Q_1 = \exists$, i.e., $\phi = \exists x_1 \ . \ \phi'$. Then there exists $e \in \mathfrak{U}$ such that $(\mathfrak{U}, \mathfrak{s}[x_1 \mapsto e], \mathfrak{h}) \models \phi'$. We construct an element $e' \in \mathfrak{U}'$ as follows. If $e = \mathfrak{s}(y)$, for some $y \in \mathrm{fv}(\phi)$, then we let $e' = \mathfrak{s}'(y)$. If $\forall y \in \mathrm{fv}(\phi), e \neq \mathfrak{s}(y)$ and if $e \in \mathrm{elems}(\mathfrak{h})$ then we let $e' = e$. Otherwise, e' is an arbitrarily chosen element in $\mathfrak{U}' \setminus (\mathfrak{s}'(\mathrm{fv}(\phi)) \cup \mathrm{elems}(\mathfrak{h}))$. Such an element necessarily exists, because by Condition 3 in Definition 6, $\mathfrak{U}'$ contains at least $m + ||\mathrm{fv}(\phi)|| \geq 1 + ||\mathfrak{s}(\mathrm{fv}(\phi))||$ elements distinct from those in $\mathrm{elems}(\mathfrak{h})$. Let $\mathcal{J} = (\mathfrak{U}, \mathfrak{s}[x_1 \mapsto e], \mathfrak{h})$ and $\mathcal{J}' = (\mathfrak{U}, \mathfrak{s}[x_1 \mapsto e], \mathfrak{h})$, we prove that $\mathcal{J} \sim^{m-1}_{\mathrm{fv}(\phi) \cup \{x_1\}} \mathcal{J}'$. This entails the required results since by the induction hypothesis we deduce $\mathcal{J}' \models \phi'$, so that $\mathcal{I}' \models \phi$.
  - Condition 1 trivially holds.
  - For Condition 2, assume that there exists a variable $x \in \mathrm{fv}(\phi) \cup \{x_1\}$ such that either $\mathfrak{s}[x_1 \mapsto e](x) \in \mathrm{elems}(\mathfrak{h})$ or $\mathfrak{s}'[x_1 \mapsto e'](x) \in \mathrm{elems}(\mathfrak{h})$, and $\mathfrak{s}[x_1 \mapsto e](x) \neq \mathfrak{s}'[x_1 \mapsto e'](x)$. Since $\mathcal{I} \sim^m_{\mathrm{fv}(\phi)} \mathcal{I}'$, if $x \in \mathrm{fv}(\phi)$ then $[\mathfrak{s}(x) \in \mathrm{elems}(\mathfrak{h}) \vee \mathfrak{s}'(x) \in \mathrm{elems}(\mathfrak{h})] \Rightarrow \mathfrak{s}(x) = \mathfrak{s}'(x)$, thus necessarily $x = x_1$. In this case, $\mathfrak{s}[x_1 \mapsto e](x) = e$ and $\mathfrak{s}'[x_1 \mapsto e'](x) = e'$. Since $e \neq e'$ by hypothesis, there can be no $y \in \mathrm{fv}(\phi)$ such that $\mathfrak{s}(y) = e$ because otherwise by construction we would have $e = \mathfrak{s}(y) = \mathfrak{s}'(y) = e'$. By definition of e' we cannot have $e \in \mathrm{elems}(\mathfrak{h})$ either, so e' is necessarily in $\mathfrak{U}' \setminus (\mathfrak{s}'(\mathrm{fv}(\phi)) \cup \mathrm{elems}(\mathfrak{h}))$ and the disjunction $e \in \mathrm{elems}(\mathfrak{h}) \vee e' \in \mathrm{elems}(\mathfrak{h})$ cannot hold.
  - Condition 3 follows from the fact that $\mathcal{I} \sim^m_{\mathrm{fv}(\phi)} \mathcal{I}'$ because we have $m - 1 + ||\mathrm{fv}(\phi) \cup \{x_1\}|| = m + ||\mathrm{fv}(\phi)||$.
  - We now establish Condition 4. Let $x, x' \in \mathrm{fv}(\phi) \cup \{x_1\}$. If $x, x' \in \mathrm{fv}(\phi)$ then $\mathfrak{s}[x_1 \mapsto e]$ and $\mathfrak{s}'[x_1 \mapsto e']$ coincide with $\mathfrak{s}$ and $\mathfrak{s}'$ respectively on x and x', hence $\mathcal{J}$ and $\mathcal{J}'$ must agree on $x \approx x'$ since $\mathcal{I} \sim^m_{\mathrm{fv}(\phi)} \mathcal{I}'$. The result also trivially holds when $x = x' = x_1$. Now assume that $x = x_1$ and $x' \neq x_1$. If $e = \mathfrak{s}(y)$ for some $y \in \mathrm{fv}(\phi)$, then $\mathcal{J} \models x \approx x'$ iff $\mathcal{I} \models y \approx x'$. By definition of e', we also have $e' = \mathfrak{s}'(y)$, hence $\mathcal{J}' \models x \approx x'$ iff $\mathcal{I}' \models y \approx x'$. Since both y and x' are in $\mathrm{fv}(\phi)$, we have $\mathcal{J} \models x \approx x' \Leftrightarrow \mathcal{I} \models y \approx x' \Leftrightarrow \mathcal{I}' \models y \approx x' \Leftrightarrow \mathcal{J}' \models x \approx x'$. If the previous condition does not hold then necessarily $e \neq \mathfrak{s}(x')$, and $\mathcal{J} \not\models x_1 \approx x'$. If $e \in \mathrm{elems}(\mathfrak{h})$, then by definition of e', we have $e' = e$. If $\mathcal{J}' \models x_1 \approx x'$ then we must have $\mathfrak{s}'(x') = \mathfrak{s}'(x_1) = e' = e \in \mathrm{elems}(\mathfrak{h})$, which by Condition 2 entails that $\mathfrak{s}'(x') = \mathfrak{s}(x') = e$, hence $\mathcal{J} \models x_1 \approx x'$, a contradiction. Finally, if $e \notin \mathrm{elems}(\mathfrak{h})$, then by definition of e', e' cannot occur in $\mathfrak{s}'(\mathrm{fv}(\phi))$, thus $\mathcal{J}' \not\models x_1 \approx x'$.
- Finally, assume that $m \geq 1$ and $Q_1 = \forall$. Then $\phi = \forall x_1 \ . \ \phi'$. Let $\phi_2 = \exists x_1 \ . \ \phi'_1$, where ϕ'_1 denotes the nnf of $\neg\phi'$. Assume that $\mathcal{I}' \not\models \phi$, then $\mathcal{I}' \models \phi_2$, because $\neg\phi \equiv \exists x_1 \ . \ \neg\phi' \equiv \exists x_1 \ . \ \phi'_1 = \phi_2$. By the previous case, using the symmetry of $\sim^m_{\mathrm{fv}(\phi)}$ and the fact that ϕ and ϕ_2 have exactly the same free variables and number of quantifiers, we have $\mathcal{I} \models \phi_2$, i.e. $\mathcal{I} \not\models \phi$, a contradiction. □

We define the following shorthands:

$$\begin{aligned} x \in h &\overset{\mathsf{def}}{=} \exists y_0, y_1, \ldots y_k \ . \ y_0 \hookrightarrow (y_1, \ldots, y_k) \wedge \textstyle\bigvee_{i=0}^{k} x \approx y_i \\ \mathsf{dist}(x_1, \ldots, x_n) &\overset{\mathsf{def}}{=} \textstyle\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{i-1} \neg(x_i \approx x_j) \\ \lambda_p &\overset{\mathsf{def}}{=} \exists x_1, \ldots, x_p \ . \ (\mathsf{dist}(x_1, \ldots, x_p) \wedge \textstyle\bigwedge_{i=1}^{p} \neg x_i \in h) \end{aligned}$$

It is clear that $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \lambda_p$ iff $||\mathfrak{U} \setminus \text{elems}(\mathfrak{h})|| \geq p$. In particular, λ_p is always true on an infinite universe. Observe, moreover, that λ_p belongs to the $\mathsf{PRE}(\mathsf{SL}^k)$ fragment, for any $p \geq 2$ and any $k \geq 1$.

The following lemma reduces the infinite satisfiability problem to the finite version of this problem. This is done by adding an axiom ensuring that there are enough locations outside of the heap. Note that there is no need to consider test formulas of the form $|U| \geq n$ [resp. $|h| \geq |U| - n$] because they always evaluate to true [resp. false] on infinite SL-structures.

Theorem 8. *Let $\phi = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_m x_m \,.\, \psi$ be a prenex SL^k formula, where $\mathsf{Q}_i \in \{\forall, \exists\}$ for $i = 1, \ldots, m$ and $\text{fv}(\phi) = \emptyset$. Assume that ψ is a boolean combination of universe-independent test formulas. The two following assertions are equivalent.*

1. *ϕ admits an infinite model.*
2. *$\phi \wedge \lambda_m$ admits a finite model.*

Proof. (1) ⇒ (2): Assume that ϕ admits an infinite model $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$. Let $\mathfrak{U}'$ be a finite subset of $\mathfrak{U}$ containing $\text{elems}(\mathfrak{h})$ and m additional elements. It is clear that $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \sim^m_\emptyset (\mathfrak{U}', \mathfrak{s}, \mathfrak{h})$. Indeed, Condition 1 holds since the two structures share the same heap, Conditions 4 and 2 trivially hold since the considered set of variables is empty, and Condition 3 holds since $\mathfrak{U}$ is infinite and the additional elements in $\mathfrak{U}'$ do not occur in $\text{elems}(\mathfrak{h})$. Thus $(\mathfrak{U}', \mathfrak{s}, \mathfrak{h}) \models \phi$ by Proposition 7, and $(\mathfrak{U}', \mathfrak{s}, \mathfrak{h}) \models \lambda_m$, by definition of $\mathfrak{U}'$.

(2) ⇒ (1): Assume that $\phi \wedge \lambda_m$ has a finite model $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$. Let $\mathfrak{U}'$ be any infinite set containing $\mathfrak{U}$. Again, we have $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \sim^m_\emptyset (\mathfrak{U}', \mathfrak{s}, \mathfrak{h})$. As in the previous case, Conditions 1, 4 and 2 trivially hold, and Condition 3 holds since $\mathfrak{U}'$ is infinite and $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \models \lambda_m$. By Proposition 7, we deduce that $(\mathfrak{U}', \mathfrak{s}, \mathfrak{h}) \models \phi$. □

4 $\mathsf{PRE}(\mathsf{SL}^1)$ is Decidable but Not Elementary Recursive

Using Lemma 4 and Theorem 8 we shall prove that the satisfiability problem is decidable for the prenex fragment of SL^1. This shows that $\mathsf{PRE}(\mathsf{SL}^1)$ is strictly less expressive than SL^1, because SL^1 has an undecidable satisfiability problem [2]. For this purpose, we first define a translation of quantified boolean combination of test formulas into FO that is sat-preserving on finite structures. Let $\mathfrak{d}$ be a unary predicate symbol and for $i = 1, \ldots, k$, let $\mathfrak{f}_i$ be a unary function symbol. We define the following transformation from quantified boolean combinations of test formulas into first order formulas:

$$\begin{aligned}
\Theta(x \approx y) &\stackrel{\text{def}}{=} x \approx y \\
\Theta(x \hookrightarrow (y_1, \ldots, y_k)) &\stackrel{\text{def}}{=} \mathfrak{d}(x) \wedge \textstyle\bigwedge_{i=1}^k y_i \approx \mathfrak{f}_i(x) \\
\Theta(\mathsf{alloc}(x)) &\stackrel{\text{def}}{=} \mathfrak{d}(x) \\
\Theta(|U| \geq n) &\stackrel{\text{def}}{=} \exists x_1, \ldots, x_n \,.\, \mathsf{dist}(x_1, \ldots, x_n) \\
\Theta(|h| \geq n) &\stackrel{\text{def}}{=} \exists x_1, \ldots, x_n \,.\, \mathsf{dist}(x_1, \ldots, x_n) \wedge \textstyle\bigwedge_{i=1}^n \mathfrak{d}(x_i) \\
\Theta(|h| \geq |U| - n) &\stackrel{\text{def}}{=} \exists x_1, \ldots, x_n \forall y \,.\, \textstyle\bigwedge_{i=1}^n y \not\approx x_i \rightarrow \mathfrak{d}(y) \\
\Theta(\neg\phi) &\stackrel{\text{def}}{=} \neg\Theta(\phi) \\
\Theta(\phi_1 \wedge \phi_2) &\stackrel{\text{def}}{=} \Theta(\phi_1) \wedge \Theta(\phi_2) \\
\Theta(\exists x \,.\, \phi) &\stackrel{\text{def}}{=} \exists x \,.\, \Theta(\phi)
\end{aligned}$$

Proposition 9. *Let ϕ be a quantified boolean combination of test formulas. The formula ϕ has a finite* SL *model if and only if $\Theta(\phi)$ has a finite* FO *model.*

Proof. An FO-structure $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{i})$ on the signature $\mathfrak{d}$, $\mathfrak{f}_1,\ldots,\mathfrak{f}_k$ *corresponds* to an SL-structure $\mathcal{I}' = (\mathfrak{U}', \mathfrak{s}', \mathfrak{h})$ iff $\mathfrak{U} = \mathfrak{U}'$, $\mathfrak{s} = \mathfrak{s}'$, $\mathfrak{d}^{\mathfrak{i}} = \mathsf{dom}(\mathfrak{h})$ and for every $j \in [\![1\,..\,k]\!]$, $\mathfrak{f}^{\mathfrak{i}}_j(x) = y_j$ if $\mathfrak{h}(x) = (y_1,\ldots,y_k)$. It is clear that for every finite first-order structure $\mathcal{I}$ there exists a finite SL-structure $\mathcal{I}'$ such that $\mathcal{I}$ corresponds to $\mathcal{I}'$ and vice-versa. Furthermore, if $\mathcal{I}$ corresponds to $\mathcal{I}'$ then it is straightforward to check that $\mathcal{I}' \models \phi \Leftrightarrow \mathcal{I} \models \Theta(\phi)$. □

If ϕ is an SL^1 formula, then clearly $\Theta(\phi)$ is in FO^1, with one monadic boolean function symbol $\mathfrak{d}$ and one function symbol $\mathfrak{f}_1$ of sort $\sigma(f) = U$. This yields the following result:

Theorem 10. *The finite and infinite satisfiability problems are decidable for* $\mathsf{PRE}(\mathsf{SL}^1)$.

Proof. Given a formula $\psi = Q_1x_1 \ldots Q_nx_n \,.\, \phi$ of $\mathsf{PRE}(\mathsf{SL}^1)$, where ϕ is quantifier-free, let $\mu \overset{\mathsf{def}}{=} \bigvee_{M \in \mu^{inf}(\phi)} M$ be the infinite-domain equivalent expansion of ϕ as a disjunction of minterms. We have $\psi \equiv^{inf} Q_1x_1 \ldots Q_nx_n \,.\, \mu$ (Lemma 4) and $Q_1x_1 \ldots Q_nx_n \,.\, \mu$ admits an infinite model if and only if $Q_1x_1 \ldots Q_nx_n \,.\, \mu \wedge \lambda_n$ admits a finite model (Theorem 8; note that μ contains no occurrence of universe-dependent formulas, as such formulas are always true or false in infinite universes). But $Q_1x_1 \ldots Q_nx_n \,.\, \mu \wedge \lambda_n$ has a finite SL model if and only if $\Theta(Q_1x_1 \ldots Q_nx_n \,.\, \mu \wedge \lambda_n)$ has a finite FO model (Proposition 9). Since the latter formula belongs to FO^1, its finite satisfiability problem is decidable (Theorem 1). The finite case is similar. □

The complexity lower bound is established thanks to the following proposition.

Proposition 11. *There is a polynomial reduction of the finite satisfiability problem for first-order formulas with one monadic function symbol f and no predicate symbols other than $\approx$ to the finite [resp. infinite] satisfiability problem for quantified boolean quantifications of test formulas in* SL^1.

Proof. By flattening we may assume that all the equations occurring in the considered first-order formula are of the form $f(x) \approx y$ or $x \approx y$, where x,y are variables. For finite domains, the reduction is immediate: it suffices to add the axiom $\forall x \,.\, \mathsf{alloc}(x)$, stating that the heap is a total function, and to replace all equations of the form $f(x) \approx y$ by $x \hookrightarrow y$. It is straightforward to check that satisfiability is preserved (f is encoded in the heap). For infinite domains, it is not possible to add the axiom $\forall x \,.\, \mathsf{alloc}(x)$ as the resulting formula is unsatisfiable[6], so the first-order formula is translated on one that holds on the (finite) domain of the heap. We thus add the axiom $\neg\mathsf{emp} \wedge \forall x,y \,.\, x \hookrightarrow y \rightarrow \mathsf{alloc}(y)$, and we replace every quantification $\forall x \,.\, \phi$ (resp. $\exists x \,.\, \phi$) by a quantification over the domain of the heap: $\forall x \,.\, \mathsf{alloc}(x) \rightarrow \phi$ (resp. $\exists x \,.\, \mathsf{alloc}(x) \wedge \phi$). It is straightforward to check that satisfiability is preserved. Note that infinite satisfiability is equivalent to finite satisfiability, since the quantifications range over elements occurring in the heap. □

[6] Since the domain of the heap is finite.

Note there is no obvious reduction from the usual first-order satisfiability problem (i.e., on arbitrary models), because the heap is always finite in SL-structures. This explains why we had to refine in Theorem 1 the complexity lower bound from [1] to cope with finite satisfiability.

Theorem 12. *The finite and infinite satisfiability problems are not elementary recursive for* $\mathsf{PRE}(\mathsf{SL}^1)$.

Proof. The proof follows immediately from the lower bound complexity result of Theorem 1 and from the reductions in Proposition 11. □

5 The $\mathsf{BSR}(\mathsf{SL}^1)$ Fragment is PSPACE-complete

The last result concerns the tight complexity of the $\mathsf{BSR}(\mathsf{SL}^1)$ fragment. For $k \geq 2$, we showed that $\mathsf{BSR}(\mathsf{SL}^k)$ is undecidable, in general, and PSPACE-complete if the positive occurrences of the magic wand are forbidden[7] [6]. Here we show that $\mathsf{BSR}(\mathsf{SL}^1)$ is PSPACE-complete. The result does not directly follow from the Σ_2^p-complexity of the satisfiability problem for $\exists^*\forall^*$ first-order formulas with one unary function symbol[8] because only partial finite functions are considered in our context. The proof is based on the following definitions and results.

Definition 13. *A model* $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ *of a formula* φ *is* minimal *if* φ *admits no model of the form* $(\mathfrak{U}', \mathfrak{s}', \mathfrak{h}')$ *with* $\mathfrak{U}' \subsetneq \mathfrak{U}$.

Proposition 14. *Let* $\varphi = \forall y_1, \ldots, y_m \,.\, \phi$ *be a prenex formula with free variables* $x_1, \ldots, x_n$ *(with* $n > 0$*) where* ϕ *is a boolean combination of universe-independent test formulas, and let* $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ *be a minimal model of* φ*. Then* $\mathfrak{U} = \{\mathfrak{h}^j(\mathfrak{s}(x_i)) \mid i \in [\![1 \,..\, n]\!], j \in \mathbb{N}\}$.

Proof. Let $\mathfrak{U}' = \{\mathfrak{h}^j(\mathfrak{s}(x_i)) \mid i \in [\![1 \,..\, n]\!], j \in \mathbb{N}\}$ and assume that $\mathfrak{U}' \neq \mathfrak{U}$; note that $\mathfrak{U}' \neq \emptyset$ since $n > 0$. Let $\mathfrak{s}'$ be a store on $\mathfrak{U}'$ coinciding with $\mathfrak{s}$ on $x_1, \ldots, x_n$ and let $\mathfrak{h}'$ be the restriction of $\mathfrak{h}$ to $\mathfrak{U}'$. Both $\mathfrak{s}'$ and $\mathfrak{h}'$ are well-defined by construction of $\mathfrak{U}'$, and $\mathfrak{h}'$ is a heap on $\mathfrak{U}'$. Since $\mathfrak{U}$ is minimal, $(\mathfrak{U}', \mathfrak{s}', \mathfrak{h}') \not\models \varphi$, thus there exist $b_1, \ldots, b_m \in \mathfrak{U}'$ such that by letting $\mathfrak{s}'_1 \overset{\text{def}}{=} \mathfrak{s}'[y_i \mapsto b_i \mid i \in [\![1 \,..\, m]\!]]$, we have $(\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h}') \models \neg\phi$. Since the atomic formulas in ϕ are universe-independent, we deduce that $(\mathfrak{U}, \mathfrak{s}'_1, \mathfrak{h}') \models \neg\phi$. Further, $\mathfrak{s}'_1$ and $\mathfrak{s}[y_i \mapsto b_i \mid i \in [\![1 \,..\, m]\!]]$ coincide on all the variables $x_1, \ldots, x_n, y_1, \ldots, y_m$ that are free in ϕ, thus $(\mathfrak{U}, \mathfrak{s}[y_i \mapsto b_i \mid i \in [\![1 \,..\, m]\!]], \mathfrak{h}') \models \neg\phi$. Finally, $\mathfrak{h}$ and $\mathfrak{h}'$ coincide on every element of $\mathfrak{U}'$ and by definition we have $\mathfrak{s}(x_i), b_j \in \mathfrak{U}'$ for $i \in [\![1 \,..\, n]\!]$ and $j \in [\![1 \,..\, m]\!]$, hence $(\mathfrak{U}, \mathfrak{s}[y_i \mapsto b_i \mid i \in [\![1 \,..\, m]\!]], \mathfrak{h}) \models \neg\phi$, and $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \not\models \varphi$, which contradicts our assumption.

Definition 15. *Let* φ *be a formula with free variables* $x_1, \ldots, x_n$ *and let* $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ *be a structure. A* line *for* $(\mathcal{I}, \varphi)$ *is a sequence of pairwise distinct elements* $a_1, \ldots, a_\ell$ *in* $\mathfrak{U}$ *such that:*

1. $\forall i \in [\![1 \,..\, \ell - 1]\!]$, $a_{i+1} = \mathfrak{h}(a_i)$.

[7] For infinite satisfiability, it is enough to forbid positive occurrences of the magic wand containing universally quantified variables only.

[8] See [1, Theorem 6.4.19].

2. *$\forall i \in \llbracket 1 \mathinner{..} \ell - 1 \rrbracket$, $\forall e \in \mathfrak{U}$, if $\mathfrak{h}(e) = a_{i+1}$ then $e = a_i$.*
3. *$\forall i \in \llbracket 1 \mathinner{..} \ell \rrbracket$, $\forall j \in \llbracket 1 \mathinner{..} n \rrbracket$, $a_i \neq \mathfrak{s}(x_j)$.*

The next proposition shows that there is a bound on the length of any line in a minimal model.

Proposition 16. *Let $\varphi = \forall y_1, \ldots, y_m \,.\, \phi$ be a prenex formula with free variables $x_1, \ldots, x_n$ where ϕ is a boolean combination of domain-independent test formulas, and let $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ be a model of φ. If $(\mathcal{I}, \varphi)$ admits a line of length strictly greater than $m+2$ then $\mathcal{I}$ is not minimal.*

Proof. Let $a_1, \ldots, a_l$ be a sequence of elements satisfying the conditions of Definition 15 with $l > m+2$. Let $\mathcal{I}' = (\mathfrak{U}', \mathfrak{s}', \mathfrak{h}')$, where $\mathfrak{U}' \overset{\text{def}}{=} \mathfrak{U} \setminus \{a_2\}$, $\mathfrak{s}'$ is a store on $\mathfrak{U}'$ coinciding with $\mathfrak{s}$ on all variables x such that $\mathfrak{s}(x) \in \mathfrak{U}'$, $\mathsf{dom}(\mathfrak{h}') \overset{\text{def}}{=} \mathsf{dom}(\mathfrak{h}) \setminus \{a_2\}$, $\mathfrak{h}'(a_1) \overset{\text{def}}{=} a_3$ and $\mathfrak{h}'(x) \overset{\text{def}}{=} \mathfrak{h}(x)$ if $x \in \mathsf{dom}(\mathfrak{h}') \setminus \{a_1\}$. Note that $\mathfrak{s}$ and $\mathfrak{s}'$ coincide on all variables $x_1, \ldots, x_n$ free in φ since $\forall i \in \llbracket 1 \mathinner{..} n \rrbracket$, $a_2 \neq \mathfrak{s}(x_i)$, by Definition 15 (3). Since $\mathcal{I}$ is minimal, necessarily $\mathcal{I}' \not\models \varphi$, thus there exist $b'_1, \ldots, b'_m \in \mathfrak{U}'$ such that by letting $\mathfrak{s}'_1 \overset{\text{def}}{=} \mathfrak{s}'[y_j \mapsto b'_j \mid j \in \llbracket 1 \mathinner{..} m \rrbracket]$, we have $(\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h}') \models \neg\phi$. Since $l > m+2$ and $a_1, \ldots, a_l$ are distinct by Definition 15, there exists $i \in \llbracket 2 \mathinner{..} l-1 \rrbracket$ such that $a_{i+1} \notin \{b'_1, \ldots, b'_m\}$. We define a sequence $b_1, \ldots, b_m \in \mathfrak{U}$ as follows. For every $j \in \llbracket 1 \mathinner{..} m \rrbracket$, if there exists $o \in \llbracket 3 \mathinner{..} i \rrbracket$ such that $b'_j = a_o$, then we let $b_j \overset{\text{def}}{=} a_{o-1}$; otherwise, $b_j \overset{\text{def}}{=} b'_j$. Note that b_j is well-defined, because $a_1, \ldots, a_l$ are distinct, hence there exists at most one o satisfying the above condition.

We emphasize some useful consequences of the above definitions before proving that $(\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h}') \models \phi$. Let $V = \{x_1, \ldots, x_n\} \cup \{y_j \mid j \in \llbracket 1 \mathinner{..} m \rrbracket, \mathfrak{s}'_1(y_j) \notin \{a_3, \ldots, a_i\}\}$. By definition $\mathfrak{s}$ and $\mathfrak{s}'$ coincide on $x_1, \ldots, x_n$, and $\mathfrak{s}_1(y_j) = b_j = b'_j = \mathfrak{s}'_1(y_j)$ if $b'_j \notin \{a_3, \ldots, a_i\}$, hence $\mathfrak{s}'_1$ and $\mathfrak{s}[y_j \mapsto b_j \mid j \in \llbracket 1 \mathinner{..} m \rrbracket]$ coincide on every variable in V. Furthermore, for every variable $x \in V$, $\mathfrak{s}_1(x) \in \mathfrak{U} \setminus \{a_2, \ldots, a_i\}$. Indeed, either $x \in \{x_1, \ldots, x_n\}$ and in this case $\mathfrak{s}(x) \notin \{a_1, \ldots, a_n\}$ by Definition 15 (3); or $x = y_j$ for some $j \in \llbracket 1 \mathinner{..} m \rrbracket$, and then $\mathfrak{s}_1(x) = b_j = b'_j \notin \{a_3, \ldots, a_i\}$, so that $\mathfrak{s}_1(x) \in \mathfrak{U}' \setminus \{a_3, \ldots, a_i\} = \mathfrak{U} \setminus \{a_2, \ldots, a_i\}$. Finally, if x occurs in ϕ and $x \notin V$ then $x = y_j$ for some $j \in \llbracket 1 \mathinner{..} m \rrbracket$ such that $b'_j = a_o$, with $o \in \llbracket 3 \mathinner{..} i \rrbracket$, thus $\mathfrak{s}'_1(x) = a_o$ and $\mathfrak{s}_1(x) = b_j = a_{o-1}$, and therefore $\mathfrak{s}'_1(x) \in \{a_3, \ldots, a_i\}$ and $\mathfrak{s}_1(x) \in \{a_2, \ldots, a_{i-1}\}$. Let $\mathfrak{s}_1 \overset{\text{def}}{=} \mathfrak{s}[y_j \mapsto b_j \mid j \in \llbracket 1 \mathinner{..} m \rrbracket]$; we show that $(\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h})$ and $(\mathfrak{U}, \mathfrak{s}_1, \mathfrak{h})$ coincide on every test formula ℓ in ϕ.

$\ell = x \approx y$. If $x, y \in V$ then the proof is immediate since $\mathfrak{s}_1$ and $\mathfrak{s}'_1$ coincide on x and y. If $x \in V$ and $y \notin V$ then $\mathfrak{s}_1(x) = \mathfrak{s}'_1(x) \in \mathfrak{U} \setminus \{a_2, \ldots, a_i\}$ and $\mathfrak{s}_1(y), \mathfrak{s}'_1(y) \in \{a_2, \ldots, a_i\}$ hence $x \approx y$ is false in both structures. The proof is symmetric if $x \notin V$ and $y \in V$. If $x, y \notin V$ then $\mathfrak{s}'_1(x) = a_o$, $\mathfrak{s}'_1(y) = a_{o'}$, with $\mathfrak{s}_1(x) = a_{o-1}$ and $\mathfrak{s}_1(y) = a_{o'-1}$. Since the $a_1, \ldots, a_l$ are pairwise distinct we have $\mathfrak{s}'_1(x) = \mathfrak{s}'_1(y) \Leftrightarrow o = o' \Leftrightarrow o-1 = o'-1 \Leftrightarrow \mathfrak{s}_1(x) = \mathfrak{s}_1(y)$.

$\ell = \mathsf{alloc}(x)$. If $x \in V$ then $\mathfrak{s}_1(x) = \mathfrak{s}'_1(x) \neq a_2$ Thus $\mathfrak{s}_1(x) \in \mathsf{dom}(\mathfrak{h}) \Leftrightarrow \mathfrak{s}'_1(x) \in \mathsf{dom}(\mathfrak{h}) \Leftrightarrow \mathfrak{s}'_1(x) \in \mathsf{dom}(\mathfrak{h}')$. If $x \notin V$ then $\mathfrak{s}'_1(x) \in \{a_3, \ldots, a_i\}$ and $\mathfrak{s}_1(x) \in \{a_2, \ldots, a_{i-1}\}$ (with $i < l$) thus $\mathsf{alloc}(x)$ is true in both structures.

$\ell = x \hookrightarrow y$. We distinguish several cases.

- If $x, y \in V$ then $\mathfrak{s}_1(x) = \mathfrak{s}'_1(x)$ and $\mathfrak{s}_1(y) = \mathfrak{s}'_1(y)$, with $\mathfrak{s}_1(x) \neq a_2$, hence $\mathfrak{h}(\mathfrak{s}_1(x)) = \mathfrak{s}_1(y) \Leftrightarrow \mathfrak{h}(\mathfrak{s}'_1(x)) = \mathfrak{s}'_1(y) \Leftrightarrow \mathfrak{h}'(\mathfrak{s}'_1(x)) = \mathfrak{s}'_1(y)$, thus $(\mathfrak{U}, \mathfrak{s}_1, \mathfrak{h}) \models \ell \Leftrightarrow (\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h}') \models \ell$.
- If $x, y \notin V$ then $\mathfrak{s}'_1(x) = a_o$, $\mathfrak{s}'_1(y) = a_{o'}$, with $\mathfrak{s}_1(x) = a_{o-1}$, $\mathfrak{s}_1(y) = a_{o'-1}$ and $o, o' \geq 3$ thus $\mathfrak{h}'(\mathfrak{s}'_1(x)) = \mathfrak{s}'_1(y) \Leftrightarrow o = o'-1 \Leftrightarrow \mathfrak{h}(\mathfrak{s}_1(x)) = \mathfrak{s}_1(y)$.

- If $x \in V$ and $y \notin V$, then $\mathfrak{s}'_1(y) = a_o$ with $\mathfrak{s}_1(y) = a_{o-1}$ and $o \in \llbracket 3 \mathinner{..} i \rrbracket$. We distinguish two cases. If $x \in \{x_1, \dots, x_n\}$, then $\mathfrak{h}(\mathfrak{s}_1(x)) \notin \{a_1, \dots, a_l\}$ (by Definition 15 (2)) thus $\mathfrak{h}(\mathfrak{s}_1(x)) = \mathfrak{h}'(\mathfrak{s}'_1(x)) \neq \mathfrak{s}_1(y), \mathfrak{s}'_1(y)$ and ℓ is false in both structures. Otherwise, $x = y_j$, for some $j \in \llbracket 1 \mathinner{..} m \rrbracket$ such that $b'_j \notin \{a_3, \dots, a_i\}$. If $b'_j = a_1$ then $\mathfrak{h}(\mathfrak{s}_1(x)) = a_2$ and $\mathfrak{h}'(\mathfrak{s}'_1(x)) = a_3$, thus $\mathfrak{h}(\mathfrak{s}_1(x)) = \mathfrak{s}_1(y) \Leftrightarrow a_2 = \mathfrak{s}_1(y) \Leftrightarrow a_2 = a_{o-1} \Leftrightarrow o = 3 \Leftrightarrow a_3 = \mathfrak{s}'_1(y) \Leftrightarrow \mathfrak{h}'(\mathfrak{s}'_1(x)) = \mathfrak{s}'(y)$, hence ℓ has the same truth value in $(\mathfrak{U}, \mathfrak{s}_1, \mathfrak{h})$ and $(\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h}')$. If $b_j \neq a_1$ then $\mathfrak{h}'(\mathfrak{s}_1(x)) = \mathfrak{h}(\mathfrak{s}_1(x))$, and $\mathfrak{s}_1(x) \notin \{a_1, \dots, a_i\}$, thus $\mathfrak{h}(\mathfrak{s}_1(x) \notin \{a_2, \dots, a_{i+1}\}$, hence ℓ is false in both structures.
- If $y \in V$ and $x \notin V$ then there exists $o \in \llbracket 3 \mathinner{..} i \rrbracket$ such that $\mathfrak{s}_1(x) = a_{o-1}$ and $\mathfrak{s}'_1(x) = a_o$, with $\mathfrak{s}_1(y) = \mathfrak{s}'_1(y) \notin \{a_2, \dots, a_i\}$. We have $\mathfrak{h}'(\mathfrak{s}'_1(x)) = a_{o+1}$ and $\mathfrak{h}(\mathfrak{s}_1(x)) = a_o$, thus $\mathfrak{h}'(\mathfrak{s}'_1(x)), \mathfrak{h}(\mathfrak{s}_1(x)) \in \{a_3, \dots, a_{i+1}\}$. By definition of i, $a_{i+1} \notin \{b_1, \dots, b_m\}$ (since $a_{i+1} \notin \{b'_1, \dots, b'_m\}$ and $i+1 > i$), moreover $a_{i+1} \notin \mathfrak{s}(\{x_1, \dots, x_n\})$ by Definition 15 (3). Thus $a_{i+1} \neq \mathfrak{s}_1(y)$. Since $\mathfrak{s}_1(y) \notin \{a_2, \dots, a_i\}$ we deduce that $\mathfrak{s}_1(y) \notin \{a_3, \dots, a_{i+1}\}$, thus ℓ is false in both structures.

As a consequence, $(\mathfrak{U}', \mathfrak{s}'_1, \mathfrak{h}')$ and $(\mathfrak{U}, \mathfrak{s}_1, \mathfrak{h})$ necessarily coincide on ϕ, and consequently $(\mathfrak{U}, \mathfrak{s}_1, \mathfrak{h}) \models \neg\phi$, hence $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h}) \not\models \forall y_1, \dots, y_m \,.\, \phi$ which contradicts our hypothesis. □

Lemma 17. *Let $\varphi = \forall y_1, \dots, y_m \,.\, \phi$ be a prenex formula of SL^1 of free variables $x_1, \dots, x_n$ (with $n > 0$) where ϕ is a boolean combination of universe-independent test formulas. If $(\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ is a finite minimal model of φ then $||\mathfrak{U}|| \leq 2n \cdot (m+3)$.*

Proof. Let $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ be a minimal finite model of φ and let $a_i = \mathfrak{s}(x_i)$ for $i = 1, \dots, n$. We inductively define a sequence l_i ($1 \leq i \leq n$) of natural numbers as follows: l_i is the minimal natural number such that either $\mathfrak{h}^{l_i}(a_i) \notin \mathsf{dom}(\mathfrak{h})$ or $\mathfrak{h}(\mathfrak{h}^{l_i}(a_i)) \in \{a_1, \dots, a_n\} \cup \{\mathfrak{h}^j(a_i) \mid j \in \llbracket 1 \mathinner{..} l_i - 1 \rrbracket\} \cup \{\mathfrak{h}^j(a_k) \mid k \in \llbracket 1 \mathinner{..} i-1 \rrbracket, j \in \llbracket 1 \mathinner{..} l_k \rrbracket\}$. Because the domain of $\mathfrak{h}$ is finite, the numbers l_i always exist, for all $i = 1, \dots, n$. Note that by construction, given $i \in \llbracket 1 \mathinner{..} i \rrbracket$, if $\mathfrak{h}^j(a_i) \neq \mathfrak{h}^k(a_i)$ for all $k < j$ and $\mathfrak{h}^j(a_i) \notin \left\{\mathfrak{h}^k(a_p) \middle| k \in \mathbb{N}\right\}$ for all $p < i$, then $j \leq l_i$. Hence, since by Proposition 14, we have $\mathfrak{U} = \{\mathfrak{h}^j(\mathfrak{s}(x_i)) \mid i \in \llbracket 1 \mathinner{..} n \rrbracket, j \in \mathbb{N}\}$, we deduce that $\mathfrak{U} = \bigcup_{i=1}^n \{\mathfrak{h}^j(a_i) \mid j \in \llbracket 0 \mathinner{..} l_i \rrbracket\}$. Furthermore, by definition of l_i, all locations $\mathfrak{h}^j(a_i)$, for $i \in \llbracket 1 \mathinner{..} n \rrbracket$ and $j \in \llbracket 1 \mathinner{..} l_i \rrbracket$, are pairwise distinct.

We define the following subsets of $\mathfrak{U}$: $\mathfrak{U}_1 \overset{\mathsf{def}}{=} \{a_i \mid i \in \llbracket 1 \mathinner{..} n \rrbracket\}$, $\mathfrak{U}_2 \overset{\mathsf{def}}{=} \{\mathfrak{h}^{l_i}(a_i) \mid i \in \llbracket 1 \mathinner{..} n \rrbracket, \mathfrak{h}^{l_i}(a_i) \notin \mathsf{dom}(\mathfrak{h})\}$, and $\mathfrak{U}_3 \overset{\mathsf{def}}{=} \{\mathfrak{h}(\mathfrak{h}^{l_i}(a_i)) \mid i \in \llbracket 1 \mathinner{..} n \rrbracket, \mathfrak{h}^{l_i}(a_i) \in \mathsf{dom}(\mathfrak{h})\}$. By definition, $\mathfrak{U}_2 \cup \mathfrak{U}_3$ contains at most n elements, thus $||\mathfrak{U}_1 \cup \mathfrak{U}_2 \cup \mathfrak{U}_3|| \leq 2n$. We have that every element c such that there exist $a \neq b$ with $\mathfrak{h}(a) = \mathfrak{h}(b) = c$ is in $\mathfrak{U}_3$. Indeed, assume that there exist two such elements $a, b \in \mathfrak{U}$. Then there exist $i, j \in \llbracket 1 \mathinner{..} n \rrbracket$, $i' \in \llbracket 0 \mathinner{..} l_i \rrbracket$, $j' \in \llbracket 0 \mathinner{..} l_j \rrbracket$ such that $a = \mathfrak{h}^{i'}(a_i)$ and $b = \mathfrak{h}^{j'}(a_j)$. We assume by symmetry that $i \leq j$. Then by definition of l_j we must have $j' = l_j$, so that $\mathfrak{h}(b) = c \in \mathfrak{U}_3$. The reader may refer to Figure 1 for an illustration. Now, consider a sequence of the form $(\mathfrak{h}^j(a_i), \dots, \mathfrak{h}^{j'}(a_i))$ (with $j \leq j'$) containing no element in $\mathfrak{U}_1 \cup \mathfrak{U}_2 \cup \mathfrak{U}_3$. By definition, this sequence fulfills Conditions 3 and 1 from Definition 15. If the sequence does not fulfill Condition 2, then there exist $k \in \llbracket j \mathinner{..} j'-1 \rrbracket$ such that $\mathfrak{h}^{k+1}(a_i)$ is a fork element, hence $\mathfrak{h}^{k+1}(a_i) \in \mathfrak{U}_3$, which contradicts our hypothesis. Consequently, $(\mathfrak{h}^j(a_i), \dots, \mathfrak{h}^{j'}(a_i))$ is a line for $(\mathcal{I}, \varphi)$. By Proposition 16 such lines cannot be of length greater than $m+2$, therefore $\mathfrak{U} \setminus (\mathfrak{U}_1 \cup \mathfrak{U}_2 \cup \mathfrak{U}_3)$ contains at most $(m+2) \cdot L$ elements, where L is the number of sequences $(\mathfrak{h}^j(a_i), \dots, \mathfrak{h}^{j'}(a_i))$ of maximal length not containing elements in $\mathfrak{U}_1 \cup \mathfrak{U}_2 \cup \mathfrak{U}_3$.

Thus $||\mathfrak{U}|| \leq (m+2) \cdot L + 2n$. By definition, all such sequences necessarily start by some element $\mathfrak{h}(a)$, where $a \in \mathfrak{U}_1 \cup \mathfrak{U}_2 \cup \mathfrak{U}_3$, thus there are at most $||\mathfrak{U}_1 \cup \mathfrak{U}_2 \cup \mathfrak{U}_3|| \leq 2n$ such sequences. Hence $L \leq 2n$ and $||\mathfrak{U}|| \leq 2n \cdot (m+3)$. □

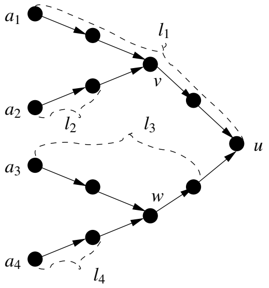

Fig. 1. Heap decomposition example. We have $l_1 = 5$, $l_2 = 1$, $l_3 = 3$ and $l_4 = 1$. Moreover, $\mathfrak{U}_1 = \{a_1, a_2, a_3, a_4\}$, $\mathfrak{U}_2 = \{u\}$ and $\mathfrak{U}_3 = \{u, v, w\}$.

Corollary 18. *The finite and infinite satisfiability problems for formulas of* BSR(SL^1) *are PSPACE-complete.*

Proof. PSPACE-hardness follows from the proof that satisfiability of the quantifier free fragment of SL^2 is PSPACE-complete [4, Proposition 5]. Indeed, this proof does not depend on the universe being infinite or the fact that $k = 2$. There remains to show PSPACE-membership for both problems. Observe that this does not directly follow from Lemmas 4 and 17, because (i) the sets $\mu^{inf}(\phi)$ and $\mu^{fin}(\phi)$ are of exponential size hence no efficient algorithm can compute them and, (ii) Lemma 17 only holds for universe-independent formulas. W.l.o.g., we assume that the considered formula contains at least one free variable and is of the form $\forall y_1, \ldots, y_m \,.\, \phi$. It is sufficient to focus on the finite satisfiability problem. Indeed, by Lemma 4, $\forall y_1, \ldots, y_m \,.\, \phi \equiv^{inf} \bigvee_{M \in \mu^{inf}(\neg\phi)} M$. By Theorem 8, $\forall y_1, \ldots, y_m \,.\, \phi$ has an infinite model iff $\forall y_1, \ldots, y_m \,.\, \phi \wedge \lambda_{n+m}$ has a finite model, where the size of λ_{n+m} is quadratic in $n+m$. Moreover, since λ_{n+m} is a BSR(SL) formula, $\forall y_1, \ldots, y_m \,.\, \phi \wedge \lambda_{n+m}$ is also a BSR(SL) formula. Hence infinite satisfiability can be reduced polynomially to finite satisfiability.

Let $\psi = \bigvee_{M \in \mu^{fin}(\neg\phi)} M$ (note that the size of ψ is exponential w.r.t. that of ϕ). Let L be the maximal number l such that a test formula $|h| \leq l$ or $|h| \leq |U| - l$ occurs in $\mu^{inf}(\phi)$. By Lemma 4, the number L is polynomial w.r.t. size(ϕ). We guess a structure $\mathcal{I} = (\mathfrak{U}, \mathfrak{s}, \mathfrak{h})$ and check that it is a model of φ as follows. We first guess the set C of literals of the form $|U| \leq i$, $|U| < i$, $|h| \leq i$, $|h| > i$, $|h| \leq |U| - i$, or $|h| > |U| - i$ with $i \in [\![0 \,..\, L]\!]$ that are true in $\mathcal{I}$. It is clear that φ is satisfiable iff $\varphi \cup C$ is satisfiable for some such set C. Up to redundancy, C contains at most 6 literals (one literal of each kind). With each test formula $\ell \in C$ we may associate an equivalent formula $\gamma(\ell)$ in BSR(SL1) built on atoms $x \approx y$ or alloc(x) using the following equivalence statements:

- $|h| \leq i \;\Leftrightarrow\; \forall x'_1, \ldots, x'_{i+1} \,.\, \mathsf{dist}(x'_1, \ldots, x'_{i+1}) \rightarrow \bigvee_{j=1}^{i+1} \neg\mathsf{alloc}(x'_j)$,
- $|h| \leq |U| - i \;\Leftrightarrow\; \exists x'_1, \ldots, x'_i \,.\, \mathsf{dist}(x'_1, \ldots, x'_i) \wedge \bigwedge_{j=1}^{i} \neg\mathsf{alloc}(x_j)$,

- $|U| \leq i \Leftrightarrow \forall x'_1,\dots,x'_{i+1}\ \neg\mathsf{dist}(x'_1,\dots,x'_{i+1})$.

Let ϑ be the conjunction of all formulas $\gamma(\ell)$ where $\ell \in C$. Note that ϑ contains (up to redundancy) at most $3L+2$ existential variables and $3L+2$ universal variables. Now consider the formula ψ' obtained from ψ by replacing every test formula such that $\ell \in C$ (resp. $\overline{\ell} \in C$) by $\top$ (resp. $\bot$). Let φ' be the formula obtained by putting $\forall y_1,\dots,y_m\ .\ \neg\psi' \wedge \vartheta$ in prenex form. It is clear that φ' is in $\mathsf{BSR}(\mathsf{SL}1)$ and that all test formulas in φ' are universe-independent, furthermore φ' contains at most $n' = n+(3L+2)$ free or existential variables and $m' = m+(3L+2)$ universal variables. Moreover, $\varphi' \equiv \varphi \wedge \vartheta$, hence φ' is satisfiable iff φ admits a model satisfying C. By Lemma 17, φ' is satisfiable iff φ' admits a model $(\mathfrak{U},\mathfrak{s},\mathfrak{h})$ such that $||\mathfrak{U}|| \leq 2n' \times (m'+3)$. We may thus check that φ' is satisfiable by fixing such a set $\mathfrak{U}$, guessing the value of $\mathfrak{s}(x)$ on each variable x free in φ, guessing some heap $\mathfrak{h}$ on $\mathfrak{U}$, and checking that $(\mathfrak{U},\mathfrak{s},\mathfrak{h}) \models C$ and that $(\mathfrak{U},\mathfrak{s},\mathfrak{h}) \models \varphi$. The former test is easy to perform by counting the number of allocated and nonallocated cells. For the latter test, we check the negation $(\mathfrak{U},\mathfrak{s},\mathfrak{h}) \not\models \varphi$, by testing that there exists a store $\mathfrak{s}'$ coinciding with $\mathfrak{s}$ on $x_1,\dots,x_n$ such that $(\mathfrak{U},\mathfrak{s}',\mathfrak{h}) \models \neg\phi$, i.e., such that $(\mathfrak{U},\mathfrak{s}',\mathfrak{h}) \models \bigvee_{M\in\mu^{fin}(\neg\phi)} M$. To this aim, we guess the value of each variable y_i in $\mathfrak{s}'$, guess a minterm M, check that $M \in \mu^{fin}(\neg\phi)$ (which can be done in polynomial space by Lemma 4) and check that $(\mathfrak{U},\mathfrak{s}',\mathfrak{h})$ validates every test formula in M (it is clear that this can be done in polynomial time). □

6 Conclusion

We have shown that the prenex fragment of Separation Logic over heaps with one selector, denoted as SL^1, is decidable in time not elementary recursive. Moreover, the Bernays-Schönfinkel-Ramsey $\mathsf{BSR}(\mathsf{SL}^1)$ is PSPACE-complete. These results settle an open question raised in [6] and allow one to draw a precise boundary between decidable and undecidable cases inside $\mathsf{BSR}(\mathsf{SL}^k)$. As far as applications are concerned, the logic $\mathsf{BSR}(\mathsf{SL}^1)$ can be used to reason on singly linked data-structures, where $*$ and $-\!\!*$ are used to state dynamic transformations of the heap and the quantifiers are useful to state general properties of the considered data-structure (e.g., to check that a loop invariant is preserved). Theorem 8, relating infinite and finite satisfiability, holds for any $k \geq 1$ and we believe that it could pave the way to further decidability results for prenex fragments of SL^k.

Acknowledgments The authors wish to thank Stéphane Demri, Etienne Lozes and Alessio Mansutti for the insightful discussions during the preparation of this paper.

References

1. Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.
2. Rémi Brochenin, Stéphane Demri, and Etienne Lozes. On the almighty wand. *Information and Computation*, 211:106 – 137, 2012.
3. C. Calcagno and D. Distefano. Infer: An Automatic Program Verifier for Memory Safety of C Programs. In *Proc. of NASA Formal Methods'11*, volume 6617 of *LNCS*. Springer, 2011.

4. Cristiano Calcagno, Hongseok Yang, and Peter W Ohearn. Computability and complexity results for a spatial assertion language for data structures. In *FST TCS 2001, Proceedings*, pages 108–119. Springer, 2001.
5. Stéphane Demri, Didier Galmiche, Dominique Larchey-Wendling, and Daniel Méry. Separation Logic with One Quantified Variable. In *CSR'14*, volume 8476 of *LNCS*, pages 125–138. Springer, 2014.
6. Mnacho Echenim, Radu Iosif, and Nicolas Peltier. The Bernays-Schönfinkel-Ramsey Class of Separation Logic on Arbitrary Domains. In *Foundations of Software Science and Computation Structures - 22nd International Conference, FOSSACS 2019, Held as part of ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings*, volume 11425 of *LNCS*, pages 242–259. Springer, 2019.
7. M. Fitting. *First-Order Logic and Automated Theorem Proving*. Texts and Monographs in Computer Science. Springer-Verlag, 1990.
8. Samin S Ishtiaq and Peter W O'Hearn. Bi as an assertion language for mutable data structures. In *ACM SIGPLAN Notices*, volume 36, pages 14–26, 2001.
9. Étienne Lozes. *Expressivité des logiques spatiales*. Thèse de doctorat, Laboratoire de l'Informatique du Parallélisme, ENS Lyon, France, November 2004. URL: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/PhD-lozes.ps.
10. Michael O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969. URL: http://www.jstor.org/stable/1995086.
11. J.C. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. In *Proc. of LICS'02*, 2002.