The Bernays-Schönfinkel-Ramsey Class of Separation Logic with Uninterpreted Predicates
Mnacho Echenim, Radu Iosif, Nicolas Peltier

To cite this version:
Mnacho Echenim, Radu Iosif, Nicolas Peltier. The Bernays-Schönfinkel-Ramsey Class of Separation Logic with Uninterpreted Predicates. ACM Transactions on Computational Logic, 2020, 21. hal-02388326

HAL Id: hal-02388326
https://hal.science/hal-02388326
Submitted on 1 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
The Bernays-Schönfinkel-Ramsey Class of Separation Logic with Uninterpreted Predicates

MNACHO ECHENIM, Univ. Grenoble Alpes, CNRS, LIG
RADU IOSIF, Univ. Grenoble Alpes, CNRS, Verimag
NICOLAS PELTIER, Univ. Grenoble Alpes, CNRS, LIG

This paper investigates the satisfiability problem for Separation Logic with \( k \) record fields, with unrestricted nesting of separating conjunctions and implications. It focuses on prenex formulæ with a quantifier prefix in the language \( \exists^* \forall^* \), that contain uninterpreted (heap-independent) predicate symbols. In analogy with first-order logic, we call this fragment Bernays-Schönfinkel-Ramsey Separation Logic \([\text{BSR}(\text{SL}^k)]\). In contrast with existing work on Separation Logic, in which the universe of possible locations is assumed to be infinite, we consider both finite and infinite universes in the present paper. We show that, unlike in first-order logic, the (in)finite satisfiability problem is undecidable for \([\text{BSR}(\text{SL}^k)]\). Then we define two non-trivial subsets thereof, for which the finite and infinite satisfiability problems are \( \text{PSPACE} \)-complete, respectively, assuming that the maximum arity of the uninterpreted predicate symbols does not depend on the input. These fragments are defined by controlling the polarity of the occurrences of separating implications, as well as the occurrences of universally quantified variables within their scope. These decidability results have natural applications in program verification, as they allow to automatically prove lemmas that occur in e.g. entailment checking between inductively defined predicates and validity checking of Hoare triples expressing partial correctness conditions.

Additional Key Words and Phrases: Separation logic, Bernays-Schönfinkel-Ramsey class, decision procedures, complexity, \( \text{PSPACE} \)-completeness

ACM Reference Format:

1 INTRODUCTION
Separation Logic \([14, 20]\) (SL) is a logical framework used in program verification to describe properties of the dynamically allocated memory, such as topologies of data structures (lists, trees), (un)reachability between pointers, etc. In a nutshell (formal definitions are given below), given an integer \( k \geq 1 \), the logic \( \text{SL}^k \) is obtained from the first-order theory of a finite partial function \( h : U \rightarrow U^k \) called a heap, by adding two non-classical connectives:
1. the \textit{separating conjunction} \( \phi_1 \ast \phi_2 \), that asserts the existence of a split of the heap into disjoint heaps satisfying \( \phi_1 \) and \( \phi_2 \) respectively, and
2. the \textit{separating implication}, or \textit{magic wand} \( \phi_1 \rightarrow \phi_2 \), stating that each extension of the heap by a disjoint heap satisfying \( \phi_1 \) must satisfy \( \phi_2 \).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
1529-3785/2019/12-Art $15.00
https://doi.org/
Intuitively, the set $U$ denotes the universe of possible of memory locations (cells) and $k$ is the number of record fields in each memory cell. The separating connectives $*$ and $\rightarrow$ may be used to express dynamic transformations of the heap. As such, they allow for concise definitions of program semantics, via weakest precondition calculi [14] and easy-to-write specifications of recursive linked data structures (e.g. singly- and doubly-linked lists, trees with linked leaves and parent pointers, etc.), when inductive definitions are added [20].

Investigating the decidability and complexity of the satisfiability problem for fragments of SL is thus of great theoretical and practical interest. In contrast to first-order logic for which the decision problem has been thoroughly investigated (see, e.g., [3]), only a few results are known for SL. The earliest such results show undecidability of $SL^k$ and the PSPACE-completeness of its quantifier-free fragment, for any $k \geq 2$ [7]. These results have been subsequently refined, by showing undecidability of $SL^1$, even if only two quantified variables are allowed [8]. Decidability of $SL^1$ is shown for the fragment without the magic wand connective, but the complexity lower bound is not elementary recursive. This lower bound drops if at most one quantified variable is allowed, in which case $SL^1$ is PSPACE-complete. Extending $SL^1$ with higher-order inductive predicates, such as reachability, leads to undecidability in the presence of the magic wand and becomes PSPACE-complete if the magic wand is not allowed [9].

A salient feature of SL is the ability of describing recursive data structures by means of inductive definitions. The axioms defining such interpreted predicates use a very restricted fragment of SL, consisting of atoms (equalities, disequalities and single cell descriptions) joined with separating conjunctions, called the symbolic heap fragment. Since negation does not occur within symbolic heaps, one must consider the satisfiability and entailment problems separately. For instance, satisfiability of a symbolic heap is EXPTIME-complete, in general, and NP-complete if the maximum arity of the predicates is a constant, not part of the input [5]. On the other hand, entailment between symbolic heaps is undecidable in general, and becomes elementary recursive under certain conditions guaranteeing that the treewidth of each model is bounded by the size of the inductive definition [13]. In particular, the problem is EXPTIME-hard [1] and the more restricted problem of the validity of entailments of the form $P(x_1, \ldots, x_n) \equiv Q(x_1, \ldots, x_n)$ has been recently shown to belong to 2EXPTIME [15].

In this paper, we consider prenex SL formulæ with a quantifier prefix in the language $\exists^* \forall^*$, possibly containing heap-independent uninterpreted\(^1\) predicate symbols. In analogy with the Bernays-Schönfinkel-Ramsey fragment of first-order logic with $\exists^* \forall^*$ quantifier prefix, equality and uninterpreted predicates and without function symbols of arity greater than 0 [BSR(FO)] [18], we call this fragment Bernays-Schönfinkel-Ramsey SL $[BSR(SL^k)]$.

As far as we are aware, all existing work on SL assumes that the universe $U$ is countably infinite. This assumption is not necessarily realistic in practice since the available memory is usually finite, although the bound depends on the hardware and is not known in advance. However, reasoning about pointer-manipulating programs under the finite memory assumption proves to be harder than under the assumption that memory is infinite, when the bound on the memory size is not known a priori. In particular, the frame rule of classical Separation Logic [?], which is a crucial enabler of local reasoning, breaks, in general, for programs that allocate memory, because, intuitively, adding frames is not possible unless enough free memory is available. Nevertheless, restricted versions of the frame rule still hold, with additional side conditions on the structure of the programs and/or the context to which it is applied. A thorough investigation of the soundness of the frame rule for bounded memory domains is, however, out of the scope of this paper and considered as future work.

In this paper we consider the satisfiability problem for $BSR(SL^k)$, with $k \geq 2$, in both cases of finite and infinite universe, referred to as finite and infinite satisfiability, respectively. We show that both problems are undecidable (unlike in the BSR fragment of first-order logic) and that they become PSPACE-complete under some additional restrictions, related to the occurrences of the magic wand and universal variables, namely:

\(^1\)By “uninterpreted” we mean that the interpretation of such predicate symbols is not fixed by a theory or by inductive definitions.

I. The infinite satisfiability problem is PSPACE-complete if the positive occurrences of $\neg$ (i.e., the occurrences of $\neg$ that are in the scope of an even number of negations) contain no universal variables.

II. The finite satisfiability problem is PSPACE-complete if there is no positive occurrence of $\neg$ (i.e., $\neg$ only occurs in the scope of an odd number of negations). This additional restriction stems from the fact that, actually, the finite satisfiability problem becomes undecidable even for only one positive occurrence of a $\neg$ with no variable within its scope.

These results establish sharp decidability frontiers within BSR($\mathcal{L}^k$). In both cases, we assume that the arity of the uninterpreted predicate symbols is bounded by a constant (the satisfiability problem is already NEXPTIME-complete for BSR first-order formulæ with unbounded predicate arity [17]). In contrast, the number $k$ of record fields is not bounded and may be part of the input. Reasoning on finite domains is more difficult than on infinite ones, due to the possibility of asserting cardinality constraints on unallocated cells, which explains that the latter condition is more restrictive than the former one. However, the finite universe hypothesis is especially useful when dealing with bounded memory issues, for instance checking that the execution of a program satisfies its postcondition, provided that there are sufficiently many available memory cells.

Theory-parameterized versions of BSR($\mathcal{L}^k$) have been shown to be undecidable in [19], e.g. when integer linear arithmetic is used to reason about locations, and claimed to be PSPACE-complete for countably infinite and finite unbounded location sorts, with no relation other than equality. In the present paper, we show that this claim is wrong, and draw a precise chart of decidability for both infinite and finite satisfiability of BSR($\mathcal{L}^k$), for $k \geq 2$. To complete the picture, the entire prenex fragment of $\mathcal{L}^1$ has been recently shown decidable but not elementary recursive, whereas the fragment BSR($\mathcal{L}^1$) is PSPACE-complete [10].

Undecidability is shown by reduction from BSR first-order formulæ with two monadic function symbols, for which satisfiability is known to be undecidable [3]. To establish the decidability results, we first show that every quantifier-free SL formula can be transformed into an equivalent boolean combination of formulæ of some specific patterns, called test formulæ. This result is interesting in itself, since it provides a precise and intuitive characterization of the expressive power of SL: it shows that separating connectives can be confined to a small set of test formulæ. Such expressive completeness results were already known for infinite universes (see, e.g., [16]), but our transformation algorithm also provides insights on the form of the obtained formulæ, especially on the polarity of occurrences of some test formulæ, which turns out to be useful latter on in the remainder of the paper. Further, we extend the expressive completeness result to finite universes, with additional test formulæ asserting cardinality constraints on unallocated cells.

One advantage of the translation to test formulæ is that the latter can be straightforwardly translated into first-order formulæ, by encoding the heap as a $(k+1)$-ary predicate. Note that another translation of quantifier-free $\mathcal{L}^k$ into first-order logic with equality has been described in [6]. There, the small model property of quantifier-free $\mathcal{L}^k$ [7] is used to bound the number of first-order variables to be considered and the separating connectives are interpreted as first-order quantifiers. The result is an equisatisfiable first-order formula. This translation scheme cannot be, however, directly applied to BSR($\mathcal{L}^k$), which does not have a small model property, being moreover undecidable.

We focus first on the infinite satisfiability problem and show that, if the above condition (I) is satisfied, then the obtained first-order formulæ are in the BSR(FO) class. The infinite satisfiability problem for BSR(FO) is thus reduced to the satisfiability problem for BSR(FO), with some additional constraints on the cardinality of the interpretation: the universe must be infinite, and the heap must be finite. We show that these constraints may be handled by relying on an existing characterization of the models of BSR(FO) formulæ with infinitely countable universe [12].

For the finite satisfiability class, the decidability proof is more involved, as the obtained first-order formulæ are not in BSR(FO), even if the above condition (II) is satisfied. However, this problem can be overcome by focusing
on some class of structures satisfying additional properties ensuring that a reduction to BSR(FO) is feasible. Note that in this case, the cardinality constraints on the universe and heap are straightforward to handle, as the BSR(FO) class is finitely controllable (i.e., every satisfiable BSR(FO) formula has a finite model).

The above transformation algorithm does not by itself provide an efficient decision procedure, as the size of the obtained boolean combination of test formulæ is exponential w.r.t. that of the initial (BSR) formula. The PSPACE upper bound thus relies on a careful analysis of the maximal size of the test formulæ. The analysis reveals that, although the boolean combination of test formulæ is of exponential size, its so-called minterms (i.e., the conjunctions in its disjunctive normal form) are of polynomial size and can be enumerated in polynomial space. The above algorithms can thus be refined to run in polynomial space.

This paper is an extended and thoroughly revised version of the conference paper [11]. The latter paper only handles SL formulæ with no uninterpreted predicate symbols. The addition of uninterpreted predicate symbols has a limited impact on the transformation of SL formulæ into boolean combinations of test formulæ. Indeed, since these predicates do not depend on the heap the corresponding atoms can be easily shifted outside of the separated connectives. However, non trivial adaptations are required in the satisfiability tests, since the presence of uninterpreted predicates makes it much more difficult to ensure that the considered formula has a model of the expected cardinality (finite or infinite).

Applications

Let us sketch two applications of our results to program verification. The first application is building proofs of validity for the entailments between inductively defined predicates in SL. The second application is proving the validity of Hoare triples with SL as base logic.

Checking Entailment between Inductively Defined Predicates. In contrast to other approaches [5?], our logic does not allow for inductively defined predicates (the predicates we consider are independent of the heap). Still, our results, embedded in inductive proof procedures, could prove useful to check entailment between formulæ containing such predicates. Consider for instance the following inductive definitions, describing a list segment with strictly increasing data fields and a possibly cyclic list segment, with no restrictions on the data, respectively:

\[ \tilde{ls}(x, y, d) \leftarrow \text{emp} \land x \approx y \lor \exists z \exists e . \ d \prec x \land (x \mapsto (d, z)) \land \tilde{ls}(z, y, e) \] sorted list segment from \( x \) to \( y \)

\[ \tilde{ls}(x, y) \leftarrow \text{emp} \land x \approx y \lor \exists z \exists f . \ x \mapsto (f, u) \lor \tilde{ls}(u, y) \] unrestricted list segment from \( x \) to \( y \)

Intuitively, a list segment is either empty, in which case the head and the tail coincide \([\text{emp} \land x \approx y]\), or it contains at least one element. We denote by \( x \mapsto (d, z) \) the fact that \( x \) is the only allocated memory location, which moreover points to a pair \((d, z)\), where \( d \) is a data field and \( z \) is a pointer field. When writing \( x \mapsto (d, z) \) and \( \tilde{ls}(z, y, e) \) we mean that \( x \mapsto (d, z) \) and \( \tilde{ls}(z, y, e) \) must hold over disjoint parts of the heap. The constraint \( d \prec e \), in the inductive definition of \( \tilde{ls} \), captures the fact that the list is strictly increasing, \( \prec \) being an uninterpreted predicate symbol that satisfies the transitivity and anti-symmetry axioms below:

\[ \forall a \forall b \forall c . \ a \prec b \land b \prec c \rightarrow a \prec c \quad \forall a \forall b . \ a \prec b \land b \prec a \rightarrow a \approx b \]

Now consider a fragment of the inductive proof showing that any sorted list segment is also a list segment:

\[ d \prec e \land x \mapsto (d, z) \land \tilde{ls}(z, y, e) \land \exists u \exists f . \ x \mapsto (f, u) \lor \tilde{ls}(u, y) \land \text{emp} \\land x \approx y \]

\[ \tilde{ls}(x, y, d) \lor \tilde{ls}(x, y) \]

\[ d \prec e \land x \mapsto (d, z) \models \exists u \exists f . \ x \mapsto (f, u) \]

by instantiation \( u \leftarrow z, f \leftarrow d \)
The bottom inference rule introduces one of the two cases produced by unfolding the inductive definitions on both sides of the sequent\(^2\). Note that the quantifications \(\exists z, e\) on the left-hand side have been omitted because they can be eliminated by using the standard \(\exists\)-left rule of the sequent calculus (if \(z\) and \(e\) are fresh variables).

The second inference rule is a reduction of the sequent obtained by unfolding, to a sequent matching the initial one (by renaming \(z\) to \(x\) and \(e\) to \(d\)), and allows to close this branch of the proof by an inductive argument, based on the principle of infinite descent [4]. The simplification applied by the second inference above relies on the validity of the entailment \(d < e \land x \rightarrow (d, z) \models \exists u \exists f. x \rightarrow (f, u)\), which reduces to the (un)satisfiability of the formula \(d < e \land x \rightarrow (d, z) \land \forall u \forall f. \neg x \rightarrow (f, u)\). The latter falls into the BSR(SL\(^3\)) fragment. A consequence of the results in this paper is that, if the inductive rules contain no occurrence of \(\leftarrow\) and \(\forall\), then there exist algorithms for solving the above entailment problem in both finite and infinite universes, in the presence of uninterpreted predicates. The only requirement is that the axiomatization of these predicates can be done using BSR(FO), i.e., that the interpretation of these predicates does not depend on the heap.

### Checking Inductive Invariants with Universal Quantifiers

Purely universal SL formule are also useful to express pre- or post-conditions asserting “local” constraints on the shape of the data structures manipulated by a program. For instance, the atomic proposition \(x \mapsto (p, n, d)\) states that the value of the heap at \(x\) is the triple \((p, n, d)\), where \(n\) (resp. \(p\)) is the location of the next (resp. previous) cell in the list and \(d\) is a data value. Moreover, \(x \mapsto (p, n, d)\) holds if and only if there is no location, other than \(x\), in the domain of the heap. With this in mind, the following formula describes a well-formed doubly-linked sorted list:

\[
\forall x_1, x_2, x_3, x_4, x_5, y_1, y_2 \cdot x_1 \mapsto (x_2, x_3, y_1) \mapsto x_2 \mapsto (x_4, x_5, y_2) \mapsto \top \rightarrow x_5 \approx x_1 \land y_1 < y_2
\]  

(1)

Such constraints cannot be expressed by using inductively defined predicates for which the entailment problem is known to be decidable\(^3\), which shows the practical relevance of the considered fragment. The separating implication (magic wand) seldom occurs in such shape constraints. However, it is useful to describe the dynamic transformations of the data structures, as in the following Hoare-style axiom, giving the weakest precondition of a universal formula \(\forall u \cdot \psi\) with respect to redirecting the \(i\)-th record field of \(x\) to \(z\) [14]:

\[
(x \mapsto (y_1, \ldots, y_k) \mapsto [x \mapsto (y_1, \ldots, y_i-1, z, \ldots, y_k) \mapsto \forall u \cdot \psi]) \cdot x.i := z \cdot \forall u \cdot \psi)
\]

For example, the Hoare-style axiom for the weakest precondition of the universal formula \(\forall u \cdot \psi\) when redirecting the ‘next’ field in a doubly-linked list is

\[
\{ x \mapsto (p, n, d) \mapsto \forall u \cdot \psi \} \cdot x.\text{next} := z \cdot \forall u \cdot \psi)\]

Intuitively, the formula \(x \mapsto (p, n, d) \mapsto [x \mapsto (p, z, d) \mapsto \forall u \cdot \psi]\) holds when the heap can be separated into disjoint parts, one in which cell \(x\) is allocated, and one that, when extended with a heap in which the ‘next’ field of \(x\) is mapped to \(z\), satisfies \(\forall u \cdot \psi\). The universal formula \(\forall u \cdot \psi\) could be the doubly-linked list invariant (1) for instance.

In the general case, the precondition for the redirection of the \(i\)-th record field of \(x\) to \(z\) is equivalent to

\[
\forall u \cdot x \mapsto (y_1, \ldots, y_k) \mapsto [x \mapsto (y_1, \ldots, y_i-1, z, \ldots, y_k) \mapsto \psi] \]  

because, although hoisting universal quantifiers outside of the separating conjunction is unsound in general, this is possible here due to the special form of the left-hand side \(x \mapsto (y_1, \ldots, y_i-1, z, \ldots, y_k)\) which unambiguously defines a single heap cell.

Checking entailment between two universal formulæ boils down to checking the satisfiability of a BSR(SL\(^3\)) formula, which can be done thanks to the decidability results in our paper. In particular, checking that \(\forall u \cdot \psi\)

is an invariant of the program statement \(x.i := z\) amounts to checking that the formula \(\forall u \cdot \psi \land \exists u \cdot \psi\)

The second case \(\text{emp} \land x = y \models \exists u \exists f. \cdot x \mapsto (f, u) \mapsto (x, u, y) \land \text{emp} \land x = y\) is trivial and omitted for clarity.

\(^3\)This is due to the fact that some of the edges, for instance those pointing to list values, may be “dangling”. In other words, this structure does not fulfill the so called establishment condition of [13].
The remainder of the paper is structured as follows. In Section 2, usual notions and results are briefly reviewed and the definition of the logic $SL^k$ is provided. In Section 4 a set of formula patterns, called test formulæ, is introduced, and it is shown that these patterns can be expressed in first-order logic. In Section 5, an algorithm is described to transform every $SL^k$ formula into an equivalent boolean combination of test formulæ. The output formula is of exponential size, however, we show that the conjunctions of literals occurring in its disjunctive normal form are of polynomial size and may be enumerated in polynomial space. In Section 6, the BSR($SL^k$) class is investigated and (un)decidability and complexity results are established based on the previous transformation algorithms. Section 7 briefly concludes the paper.

2 PRELIMINARIES

2.1 First Order Logic

Syntax. We denote by $\mathbb{Z}$ and $\mathbb{N}$ the sets of integer and natural numbers, respectively. Let $\mathbb{Z}_\infty = \mathbb{Z} \cup \{\infty\}$ and $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$, where for each $n \in \mathbb{Z}$ we have $n + \infty = \infty$ and $n < \infty$. For any countable set $S$, we denote by $|S|$ the cardinality of $S$.

Let $U$ be a sort symbol denoting a universe sort and let $B$ be the usual boolean sort. We consider a countably infinite set $\text{Var}$ of variables of sort $U$, ranged over by $x, y, z$, and a countably infinite set $\mathcal{F}$ of function symbols. Each function symbol $f \in \mathcal{F}$ has a sort $\sigma(f) \in \{U, B\}$. A function symbol $f$ takes $\#(f) \geq 0$ arguments of sort $U$. If $\#(f) = 0$ we call $f$ a constant and if $\#(f) = 1$ we say that $f$ is monadic. If $\sigma(f) = B$, $f$ is called a predicate. First-order (FO) terms $t$ and formulæ $\phi$ are defined by the following grammar:

$$
t ::= x \mid f(t_1, \ldots, t_{\#(f)}) \quad \phi ::= \bot \mid t_1 \approx t_2 \mid q(t_1, \ldots, t_{\#(q)}) \mid \phi \land \psi \mid \neg \phi \mid \exists x . \phi
$$

where $x \in \text{Var}$, $f, q \in \mathcal{F}$, $\sigma(f) = U$ and $\sigma(q) = B$. The logical symbols $\bot$ and $\top$ denote the boolean constants false and true, respectively. As usual, $f(t_1, \ldots, t_n)$ is simply written $f$ if $n = 0$. We write $\phi_1 \lor \phi_2$ for $\neg(\neg\phi_1 \land \neg\phi_2)$.

Semantics. First-order formulæ are interpreted over FO-structures $\mathcal{S} = (\mathcal{U}, s, I)$, where $\mathcal{U}$ is a nonempty countable set, called a universe, the elements of which are called locations; $s : \text{Var} \rightarrow \mathcal{U}$ is a partial mapping of variables to elements of $\mathcal{U}$, called a store and $I$ interprets each function symbol $f$ by a function $f^I : \mathcal{U}^{\#(f)} \rightarrow \mathcal{U}$ if $\sigma(f) = U$ or by a relation $f^I \subseteq \mathcal{U}^{\#(f)}$ if $\sigma(f) = B$. A structure $\mathcal{S}$ is finite when $|\mathcal{U}| \in \mathbb{N}$ and infinite otherwise.

By writing $\mathcal{S} \models \phi$, for a structure $\mathcal{S} = (\mathcal{U}, s, I)$, we mean that $\text{Var}(\phi) \subseteq \text{dom}(s)$ and $\phi$ is true when interpreted in $\mathcal{S}$. This relation is defined recursively on the structure of $\phi$, as usual. When $\mathcal{S} \models \phi$, we say that $\mathcal{S}$ is a model of $\phi$.

A formula is [finitely] satisfiable when it has a [finite] model. Given two formulæ $\phi_1$ and $\phi_2$, we say that $\phi_1$ entails $\phi_2$ (written $\phi_1 \models \phi_2$) when every model of $\phi_1$ is a model of $\phi_2$, and that $\phi_1$ and $\phi_2$ are equivalent (written $\phi_1 \equiv \phi_2$) when $(\mathcal{U}, s, I) \models \phi_1 \iff (\mathcal{U}, s, I) \models \phi_2$, for every structure $(\mathcal{U}, s, I)$. For any store $s$ on $\mathcal{U}$, variables $x_1, \ldots, x_n$ and

$$(y_1, \ldots, y_k) \to (x \mapsto (y_1, \ldots, y_{i-1}, z, \ldots, y_k) \to \psi))$$

is unsatisfiable. Because the magic wand occurs negated, this formula falls into a decidable class defined in the present paper, for both finite and infinite satisfiability.
elements \( \ell_1, \ldots, \ell_n \in \mathcal{U} \), we denote by \( s[x_1 \leftarrow \ell_1, \ldots, x_n \leftarrow \ell_n] \) the store that coincides with \( s \) on every variable not in \([x_1, \ldots, x_n]\) and maps \( x_i \) to \( \ell_i \), for all \( i = 1, \ldots, n \). We also call \( s[x_1 \leftarrow \ell_1, \ldots, x_n \leftarrow \ell_n] \) an extension of \( s \).

If \( y = (y_1, \ldots, y_n) \) is a vector of variables, and \( s \) is a store, then \( s(y) \) denotes the vector \((s(y_1), \ldots, s(y_n))\).

**BSR(FO) Formulae.** The Bernays-Schönfinkel-Ramsey fragment of FO [BSR(FO)] is the set of formulæ of the form \( \exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \cdot \phi \), where \( \phi \) is a quantifier-free formula and all function symbols \( f \in \mathcal{F}(\phi) \) of arity \( \#(f) > 0 \) have sort \( \sigma(f) = B \). For simplicity we often restrict ourselves to BSR(FO) formulæ containing no existential quantification. This is without any loss of generality, since \( \exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \cdot \phi \) is satisfiable if and only if \( \forall y_1 \ldots \forall y_m \cdot \phi \) is satisfiable.

**Definition 2.1.** Consider the structures \( S = (\mathcal{U}, s, I) \) and \( S' = (\mathcal{U}', s', I') \). The structure \( S' \) is called a restriction of \( S \) to \( \mathcal{U}' \) if \( \mathcal{U}' \subseteq \mathcal{U} \), \( s'(x) = s(x) \) for every \( x \in \text{dom}(s) \), \( q^{I'} = q^I \cap \mathcal{U}'^q \) for every predicate symbol \( q \) and \( f^{I'} = f^I \) for every function symbol \( f \).

The following proposition states a well-known property of BSR(FO):

**Proposition 2.2.** Let \( \phi \) be a formula in BSR(FO) with no existential quantifier and let \( S = (\mathcal{U}, s, I) \) be a model of \( \phi \). If \( \mathcal{U}' \) is a nonempty subset of \( \mathcal{U} \) such that \( \{s(x) : x \in \text{Var}(\phi)\} \cup \{c^I : c \in \text{Const}(\phi)\} \subseteq \mathcal{U}' \) and \( S' = (\mathcal{U}', s', I') \) is a restriction of \( S \) to \( \mathcal{U}' \), then \( S' \) is a model of \( \phi \). As a consequence, if \( \phi \) is satisfiable, then it admits a model \((\mathcal{U}, s, I)\) such that \(|\mathcal{U}| \leq \max\{1, ||\text{Var}(\phi)|| + ||\text{Const}(\phi)||\} \).

**Proof.** See for instance \cite[Theorem 3]{12}.

The decidability of BSR(FO) is a consequence of the above small model property. It is known that the satisfiability problem for this class is \( \text{NEXPTIME} \)-complete \cite{?}. The condition requiring the absence of function symbols of sort \( U \) in BSR(FO) is justified by the fact that undecidability occurs as soon as two monadic function symbols are allowed. Let BSR\(^2\)(FO) be the extension of BSR(FO) consisting of the formulæ \( \exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \cdot \phi \), where \( \phi \) is a quantifier-free formulæ in which at most two monadic function symbols occur.

**Proposition 2.3.** The satisfiability problem is undecidable for BSR\(^2\)(FO), even if only one universal quantifier and no predicates are allowed.

**Proof.** See \cite[Theorem 4.1.8]{3}.

### 2.2 Separation Logic

**Syntax** Let \( k \) be a strictly positive integer. The logic SL\(^k\) is the set of formulæ generated by the grammar:

\[
\phi \; ::= 
\begin{array}{c}
\bot \mid \top \mid \text{emp} \mid x \equiv y \mid x \mapsto (y_1, \ldots, y_k) \mid q(x_1, \ldots, x_{s(q)}) \mid \phi \land \psi \mid \neg \phi \mid \phi \ast \psi \mid \phi \dashv \psi \mid \exists x \cdot \phi
\end{array}
\]

where \( x, y, y_1, \ldots, y_k, x_1, \ldots, x_{s(q)} \in \text{Var}, q \in \mathcal{F} \) and \( \sigma(q) = B \). The connectives \( \ast \) and \( \dashv \) are respectively called the *separating conjunction* and *separating implication* (or *magic wand*). The *size* \( |\phi| \) and set of free variables \( \text{Var}(\phi) \) of an SL\(^k\) formula \( \phi \) are defined as for first-order formulæ, as well as the formulæ \( \phi_1 \lor \phi_2, \phi_1 \rightarrow \phi_2, \phi_1 \leftrightarrow \phi_2 \) and \( \forall x \cdot \phi \). Moreover, we write \( \phi_1 \dashv \phi_2 \) for \( \neg (\phi_1 \rightarrow \neg \phi_2) \) and call the symbol \( \dashv \) *separation* in the following. Throughout the paper, we assume that the arity of the predicate symbols occurring in the SL\(^k\) formulæ is bounded by a constant, whereas \( k \) is not necessarily bounded.

**Definition 2.4.** Given a SL\(^k\) formula \( \phi \) and a subformula \( \psi \) of \( \phi \), we say that \( \psi \) *occurs at polarity* \( p \in \{-1, 0, 1\} \) if one of the following holds:

1. \( \phi = \psi \) and \( p = 1 \),
2. \( \phi = \neg \phi_1 \) and \( \psi \) occurs at polarity \( -p \) in \( \phi_1 \),
3. \( \phi = \phi_1 \land \phi_2 \) or \( \phi = \phi_1 \ast \phi_2 \), and \( \psi \) occurs at polarity \( p \) in \( \phi_i \), for some \( i = 1, 2 \),
4. \( \phi = \phi_1 \rightarrow \phi_2 \) and either \( \psi \) is a subformula of \( \phi_1 \) and \( p = 0 \), or \( \psi \) occurs at polarity \( p \) in \( \phi_2 \), or
(5) $\phi = \exists x . \phi_1$ and $\psi$ occurs at polarity $p$ in $\phi_1$.

A polarity of $1, 0$ or $-1$ is also referred to as positive, neutral or negative, respectively.

Note that our notion of polarity is slightly different than the usual one, because the antecedent of a separating implication is of neutral polarity while the antecedent of an implication is usually of negative polarity. This is meant to strengthen upcoming decidability results (see Remark 3.4).

**Semantics** $\mathsf{SL^k}$ formulae are interpreted over $\mathsf{SL}$-structures $S = (\mathcal{U}, s, I, h)$, where $\mathcal{U}$, $s$, $I$, and $h$ are defined as for FO and $h : \mathcal{U} \rightarrow_{fo} \mathcal{U}^k$ is a finite partial mapping of $\mathcal{U}$ to $k$-tuples of elements of $\mathcal{U}$, called a heap. As for FO, a $\mathsf{SL}$-structure $S$ is **finite** when $|\mathcal{U}| \in \mathbb{N}$ and **infinite** otherwise. We denote by $\text{dom}(h)$ the domain of the heap $h$ and by $||h|| \in \mathbb{N}$ the cardinality of $\text{dom}(h)$. A location $\ell \in \mathcal{U}$ (resp. a variable $x$) is **allocated** in $S$ if $\ell \in \text{dom}(h)$ (resp. if $s(x) \in \text{dom}(h)$). Two heaps $h_1$ and $h_2$ are disjoint iff $\text{dom}(h_1) \cap \text{dom}(h_2) = \emptyset$, in which case $h_1 \uplus h_2$ denotes their union. $h'$ is an extension of $h$ iff $h' = h \uplus h''$, for some heap $h''$. The relation $(\mathcal{U}, s, I, h) \models \varphi$ is defined recursively on the structure of $\varphi$, as follows:

$$(\mathcal{U}, s, I, h) \models \top \iff \text{always}$$

$$(\mathcal{U}, s, I, h) \models \bot \iff \text{never}$$

$$(\mathcal{U}, s, I, h) \models x = y \iff s(x) = s(y)$$

$$(\mathcal{U}, s, I, h) \models \varphi_1 \land \varphi_2 \iff (\mathcal{U}, s, I, h) \models \varphi_1, \text{for all } i = 1, 2$$

$$(\mathcal{U}, s, I, h) \models \varphi \iff (\mathcal{U}, s, I, h) \models \psi$$

$$(\mathcal{U}, s, I, h) \models \exists x . \varphi_1 \iff \text{there exists } u \in \mathcal{U} \text{ such that } (\mathcal{U}, s[x \leftarrow u], I, h) \models \varphi_1$$

$$(\mathcal{U}, s, I, h) \models \varphi_1 \lor \varphi_2 \iff \text{there exist disjoint heaps } h_1, h_2 \text{ such that } h = h_1 \uplus h_2$$

$$(\mathcal{U}, s, I, h) \models \varphi_1 \rightarrow \varphi_2 \iff \text{for all heaps } h' \text{ disjoint from } h \text{ such that } (\mathcal{U}, s, I, h') \models \varphi_1,$$

we have $(\mathcal{U}, s, I, h \uplus h') \models \varphi_2$.

Satisfiability, entailment and equivalence are defined for $\mathsf{SL^k}$ as for FO formulae. We write $\phi \equiv_{\text{fin}} \psi$ (resp. $\phi \equiv_{\text{inf}} \psi$) if $\phi$ has the same truth value as $\psi$ in all finite (resp. infinite) structures.

**Remark 2.5.** The cardinality of the universe has a deep impact on the semantics of $\mathsf{SL}$ formulae. For instance, the formula $\phi = \neg \text{emp} \rightarrow \bot$ states that no nonempty heap disjoint from the current heap exists, which is always false in an infinite universe (since every heap is finite) but is true in a finite universe where all elements are allocated.

3 THE BSR($\mathsf{SL^k}$) CLASS

In this section, we give the definition of the Bernays-Schönfinkel-Ramsey fragment of $\mathsf{SL^k}$ and provide a brief summary of the results proved in this paper.

**Definition 3.1.** The Bernays-Schönfinkel-Ramsey fragment of $\mathsf{SL^k}$, denoted by $\mathsf{BSR}(\mathsf{SL^k})$, is the set of formulæ of the form $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m . \varphi$, where $\varphi$ is a quantifier-free $\mathsf{SL^k}$ formula.

Note that, since there is no function symbol of sort $U$ in $\mathsf{SL^k}$, there is no restriction, other than the form of the quantifier prefix, defining $\mathsf{BSR}(\mathsf{SL^k})$. As for FO, we will often restrict ourselves to $\mathsf{BSR}(\mathsf{SL^k})$ formulae containing no existencial quantifier. As satisfiability is not decidable for $\mathsf{BSR}(\mathsf{SL^k})$ (see Theorem 3.3 below), we define two fragments of $\mathsf{BSR}(\mathsf{SL^k})$ for which finite and infinite satisfiability are respectively decidable. The definition is based on the polarity (see Definition 2.4) of the occurrences of the symbol $\rightarrow$ and on the universal variables occurring within their scope.
Definition 3.2. Given an integer \( k \geq 1 \), we define:

1. \( \text{BSR}^{\text{inf}}(\mathcal{L}^k) \) as the set of formulae \( \forall y_1 \ldots \forall y_m \cdot \phi \) such that for all \( i \in [1, m] \) and all formulae \( \psi_1 \to \psi_2 \)
   occurring at polarity 1 in \( \phi \), we have \( y_i \not\in \text{Var}(\psi_1) \cup \text{Var}(\psi_2) \).
2. \( \text{BSR}^{\text{fin}}(\mathcal{L}^k) \) as the set of formulae \( \forall y_1 \ldots \forall y_m \cdot \phi \) such that no formula \( \psi_1 \to \psi_2 \) occurs at polarity 1 in \( \phi \).

Note that \( \text{BSR}^{\text{fin}}(\mathcal{L}^k) \subseteq \text{BSR}^{\text{inf}}(\mathcal{L}^k) \subseteq \text{BSR}(\mathcal{L}^k) \), for any \( k \geq 1 \). We know state the main results of the paper.

**Theorem 3.3.** The satisfiability problem is undecidable for \( \text{BSR}(\mathcal{L}^k) \). The infinite satisfiability problem for \( \text{BSR}^{\text{inf}}(\mathcal{L}^k) \) and the finite satisfiability problem for \( \text{BSR}^{\text{fin}}(\mathcal{L}^k) \) are both \( \text{PSPACE}\)-complete.

The remainder of the paper is devoted the proof of Theorem 3.3 (see Theorems 6.1, 6.11 and 6.20).

**Remark 3.4.** Because the polarity of the antecedent of a separating implication is neutral, Definition 3.2 imposes no constraint on the occurrences of separating implications at the left of an occurrence of \( \to \).

## 4 TEST FORMULÆ FOR \( \mathcal{L}^k \)

### 4.1 Definition and Basic Properties

We define a small set of \( \mathcal{L}^k \) patterns of formulae, possibly parameterized by a positive integer, called test formulæ. These patterns capture properties related to allocation, points-to relations in the heap and cardinality constraints.

**Definition 4.1.** The following patterns are called test formulæ:

\[
\begin{align*}
& x \to y \quad \text{def} \quad x \mapsto (y_1, \ldots, y_k) \cdot \top \\
& \text{alloc}(x) \quad \text{def} \quad x \mapsto (x, \ldots, x) \cdot \bot \\
& x \approx y \\
& q(x_1, \ldots, x_{\#(q)}) \quad |h| \geq m \quad \text{def} \quad \begin{cases} |h| \geq m - 1 \cdot \text{emp}, & \text{if } 0 < m < \infty \\ \top, & \text{if } m = 0 \\ \bot, & \text{if } m = \infty \end{cases}
\end{align*}
\]

where \( x, y \in \text{Var}, q \in \mathcal{T}, \sigma(q) = B, x_1, \ldots, x_{\#(q)} \in \text{Var}, n \in \mathbb{N} \) and \( m \in \mathbb{N}_\infty \).

If \( \phi \) is a test formula of the form \( t \geq s \) then the formula \( \neg \phi \) will often be denoted by \( t < s \). For a set of variables \( X \subseteq \text{Var} \), let \( \text{alloc}(X) \equiv \bigwedge_{x \in X} \text{alloc}(x) \) and \( \text{nalloc}(X) \equiv \bigwedge_{x \in X} \neg \text{alloc}(x) \). The trivial test formulæ \( |h| \geq 0 \) and \( |h| \geq \infty \) are introduced for reasons that will become clear in Section 5. The semantics of test formulæ is very natural: \( x \to y \) means that \( x \) points to vector \( y \), \( \text{alloc}(x) \) means that \( x \) is allocated, and the arithmetic expressions are interpreted as usual, where \( |h| \) and \( |U| \) respectively denote the number of allocated cells and the number of locations (possibly infinite). Formally:

**Proposition 4.2.** Given an \( \mathcal{L} \)-structure \( (\mathbb{N}, \sigma, I, h) \), the following equivalences hold, for all variables \( x, y_1, \ldots, y_k \in \text{Var} \) and integers \( n \in \mathbb{N} \):

\[
\begin{align*}
& (\mathbb{N}, \sigma, I, h) \models x \to y \quad \iff \quad h(s(x)) = s(y) \\
& (\mathbb{N}, \sigma, I, h) \models |h| \geq |U| - n \quad \iff \quad |h| \geq |\mathbb{N}| - n \\
& (\mathbb{N}, \sigma, I, h) \models \text{alloc}(x) \quad \iff \quad s(x) \in \text{dom}(h)
\end{align*}
\]

**Proof.** Let \( S = (\mathbb{N}, \sigma, I, h) \) be an \( \mathcal{L} \)-structure. We establish each statement separately.

- **S \models x \to y \iff b(s(x)) = (s(y_1), \ldots, s(y_k))**. Assume that \( S \models x \to y \). Then by definition, there exist disjoint heaps \( b_1, b_2 \) such that \( (\mathbb{N}, \sigma, I, b_1) \models x \to y \), \( (\mathbb{N}, \sigma, I, b_2) \models \top \) and \( h = b_1 \uplus b_2 \). Thus \( s(x) \in \text{dom}(b_1) \subseteq \text{dom}(h) \) and \( b(s(x)) = b_1(s(x)) = (s(y_1), \ldots, s(y_k)) \). Conversely, assume \( b(s(x)) = (s(y_1), \ldots, s(y_k)) \). Then \( b \) is of the form \( b_1 \uplus b_2 \), where \( b_1 \) is the restriction of \( b \) to \( \{s(x)\} \) and \( b_2 \) is the restriction of \( b \) to \( \mathbb{N} \setminus \{s(x)\} \). By definition, \( b_1 = (s(x), (s(y_1), \ldots, s(y_k))) \), hence \( b_1 \models x \to y \). Furthermore, \( b_2 \models \top \). Thus \( S \models x \to y \).
• \(\mathcal{S} \models \text{alloc}(x) \iff s(x) \in \text{dom}(h)\). Assume that \(\mathcal{S} \models \text{alloc}(x)\). This means that there is no heap \(b'\) disjoint from \(h\) such that \((\mathcal{S}, s, I, b') \models x \leftrightarrow (\ldots, x)\). If \(s(x) \notin \text{dom}(h)\), then the heap \(b'\) defined as \(b' = (s(x), (s(x), \ldots, s(x)))\) is disjoint from \(h\) and we have \((\mathcal{S}, s, I, b') \models x \leftrightarrow (\ldots, x)\). Thus \(s(x) \in \text{dom}(h)\). Conversely, assume \(s(x) \in \text{dom}(h)\). By definition, for any heap \(b'\) such that \((\mathcal{S}, s, I, b') \models x \leftrightarrow (\ldots, x)\) we have \(s(x) \in \text{dom}(b')\), hence \(b' \cap h = \emptyset\). Thus \(\mathcal{S} \models \text{alloc}(x)\).

• \(\mathcal{S} \models |h| \geq n \iff ||b|| \geq n\). Assume that \(\mathcal{S} \models |h| \geq n\). Then since \(h\) has a finite domain, it is clear that \(||b|| \geq n\) if \(n = 0\) and that no such structure exists if \(n = \infty\). When \(n \geq 1\), we prove the result by induction on \(n\). By definition, \(\mathcal{S} \models |h| \geq n - 1 \iff \neg \text{emp}\), hence there exist disjoint heaps \(b_1, b_2\) such that \((\mathcal{S}, s, I, b_1) \models |h| \geq n - 1\), \((\mathcal{S}, s, I, b_2) \models \neg \text{emp}\) and \(b_1 \cap b_2 = \emptyset\). By the induction hypothesis \(||b_1|| \geq n - 1\) and by definition, \(||b_2|| \geq 1\), so that \(||b_1 \cup b_2|| \geq n\). Conversely, assume that \(||b|| \geq n\). Since \(h\) is finite, this entails that \(n \neq \infty\). If \(n = 0\) then \(\mathcal{S} \models |h| \geq n\) always holds. Otherwise, we prove the result by induction on \(n\). Consider \(\ell \in \text{dom}(h)\) and let \(b_1\) and \(b_2\) respectively denote the restrictions of \(b\) to \(U \setminus \{\ell\}\) and to \{\ell\}, so that \(h = b_1 \cup b_2\). Since \(||b_1|| \geq n - 1\), by the induction hypothesis \((\mathcal{S}, s, I, b_1) \models |h| \geq n - 1\), and since \(\text{dom}(b_2) \neq \emptyset\), \((\mathcal{S}, s, I, b_2) \models \neg \text{emp}\). Thus \(\mathcal{S} \models |h| \geq n\).

• \(\mathcal{S} \models |U| \geq n \iff ||I|| \geq n\). Assume that \(\mathcal{S} \models |U| \geq n\). Then there exists a heap \(b_1\) disjoint from \(h\) such that \((\mathcal{S}, s, I, b \cup b_1) \models |h| \geq n\). This entails that \(|b \cup b_1| \geq n\) and since \(\text{dom}(b \cup b_1) \subseteq U\), necessarily, \(||I|| \geq n\). Conversely, if \(||I|| \geq n\), then there exists a set \(L \subseteq U\) such that \(\text{dom}(b) \cap L = \emptyset\) and \(||I|| = n - ||b||\). Let \(b'\) be any heap of domain \(L\). Then \(b\) and \(b'\) are disjoint and \((\mathcal{S}, s, I, b \cup b') \models |h| \geq n\), which proves that \(\mathcal{S} \models |U| \geq n\).

• \(\mathcal{S} \models |h| \geq |U| - n \iff ||b|| \geq ||I|| - n\). Assume that \(\mathcal{S} \models |h| \geq |U| - n\). By definition, this entails that there is no heap disjoint from \(h\) with a domain of cardinality at least \(n + 1\). In particular, if \(L = U \setminus \text{dom}(h)\), and \(b'\) is any heap of domain \(L\), then \(\text{dom}(b) \cap \text{dom}(b') = \emptyset\), hence \(||b'|| \leq n\). Since \(||I|| = ||b|| + ||b'||\), we deduce that \(||b|| \geq ||I|| - n\). Conversely, if \(||b|| \geq ||I|| - n\) then \(||U \setminus \text{dom}(b)|| \leq n\), hence there is no heap disjoint from \(h\) with a domain of cardinality at least \(n + 1\), so that \(\mathcal{S} \models |h| \geq |U| - n\).

\[\square\]

Not all atoms of \(\text{SL}^k\) are test formul\(\text{ae}, for instance \(x \mapsto y\) and \(\text{emp}\) are not test formul\(\text{ae}. However, by Proposition 4.2, we have the equivalences \(x \mapsto y \equiv x \mapsto y \wedge \neg |h| \geq 2\) and \(\text{emp} \equiv \neg |h| \geq 1\). Note that, for any \(n \in \mathbb{N}\), the test formul\(\text{ae} \geq n\) and \(|h| \geq |U| - n\) are trivially true and false respectively, if the universe is infinite.

### 4.2 A Generalization of Test Formul\(\text{ae}

For technical convenience, we extend the previous patterns to express more general cardinality constraints. For every \(n \in \mathbb{N}\), we denote by \(|U| \approx n\) (resp., \(|h| \approx n\)) the formul\(\text{ae} \geq n \wedge |U| < n + 1\) (resp., \(|h| \geq n \wedge |h| < n + 1\)). Similarly, \(|h| \approx |U| - n\) denotes either \(|h| \geq |U| - n \wedge |h| < |U| - (n + 1)\) (if \(n > 0\)) or \(|h| \geq |U| - 0\) (if \(n = 0\)). We then extend the notation \(|h| \approx t\) to the case where \(t\) is a linear function of \(|U|\), with coefficients in \(\mathbb{Z}\).

\[\text{Definition 4.3.} \text{Given integers } \alpha, \beta \in \mathbb{Z}, \text{where } \alpha \notin \{0, 1\}, \text{let}
\]

\[|h| \geq \alpha \cdot |U| + \beta \\quad \text{def} \quad \begin{cases} 
\bot & \text{if } \alpha > 1, \beta > 0 \\
\top & \text{if } \alpha, \beta < 0 \\
|U| < \left(\frac{1 - \beta}{\alpha}\right)^\beta \wedge \left(\frac{1}{\alpha + \beta}\right)^\beta (|U| \approx n \rightarrow |h| \geq \alpha \cdot n + \beta) & \text{if } \alpha > 1, \beta \leq 0 \\
\left(\frac{1}{\alpha + \beta}\right)^\beta (|U| \approx n \rightarrow |h| \geq \alpha \cdot n + \beta) & \text{if } \alpha < 0, \beta \geq 0 
\end{cases}
\]

If \(\alpha = 0\) and \(\beta < 0\) then \(|h| \geq |U| + \beta \\quad \text{def} \quad \top\). If \(\alpha = 1\) and \(\beta > 0\) then \(|h| \geq \alpha \cdot |U| + \beta \\quad \text{def} \quad \bot\).

Note that the cases \(\alpha = 0, \beta \geq 0\) and \(\alpha = 1, \beta \leq 0\) are already covered by Definition 4.1. The following proposition states that the semantics of these formul\(\text{ae}\) as is expected.
Proposition 4.4. Given an SL-structure \((\mathcal{U}, s, I, h)\), we have \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot |U| + \beta \iff |b| \geq \alpha \cdot |U| + \beta\), for all \(\alpha, \beta \in \mathbb{Z}, \alpha \notin \{0, 1\}\).

Proof. We distinguish the four cases below:

\begin{itemize}
  \item If \(\alpha > 1\) and \(\beta > 0\) then \(||U|| \geq |b|| \geq \alpha \cdot |U|| + \beta\) never holds.
  \item If \(\alpha < 0\) and \(\beta < 0\) then \(|b| \geq 0 \geq \alpha \cdot |U|| + \beta\), always holds.
  \item If \(\alpha > 1\) and \(\beta \leq 0\), assume first that \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot |U| + \beta\). Then \((\mathcal{U}, s, I, h) \models |U| < \left\lceil \frac{-\beta}{\alpha - 1} \right\rceil\), by Proposition 4.2. If \(||U|| > \left\lceil \frac{-\beta}{\alpha - 1} \right\rceil\) then \(||U|| \geq \left\lceil \frac{-\beta}{\alpha - 1} \right\rceil + 1 = \left\lceil \frac{1 - \beta}{\alpha - 1} \right\rceil\), which contradicts \((\mathcal{U}, s, I, h) \models |U| < \left\lceil \frac{-\beta}{\alpha - 1} \right\rceil\), by Proposition 4.2. Otherwise, we have \(||U|| = n\), with \(1 \leq n \leq \left\lceil \frac{-\beta}{\alpha - 1} \right\rceil\). In this case \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot n + \beta\), which implies \(|b|| \geq \alpha \cdot |U|| + \beta\), by Proposition 4.2. Conversely, assume that \(||U|| \geq \alpha \cdot |U|| + \beta\). Since necessarily \(||U|| \geq |b||\), we obtain \(||U|| \geq \alpha \cdot |U|| + \beta\), i.e., \(||U|| > \alpha \cdot |U|| + \beta - 1\) and thus \(||U|| < \left\lceil \frac{-\beta}{\alpha - 1} \right\rceil\), so that \((\mathcal{U}, s, I, h) \models |U| < \left\lceil \frac{1 - \beta}{\alpha - 1} \right\rceil\). Moreover, if \(n = ||U||\) then \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot n + \beta\) by Proposition 4.2.
  \item If \(\alpha < 0\) and \(\beta \geq 0\), assume first that \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot |U| + \beta\). If, moreover, \(||U|| \geq \frac{-\beta}{\alpha}\), then \(|\alpha \cdot |U|| + \beta \leq 0\), thus \(|b|| \geq 0 \geq \alpha \cdot |U|| + \beta\) holds. Otherwise, \(1 \leq \alpha \cdot |U|| + \beta\), and if \((\mathcal{U}, s, I, h) \models |U| = n\), for some \(1 \leq n < \left\lceil \frac{-\beta}{\alpha} \right\rceil\), then we have \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot n + \beta\), thus \(|b|| \geq \alpha \cdot |U|| + \beta\), by Proposition 4.2. Conversely, assume that \(|b|| \geq \alpha \cdot |U|| + \beta\) and \((\mathcal{U}, s, I, h) \models |U| = n\), for some integer \(1 \leq n < \left\lceil \frac{-\beta}{\alpha} \right\rceil\).
\end{itemize}

By Proposition 4.2, we have \(||U|| = n\) and \(|b|| \geq \alpha \cdot n + \beta\), thus \((\mathcal{U}, s, I, h) \models |h| \geq \alpha \cdot |U| + \beta\).

\[\square\]

4.3 From Test formulæ to FO

The introduction of test formulæ (Definition 4.1) is motivated by the reduction of the (in)finite satisfiability problem for quantified boolean combinations thereof to the same problem for FO. The reduction is based on a straightforward encoding of the heap as a \((k + 1)\)-ary predicate symbol, however it is devised below in such a way that the obtained formula is in the BSR class, if possible. To this purpose, we also use a monadic predicate symbol encoding the domain of the heap and boolean constants encoding cardinality constraints. We thus introduce several special (pairwise distinct) function symbols: a \((k + 1)\)-ary predicate \(p\), a monadic predicate \(b\), boolean constants \(a_n, b_n, a_{n+1}\) and \(c_n\), and the following constants of sort \(U: u_n, u_n^i, v_n, w_n\) and \(\xi_n^i\), for \(n \geq 0\), \(i \in [1, k]\) and \(x \in \text{Var}\). The symbol \(p\) will encode the heap, \(b\) will encode the domain of the heap, the constants \(a_n, b_n\) and \(c_n\) encode the constraints over the number of (allocated or unallocated) locations, and \(u_n, u_n^i, v_n, w_n\) and \(\xi_n^i\) are interpreted as pairwise distinct elements of the universe, used to express such constraints in FO.

Given a quantified boolean combination of test formulæ \(\phi\) not containing the above symbols, the FO formula \(\tau(\phi)\) is defined by induction on the structure of \(\phi\):

\begin{align*}
  \tau(|h| \geq n) & \overset{\text{def}}{=} a_n \\
  \tau(|U| \geq n) & \overset{\text{def}}{=} b_n \\
  \tau(x \leftarrow y) & \overset{\text{def}}{=} p(x, y, \ldots, y_k) \\
  \tau(\text{alloc}(x)) & \overset{\text{def}}{=} d(x) \\
  \tau(\phi_1 \wedge \phi_2) & \overset{\text{def}}{=} \tau(\phi_1) \wedge \tau(\phi_2) \\
  \tau(\exists x . \phi_1) & \overset{\text{def}}{=} \exists x . \tau(\phi_1) \\
  \tau(q(x_1, \ldots, x_{n(q)})) & \overset{\text{def}}{=} q(x_1, \ldots, x_{n(q)}) \\
  \tau(x \equiv y) & \overset{\text{def}}{=} x \equiv y
\end{align*}

The special symbols are related by the following axioms:

\begin{align*}
  (\text{Heap}) & \quad \forall x \forall y \forall y'. p(x, y) \wedge p(x, y') \rightarrow y \equiv y' \\
  (\text{Dom}_x) & \quad d(x) \rightarrow p(x, \xi_x^1, \ldots, \xi_x^k) \quad \text{for each } x \in \text{Var}(\phi)
\end{align*}
\[ (A_0) \quad a_0 \quad (A_n) \quad \{ a_n \rightarrow (a_{n-1} \land p(u_n, u_{n-1}, \ldots, u_0) \land \land_{i=1}^{n-1} \neg u_i \equiv u_n) \} \]

\[ \land \land_{i=1}^{n-1} x \equiv u_i \]

\[ (B_0) \quad b_0 \quad (B_n) \quad \{ b_n \rightarrow (b_{n-1} \land \land_{i=1}^{n-1} \neg v_i \equiv v_n) \} \]

\[ \land \land_{i=1}^{n-1} x \equiv v_i \]

\[ (C_0) \quad c_0 \quad (C_n) \quad \forall y \cdot c_n \rightarrow (c_{n-1} \land \neg p(w_n, y) \land \land_{i=1}^{n-1} \neg w_n \equiv w_i) \]

Intuitively, \( p \) encodes the heap in the following sense. If \( (\mathcal{U}, s, I) \models Heap \) then there exists a heap \( \mathcal{H} \) on \( \mathcal{U} \) such that \( y = b(x) \Leftrightarrow (x, y) \in p^f \). The constant \( a_n \) (resp. \( b_n \)) is true if there are at least \( n \) cells in the domain of the heap (resp. in the universe), namely \( u_1, \ldots, u_n \) (resp. \( v_1, \ldots, v_n \)). If \( c_n \) is true, then there are at least \( n \) locations \( w_1, \ldots, w_n \) outside of the domain of the heap (i.e., \( n \) unallocated locations), but the converse does not hold. Indeed, the axioms \( C_n \) do not state the equivalence of \( c_n \) with the existence of at least \( n \) free locations, because such an equivalence cannot be expressed in BSR(FO). Similarly, the axiom \( Dom \) states that if \( x \) is allocated then \( b(x) \) holds, but the converse is true only for \( x \in \text{Var}(\phi) \) (as stated by the axiom \( Dom_x \)). Again, adding the implication \( \forall x \cdot b(x) \rightarrow \exists y_1, \ldots, y_k \cdot p(x, y_1, \ldots, y_k) \) would result in a formula that is not in BSR(FO). Instead, we only assert finitely many (skolemized) instances of the latter formula, for every free variable \( x \), which is sufficient for our purpose. As a consequence, the transformation preserves sat-equivalence only if the formulae \( |h| \geq |U| - n \) or \( \text{alloc}(x) \) with \( x \notin \text{Var}(\phi) \) occur only at negative polarity (see Lemma 4.9, Point 2).

**Definition 4.5.** Given a structure \( (\mathcal{U}, s, I) \) such that \( (\mathcal{U}, s, I) \models Heap \) and a heap \( \mathcal{H} \) on \( \mathcal{U} \), if \( x = b(y) \Leftrightarrow (x, y) \in p^f \), then we say that \( \mathcal{H} \) is associated with \( (\mathcal{U}, s, I) \). An element \( x \in \mathcal{U} \) is allocated in \( (\mathcal{U}, s, I) \) (resp. points to \( y \) in \( (\mathcal{U}, s, I) \)) if there exists \( y \in \mathcal{U}^k \) such that \( (x, y) \in p^f \) (resp. if \( (x, y) \in p^f \)).

**Definition 4.6.** For a quantified boolean combination of test formulae \( \phi \), we let \( N(\phi) \) be the maximum integer \( n \) occurring in a test formula \( \theta \) of the form \( |h| \geq n, |U| \geq n \), or \(|h| \geq |U| - n \) from \( \phi \) and define \( \mathcal{A}(\phi) \equiv \text{Heap} \land \land_{i=0}^{N(\phi)} A_i \land \land_{i=0}^{N(\phi)} B_i \land \land_{i=0}^{N(\phi)+1} C_i \land \text{Dom} \land \land_{x \in \text{Var}(\phi)} \text{Dom}_x \) as the conjunction of axioms related to \( \phi \).

**Example 4.7.** Let \( \phi \) be the SL\(^1\) formula: \( x \rightarrow y \land |h| \geq 2 \land |U| < |U| \). Then \( \tau(\phi) = p(x, y) \land a_2 \land c_1 \), and \( \mathcal{A}(\phi) \) contains, among others, the following formulae:\

\[ \forall x, y, z \cdot p(x, y) \land p(x, z) \rightarrow y \equiv z \]

\[ a_0 \land (a_1 \rightarrow a_0 \land p(u_1, u_0)) \land (a_2 \rightarrow a_1 \land p(u_2, u_1) \land \neg u_1 \equiv u_2) \]

\[ c_0 \land (c_1 \rightarrow c_0 \land \neg p(w_1, y)) \]

The formula \( \tau(\phi) \land \mathcal{A}(\phi) \) states that \( p(x, y) \) holds, that \( p \) is a partial function and that there exist at least two distinct allocated elements (namely \( u_1 \) and \( u_2 \)) and one unallocated element (\( w_1 \)).

Let \( \phi' \) be the SL\(^2\) formula \( \text{alloc}(u) \land \forall y \cdot \neg u \equiv y \rightarrow \neg \text{alloc}(y) \). Then \( \tau(\phi') = b(u) \land (\forall y \cdot \neg u \equiv y \rightarrow \neg b(y)) \), where the relevant axioms in \( \mathcal{A}(\phi') \) are:

\[ \forall x, y_1, y_2, z_1, z_2 \cdot p(x, y_1, y_2) \land p(x, z_1, z_2) \rightarrow y_1 \equiv z_1 \land y_2 \equiv z_2 \]

\[ \forall x, y_1, y_2 \cdot p(x, y_1, y_2) \rightarrow b(x) \]

\[ b(u) \rightarrow p(u, f_1, f_2) \]

The relationship between \( \phi \) and \( \tau(\phi) \) is stated below.

\(^{5}\)The converse of \( C_n \), \( \forall x \cdot (\neg c_n \land \forall y \cdot \neg p(x, y)) \rightarrow \land_{i=1}^{n-1} x \equiv w_i \) is not in BSR(FO).

\(^{6}\)For simplicity, only the relevant axioms are given.

\(\text{Definition 4.8.} \) A formula \( \phi \) is BSR-compatible if: (i) each test formula \(|h| \geq |U| - n\) in \( \phi \) occurs at a negative polarity (ii) if a formula \( \text{alloc}(x) \) occurs at positive polarity in \( \phi \), then \( x \in \text{Var}(\phi) \).

\(\text{Lemma 4.9.} \) Let \( \phi \) be a quantified boolean combination of test formulæ. The following hold, for any universe \( U \) and any store \( s \):

1. \( (U, s, I, b) \models \phi \), for a heap \( b \), then \( (U, s, J) \models \tau(\phi) \land A(\phi) \) for an interpretation \( J \) coinciding with \( I \) on every symbol not occurring in \( \phi \) and such that \( b \) is associated with \( (U, s, J) \);
2. \( \phi \) is BSR-compatible and \( (U, s, I) \models \tau(\phi) \land A(\phi) \) for an interpretation \( I \) such that \( ||p^J|| \in \mathbb{N} \), then \( (U, s, I, b) \models \phi \), where \( b \) denotes the heap associated with \( (U, s, I) \).

\(\text{Proof.} \) (1) Let \( (U, s, I, b) \) be a model of \( \phi \). Let \( J \) be an interpretation coinciding with \( I \) on every symbol occurring in \( \phi \), and extended to the symbols \( v, a_i, b_j, c_k, u_i, v_j, w_i \), for \( i \in [0, N(\phi)] \) and \( j \in [0, N(\phi) + 1] \), as follows:

\[a_i^J \defeq \begin{cases} \top & \text{if } 0 \leq i \leq \min(||b||, N(\phi)) \\ \bot & \text{if } i > \min(||b||, N(\phi)) \end{cases}\]

\[b_j^J \defeq \begin{cases} \top & \text{if } 0 \leq i \leq \min(||b||, \min(||b||, N(\phi))) \\ \bot & \text{if } i > \min(||b||, N(\phi)) \end{cases}\]

\[c_k^J \defeq \begin{cases} \top & \text{if } 0 \leq i \leq \min(||b||, ||b||, N(\phi) + 1) \\ \bot & \text{if } i > \min(||b||, ||b||, N(\phi) + 1) \end{cases}\]

The constants of sort \( U \) are interpreted as locations, and the symbols \( v, a_i, b_j, c_k, u_i, v_j, w_i \) are interpreted as locations, as follows:

- \( v^J, \ldots, v^J_{\min(||b||, N(\phi))} \) are pairwise distinct locations in \( \text{dom}(b) \) and \( v^J_i \) is the \( i \)-th component of the vector referred to by \( v \).
- \( w^J, \ldots, w^J_{\min(||b||, N(\phi))} \) are pairwise distinct locations in \( U \).
- \( w^J, \ldots, w^J_{\min(||b||||b||, N(\phi) + 1)} \) are pairwise distinct locations in \( U \setminus \text{dom}(b) \).

The other symbols are interpreted arbitrarily. It is straightforward to check that \( (U, s, J) \models A(\phi) \). We prove that \( (U, s, I, b) \models \psi \iff (U, s, J) \models \tau(\psi) \) for every subformula \( \psi \) of \( \phi \) by induction on the structure of \( \phi \):

- \( \psi = x \approx y \): We have \( \tau(\psi) = \psi \). Further, \( (U, s, I, b) \models \psi \iff s(x) = s(y) \iff (U, s, J) \models \psi \).
- \( \psi = q(x_1, \ldots, x_n) \): We have \( \tau(\psi) = \psi \). Moreover, \( (U, s, I, b) \models \psi \iff (s(x_1), \ldots, s(x_n)) \in q^J \) and \( (U, s, J) \models \psi \iff (s(x_1), \ldots, s(x_n)) \in q^J \). Because \( I \) and \( J \) coincide on every symbol occurring in \( \psi \), \( q^J = q^J \). Thus \( (U, s, I, b) \models \psi \iff (U, s, J) \models \psi \).
- \( \psi = |h| \geq n: (U, s, J, b) \models \psi \iff ||b|| \geq n \) by Proposition 4.2. Since \( n \leq N(\psi) \), we have \( ||b|| \geq n \iff n \leq \min(||b||, N(\psi)) \), thus \( a_n^J = \top \iff (U, s, J) \models \tau(\psi) \).
- \( \psi = |U| \geq n: (U, s, J, b) \models \psi \iff ||b|| \geq n \) by Proposition 4.2. Since \( n \leq N(\psi) \), we have \( ||b|| \geq n \iff n \leq \min(||b||, N(\psi)) \), thus \( b_{2n}^J = \top \iff (U, s, J) \models \tau(\psi) \).
- \( \psi = |h| \leq |U| - n: (U, s, J, b) \models \psi \iff ||b|| \leq ||U|| - n \) by Proposition 4.2, i.e., \( \iff n + 1 > ||b|| \). Since \( n \leq N(\psi) \), we have \( (U, s, I, b) \models \psi \iff n + 1 > \min(||U|| - ||b||, N(\psi) + 1) \iff c_{n+1}^J = \top \iff (U, s, J) \models \tau(\psi) \).
- \( \psi = \text{alloc}(x): (U, s, J, b) \models \psi \iff s(x) \in \text{dom}(b) \) iff \( s(x) \in b^J \) iff \( (U, s, J) \models \psi \).
- The cases \( \psi = \gamma \land \phi \), \( \phi = \lnot \gamma \), \( \psi = \exists x. \gamma \) are by the inductive hypothesis, since \( (U, s, I, b) \models \psi \iff (U, s, J) \models \tau(\psi) \), for all \( l \leq 1, 2 \).

(2) Let \( (U, s, I) \) be a model of \( \tau(\phi) \land A(\phi) \), such that \( ||p^J|| \in \mathbb{N} \). We define a heap \( b \) as follows: for each \( (k + 1) \)-tuple of locations \( \ell_0, \ldots, \ell_k \in U \) such that \( (\ell_0, \ldots, \ell_k) \in \mathcal{V} \), we set \( b(\ell_0) \defeq (\ell_1, \ldots, \ell_k) \). Since \( (U, s, I) \models \text{Heap} \).
and \(|\varphi|\) \in \mathbb{N}, \(b\) is a finite partial function. Let \(\phi_{\text{nf}}\) be the negation normal form of \(\phi\). It is easy to check that \(\tau(\phi_{\text{nf}}) \equiv \tau(\phi)\). We prove that \((I, s, I, h) \models \tau(\psi) \Rightarrow (I, s, I, h) \models \psi\) for every subformula \(\psi\) in \(\phi_{\text{nf}}\):

- \(\psi = [h] \geq n: \tau(\psi) = a_n\) and \((I, s, I, h) \models a_n \Rightarrow a_n^I = \top\). Since \(n \leq N(\psi)\) and \((I, s, I) \models \bigwedge_{i=0}^{N(\psi)} A_j\), we have \(a_j^I = \top\) and \(a_j^I \in \text{dom}(b)\), for all \(j \in [1, n]\). Because \(a_j^I\) are pairwise distinct, for \(j \in [1, n]\), we obtain that \(|\text{dom}(b)| \geq n\), and \((I, s, I, h) \models \psi\), by Proposition 4.2.

- \(\psi = [h] < n: \tau(\psi) = a_n\) and \((I, s, I, h) \models a_n \Rightarrow a_n^I = \bot\). Since \(n \leq N(\psi)\) and \((I, s, I) \models \bigwedge_{i=0}^{N(\psi)} A_j\), each location \(j \in \text{dom}(b)\) must be one of \(u_j^I, \ldots, u_{n-1}^I\), thus \(|\text{dom}(b)| \leq n - 1\) and \((I, s, I, h) \models |h| < n\), follows, by Proposition 4.2.

- \(\psi = [U] \geq n: \tau(\psi) = b_n\) and \((I, s, I, h) \models b_n \Rightarrow b_n^I = \top\). Since \(n \leq N(\psi)\) and \((I, s, I) \models \bigwedge_{i=0}^{N(\psi)} B_j\), we have \(b_j^I = \top\), for all \(j \in [1, n]\). Because \(b_j^I\) are pairwise distinct, for all \(j \in [1, n]\), we obtain that \(|U| \geq n\), and \((I, s, I, h) \models \psi\), by Proposition 4.2.

- \(\psi = [U] < n: \tau(\psi) = b_n\) and \((I, s, I, h) \models b_n \Rightarrow b_n^I = \bot\). Since \(n \leq N(\psi)\) and \((I, s, I) \models \bigwedge_{i=0}^{N(\psi)} B_j\), we have that each location \(j \in U\) must be one of \(v_j^I, \ldots, v_{n-1}^I\), thus \(|U| \leq n - 1\) and \((I, s, I, h) \models |h| < n\), follows, by Proposition 4.2.

- \(\psi = [h] \geq |U| - n: \tau(\psi) = c_{n+1}\) and \((I, s, I, h) \models c_{n+1} \Rightarrow c_{n+1}^I = \top\). Since \(n \leq N(\psi)\) and \((I, s, I) \models \bigwedge_{i=0}^{N(\psi)+1} C_j\), we obtain that \(a_j^I \in \text{dom}(b)\), for all \(j \in [1, n+1]\). Since \(a_j^I\) are pairwise disjoint, we obtain \(|U| - |h| \geq n + 1\) thus \((I, s, h) \models \psi\), follows, by Proposition 4.2.

- \(\psi = \text{alloc}(x)\). Since \(\psi\) occurs at positive polarity and \(\phi\) is BSR-compatible, necessarily \(x \in \text{Var}(\phi)\). Since \((I, s, I) \models b(x)\) and \((I, s, I) \models Dom_x\), we have \((I, s, I) \models \forall x. c(x, x_1^{\phi}, \ldots, x_k^{\phi})\), and therefore \(s(x) \in \text{dom}(b)\). Thus \((I, s, I, h) \models \psi\).

- \(\psi = \neg\text{alloc}(x)\). Since \((I, s, I) \models b(x)\) and \((I, s, I) \models Dom\), we have \((I, s, I, h) \models \forall y_1, \ldots, y_k. \neg v(x, y_1, \ldots, y_k)\), thus \(s(x) \notin \text{dom}(b)\). Hence \((I, s, I, h) \models \psi\).

- \(\psi \in \{x \equiv y, \neg x \equiv y, q(x), \neg q(x), x \leftrightarrow y, \neg x \leftrightarrow y\}\): The equivalence statement \((I, s, I, h) \models \psi \Rightarrow (I, s, I) \models \psi\) is proven in the same way as for point (1).

- The cases \(\psi = \psi_1 \land \psi_2, \psi = \psi_1 \lor \psi_2, \exists x. \psi_1\) are by inductive hypothesis.

\(\square\)

The following proposition states essential syntactic properties of \(\tau(\phi) \land A(\phi)\).

**Proposition 4.10.** Let \(\varphi = \forall y. \phi\), where \(\phi\) is a boolean combination of test formulae, with \(\text{Var}(\phi) = \{x_1, \ldots, x_n\}\).

The formula \(\tau(\phi) \land A(\phi)\) is a BSR(FO) formula with no existential quantifier such that \(|\text{Const}(\tau(\phi) \land A(\phi))| = k \cdot n + (k + b) \cdot N(\phi) + 5\) (where \(k\) denotes the number of record fields) and \(|\text{Var}(\tau(\phi) \land A(\phi))| = \text{Var}(\phi)\).

**Proof.** The proof is by a straightforward inspection of \(\tau(\phi)\) and of the axioms in \(A(\phi)\). There are \(k \cdot n\) constants \(c_{x_1, \ldots, x_n}^{\phi} N(\phi) + 1\) constants \(a_1, b_1, \ldots, a_n, b_n, \phi_1, \ldots, \phi_n\), \(N(\phi)\) constants \(a_i, b_i, \phi_i, a_i^I, b_i^I + 2\) constants \(c_i, k \cdot N(\phi)\) constants \(u_i^I\). \(\square\)

5 FROM QUANTIFIER-FREE SL\(^k\) TO TEST FORMULA

This section establishes the expressiveness completeness result of the paper, namely that any quantifier-free SL\(^k\) formula is equivalent, on both finite and infinite models, to a boolean combination of test formulae. Starting from a quantifier-free SL\(^k\) formula \(\phi\), we define a set \(\mu(\phi)\) of conjunctions of test formulae and their negations, called

"minterms", such that \(\phi \equiv \bigvee_{M \in \mu(\phi)} M\). The definition of \(\mu(\phi)\) depends on the cardinality of the universe (finite or

The number of minterms in \(\mu(\phi)\) is exponential in the size of \(\phi\), however, the size of every \(M \in \mu(\phi)\) is bounded by a polynomial in the size of \(\phi\) and, as we show, checking the membership of a given minterm \(M\) in

\(\mu(\phi)\) can be done in PSPACE.

5.1 Minterms

We introduce some definitions and notations, and establish basic properties.

Definition 5.1. A literal is a test formula or its negation. A minterm $M$ is a set of literals, interpreted as the conjunction of its elements, that contains:

- at most one literal of the form $|U| \geq n$;
- at most one literal of the form $|U| < n$;
- exactly one literal $|h| \geq \min_M$, where $\min_M \in \mathbb{N} \cup \{|U| - n \mid n \in \mathbb{N}\}$;
- exactly one literal $|h| < \max_M$, where $\max_M \in \mathbb{N}_\infty \cup \{|U| - n \mid n \in \mathbb{N}\}$.

Definition 5.2. Given a minterm $M$, we define the sets:

- $M^c \overset{\text{def}}{=} M \cap \{x \approx y, \neg x \approx y \mid x, y \in \text{Var}\}$
- $M^a \overset{\text{def}}{=} M \cap \{\text{alloc}(x), \neg\text{alloc}(x) \mid x \in \text{Var}\}$
- $M^u \overset{\text{def}}{=} M \cap \{|U| \geq n, |U| < n \mid n \in \mathbb{N}\}$
- $M^p \overset{\text{def}}{=} M \cap \{x \leftrightarrow y, \neg x \leftrightarrow y \mid x, y \in \text{Var}^{k+1}\}$
- $M^f \overset{\text{def}}{=} M \cap \{\sigma(q) = B, x \in \text{Var}^q\}$

Thus, $M = M^c \cup M^u \cup M^a \cup M^p \cup M^f \cup \{|h| \geq \min_M, |h| < \max_M\}$, for each minterm $M$.

Proposition 5.3. Given a minterm $M$, for all structures $S = (\mathcal{U}, s, I, h)$ and $S' = (\mathcal{U}, s, I, h')$ we have $S \models M^c \wedge M^a \wedge M^f \iff S' \models M^c \wedge M^a \wedge M^f$.

Proof. This is immediate, since the semantics of the test formulæ in $M^c \wedge M^a \wedge M^f$ does not depend on the heap. \hfill $\Box$

Definition 5.4. Given a set of variables $X \subseteq \text{Var}$, a minterm $M$ is (1) E-complete for $X$ iff for all $x, y \in X$, exactly one of $x \approx y \in M$, $\neg x \approx y \in M$ holds, and (2) A-complete for $X$ if for each $x \in X$ exactly one of alloc$(x) \in M$, $\neg$alloc$(x) \in M$ holds.

For a literal $\ell$, we denote by $\overline{\ell}$ its complement, i.e., $\overline{\theta} \overset{\text{def}}{=} \neg \theta$ and $\neg \theta \overset{\text{def}}{=} \theta$, where $\theta$ is a test formula. If $T$ is a set of literals, then we denote by atoms$(T)$ the set of all test formulæ $\phi$ such that either $\phi$ or $\neg\phi$ occurs in $T$. The equivalence relation $x \equiv_T y$ is defined as $T \models x \approx y$ and we write $x \not\equiv_T y$ for $T \models \neg x \approx y$. Observe that $x \not\equiv_T y$ is not the complement of $x \equiv_T y$. For a set $X$ of variables, $|X|_T$ is the number of equivalence classes of $\equiv_T$ in $X$.

Two tuples $y, y' \in \text{Var}^k$ are $T$-distinct if $y_i \not\equiv_T y'_i$, for some $i \in [1, k]$.

Proposition 5.5. If $M$ is E-complete for Var$(M)$, $(\mathcal{U}, s, I, h) \models M$ and $X \subseteq \text{Var}(M)$, then $|X|_M = ||s(X)||$.

Proof. This is an immediate consequence of the fact that if $x, x' \in X$, then $s(x) = s(x')$ if and only if $M \models x \approx x'$. \hfill $\Box$

Definition 5.6. For a set $T$ of literals, let:

- $\text{av}(T) \overset{\text{def}}{=} \{x \in \text{Var} \mid \exists x' \in \text{Var} \cdot x \equiv_T x', T \cap \{\text{alloc}(x'), x' \leftrightarrow y \mid y \in \text{Var}^k\} \neq \emptyset\}$
- $\text{nv}(T) \overset{\text{def}}{=} \{x \in \text{Var} \mid \exists x' \in \text{Var} \cdot x \equiv_T x', \neg\text{alloc}(x') \in T\}$
- $\text{fp}_X(T) \overset{\text{def}}{=} T \cap \{\text{alloc}(x), \neg\text{alloc}(x), x \leftrightarrow y, \neg x \leftrightarrow y \mid x \in X, y \in \text{Var}^k\}$
- $\#_a(T) \overset{\text{def}}{=} |\text{av}(T)|_T$
- $\#_s(X, T) \overset{\text{def}}{=} |X \cap \text{nv}(T)|_T$

For notational convenience, we also let $\text{fp}_s(T) \overset{\text{def}}{=} \text{fp}_s(T)(T)$.
Intuitively, $\text{av}(T)$ (resp. $\text{nv}(T)$) is the set of variables that must be (resp. are never) allocated in every (resp. any) model of $T$. The symbol $\#_a(T)$ represents the number of equivalence classes of $\approx_T$ containing variables allocated in every model of $T$; $\#_a(X,T)$ represents the number of equivalence classes of $\approx_T$ containing variables from $X$ that are not allocated in any model of $T$ and $\text{fp}_X(T)$ is the footprint of $T$ relative to the set $X \subseteq \text{Var}$, i.e. the set of formulæ describing allocation and points-to relations over variables from $X$. For example, if $T = \{x \approx z, \text{alloc}(x), \neg\text{alloc}(y), \neg z \leftarrow y\}$, then $\text{av}(T) = \{x, z\}; \text{nv}(T) = \{y\}; \#_a(T) = 1; \#_a([y], T) = 1; \text{fp}_a(T) = \{\text{alloc}(x), \neg z \leftarrow y\}$ and $\text{fp}_{\text{nv}(T)}(T) = \{\neg\text{alloc}(y)\}$.

**Proposition 5.7.** Given a set $T$ of test formulæ and a structure $(\llbracket I, s, \bar{h} \rrbracket)$, if $(\llbracket I, s, \bar{h} \rrbracket) \models \text{fp}_a(T)$, then $(\llbracket I, s, \bar{h} \rrbracket, b') \models \text{fp}_a(T)$ for every extension $b'$ of $b$.

**Proof.** Assume that $(\llbracket I, s, \bar{h} \rrbracket) \models \text{fp}_a(T)$ and let $\phi \in \text{fp}_a(T)$. If $\phi$ is of the form $\neg\text{alloc}(x)$, then since $x \in \text{av}(T)$, necessarily, $T$ contains an atom of the form $\text{alloc}(x')$ or $x' \leftarrow y$, where $x'$ is a variable such that $x' \approx_T x$. In both cases, $\text{fp}_a(T)$ must be unsatisfiable, contradicting the assumption that $(\llbracket I, s, \bar{h} \rrbracket) \models \text{fp}_a(T)$. If $\phi$ is of the form $x \leftarrow y$, then, since $(\llbracket I, s, \bar{h} \rrbracket, b') \models \phi$, we have $b(s(x)) \equiv s(y)$, thus $b'(s(x)) \equiv s(y)$ (since $b'$ is an extension of $b$) so that $(\llbracket I, s, \bar{h} \rrbracket, b') \models \phi$. The proof is similar if $\phi = \text{alloc}(x)$. If $\phi = \neg x \leftarrow y$ and $T$ contains an atom of the form $\text{alloc}(x')$ for some variable $x'$ such that $x \approx_T x'$, then $s(x) \in \text{dom}(b)$ and $b(s(x)) \neq s(y)$. This entails that $b'(s(x)) \neq s(y)$ (since $b'$ is an extension of $b$) and $(\llbracket I, s, \bar{h} \rrbracket, b') \models \phi$. Otherwise, because $x \in \text{av}(T)$, $T$ must contain an atom of the form $x' \leftarrow y'$ for some variable $x'$ such that $x \approx_T x'$. Thus, $b'(s(x)) = s(y') \neq s(y)$, and we deduce that $b'(s(x)) \neq s(y)$. □

**Definition 5.8.** Given the minterms $M_1, M_2$, let $\text{npto}(M_1, M_2) \overset{\text{def}}{=} (M_1 \cap M_2) \cap \{ \neg x \leftarrow y \mid x \notin \text{av}(M_1 \cup M_2), y \in \text{Var}_k \}$ be the set of negative points-to literals common to $M_1$ and $M_2$, involving left-hand side variables not allocated in either $M_1$ or $M_2$.

For example, if $M_1 = \{x \leftarrow y, \neg y \leftarrow z, \neg y \leftarrow u, \neg z \leftarrow u, |h| \geq 1, |h| < \infty\}$ and $M_2 = \{x \leftarrow y, \neg y \leftarrow z, \neg z \leftarrow u, \text{alloc}(z), |h| \geq 1, |h| < \infty\}$, then $\text{npto}(M_1, M_2) = \{\neg y \leftarrow z\}$. Observe that $M_1 \cap M_2$ necessarily entails $\text{npto}(M_1, M_2)$, since the assertion $y \leftarrow z$ cannot hold in any part of the heap.

We now introduce some conditions that are necessary for a minterm to be satisfiable. The first condition is that the same element cannot point to distinct vectors.

**Definition 5.9.** Given a minterm $M$, its points-to closure is $\text{pc}(M) \overset{\text{def}}{=} \bot$ if there exist literals $x \leftarrow y, x' \leftarrow y' \in M$ such that $x \approx_M x'$ and $y, y'$ are $M$-distinct; and $\text{pc}(M) \overset{\text{def}}{=} M$, otherwise.

Intuitively, $\text{pc}(M)$ is $\bot$ iff $M$ contradicts the fact that the heap is a partial function. For instance, let $M = \{x \leftarrow (y_1, y_2), x' \leftarrow (y'_1, y'_2), x \approx x', \neg y_1 \approx y'_1, |h| \geq 1, |h| < \infty\}$. We have $\text{pc}(M) = \bot$, and it is clear that $M$ is unsatisfiable as the same location cannot point to both $(y_1, y_2)$ and $(y'_1, y'_2)$. Note that we do not assert the equality $y \approx y'$, instead we only check that it is not falsified. This is sufficient for our purpose because in the following we always assume that the considered minterms are $E$-complete.

The second condition is that the alloc and point-to literals should be consistent:

**Definition 5.10.** A minterm $M$ is footprint-consistent if for all $x, x' \in \text{Var}$ and $y, y' \in \text{Var}_k$, such that $x \approx_M x'$ and $y_i \approx_M y'_i$ for all $i \in [1, k]$, we have (1) if $\text{alloc}(x) \in M$ then $\neg\text{alloc}(x') \notin M$, and (2) if $x \leftarrow y \in M$ then $\neg\{\neg\text{alloc}(x'), \neg x' \leftarrow y'\} \cap M = \emptyset$.

**Proposition 5.11.** If $M$ is a footprint-consistent minterm, then $\text{nv}(M) \cap \text{av}(M) = \emptyset$. If, moreover, $M$ is $E$-complete for $\text{Var}(M)$, then $s(X) \cap s(\text{av}(M)) = \emptyset$ for each set $X$ disjoint from $\text{av}(M)$ and each model $(\llbracket I, s, \bar{h} \rrbracket)$ of $M$.

**Proof.** Suppose first that $x \in \text{nv}(M) \cap \text{av}(M)$. Then there exist literals $\neg\text{alloc}(x')$ and $\text{alloc}(x'')$ in $M$ such that $x \approx_M x'$ and $x \approx_M x''$, which contradicts the footprint consistency of $M$. For the second point, suppose
that $\ell \in s(X) \cap s(\text{av}(M))$. Then there exist variables $x \in X$ and $x' \in \text{av}(M)$ such that $s(x) = s(x') = \ell$. If $M$
 is E-complete, either $x \approx x' \in M$ or $\neg x \approx x' \in M$. The first case contradicts $x \not\in \text{av}(M)$ and the second case
contradicts $(\mathcal{U}, s, I, h) \models M$. \hfill \square

Footprint-consistency is not sufficient for satisfiability. For example, $\{x \leftarrow y, x' \leftarrow y', \neg y \approx y', |h| < 2\}$ is at
the same time footprint-consistent and unsatisfiable, because $x$ and $x'$ point to distinct elements but there is
at most one allocated location. We thus introduce additional conditions related to the cardinality of the heap
of the universe. Intuitively, for any minterm $M$, we define a formula $\text{dc}(M)$ that asserts that $\min_M < \max_M$
and that the domain contains enough elements to allocate all cells. Essentially, given a structure $(\mathcal{U}, s, I, h)$, if
$b(x)$ is known to be defined and distinct from $n$ pairwise distinct vectors of locations $v_1, \ldots, v_n$, then necessarily
at least $n + 1$ vectors must exist. Since there are $||\mathcal{U}||^k$ vectors of length $k$, we must have $||\mathcal{U}||^k \geq n + 1$, hence
$||\mathcal{U}|| \geq \sqrt[k]{n+1}$. For instance, if

$$M = \{\neg x \leftarrow y_i \mid i \in [1, n]\} \cup \{\text{alloc}(x)\} \cup \{y_i \not\approx y_j \mid i, j \in [1, n], i \neq j\}$$
then it is clear that $M$ is unsatisfiable if there are less than $n$ locations, since $x$ cannot be allocated in this case.

Definition 5.12. Given a minterm $M$, the domain closure of $M$ is $\text{dc}(M) \triangleq \bot$ if either $\min_M = n_1$ and $\max_M = n_2$
for some $n_1, n_2 \in \mathbb{Z}$ such that $n_1 \geq n_2$, or $\min_M = |U| - n_1$ and $\max_M = |U| - n_2$, where $n_2 \geq n_1$; and otherwise:

$$\text{dc}(M) \triangleq \mathcal{U} \cup \{\{U\} \geq \sqrt[k]{\max x\in\text{av}(M)}(\delta_x(M) + 1)\}$$
$$\cup \{|U| \geq n_1 + n_2 + 1 \mid \min_M = n_1, \max_M = |U| - n_2, n_1, n_2 \in \mathbb{N}\}$$
$$\cup \{|U| < n_1 + n_2 \mid \min_M = |U| - n_1, \max_M = n_2, n_1, n_2 \in \mathbb{N}\},$$

where $\delta_x(M)$ is the number of pairwise $M$-distinct tuples $y$ for which there exists $\neg x' \leftarrow y \in M$ such that
$x \approx_M x'$. For any SL-structure $S = (\mathcal{U}, s, I, h)$, we denote by $\min_S^S, \max_S^S \in \mathbb{N}_\infty$ the values obtained by replacing
$\mathcal{U}$ with $||\mathcal{U}||$ in $\min_M$ and $\max_M$, respectively.

Example 5.13. Let $M = \{|h| \geq 0, |h| < \infty, \text{alloc}(y_0)\} \cup \{|y_i \approx y_j \mid i, j \in [0, n], i \neq j\} \cup \{y_0 \leftarrow y_i \mid i \in [1, n]\}$. Then $y_0 \in \text{av}(M)$, $\delta_x(M) = n$ and $\text{dc}(M) = \mathcal{U} \cup \{|U| \geq n + 1\}$. This states that all models of $M$ contain at least
$n + 1$ locations: $y_1, \ldots, y_n$ and the image of $y_0$ by the heap.

Let $M' = \{|h| \geq 1, |h| < |U| - 1\}$. Then $\text{dc}(M') = M' \cup \{|U| \geq 3\}$. All models of $M'$ contain at least 3 locations
(one allocated and two non allocated). \hfill \blacksquare

Proposition 5.14. Given a minterm $M$, $\min_S^S < \max_S^S$ for every model $S$ of $\text{dc}(M)^n$.

Proof. Let $S = (\mathcal{U}, s, I, h)$ and $n_1, n_2 \in \mathbb{N}_\infty$. We distinguish the following cases:

- If $\min_M = n_1$ and $\max_M = n_2$ then $n_1 < n_2$ must be the case, or else $\text{dc}(M) \equiv \bot$, in contradiction with
  $S \models \text{dc}(M)^n$.
- If $\min_M = n_1$ and $\max_M = |U| - n_2$ then $|U| = n_1 + n_2 + 1 \in \text{dc}(M)$ and since $S \models \text{dc}(M)^n$, we obtain
  $n_1 < ||\mathcal{U}|| - n_2$.
- If $\min_M = |U| - n_1$ and $\max_M = n_2$ then $|U| < n_1 + n_2 \in \text{dc}(M)$ and since $S \models \text{dc}(M^n)$, we obtain
  $||\mathcal{U}|| - n_1 < n_2$.
- If $\min_M = |U| - n_1$ and $\max_M = |U| - n_2$ then $n_2 < n_1$ must be the case, or else $\text{dc}(M) \equiv \bot$, in contradiction
  with $S \models \text{dc}(M)^n$. \hfill \blacksquare

Proposition 5.15. For any minterm $M$, we have $M \equiv \text{pc}(M) \equiv \text{dc}(M)$.

Proof. It is clear that $\text{pc}(M) \models M$ and $\text{dc}(M) \models M$. Let $S = (\mathcal{U}, s, I, h)$ be a model of $M$. If $S \not\models \text{pc}(M)$ then
necessarily $\text{pc}(M) = \bot$ and there exist variables $x, x' \in \text{Var}(M)$ such that $x \leftarrow (y_1, \ldots, y_k), x' \leftarrow (z_1, \ldots, z_k) \in$

We have \( b(s(x)) = (s(y_1), \ldots, s(y_k)) \) and \( b(s(x')) = (s(y_1), \ldots, s(y_k)) \) and \( b(s(x')) = b(s(x')) \), thus \( s(y_1) = s(z_1) \), for all \( i \in [1, k] \), a contradiction. Thus \( S \models \neg \sigma \). For a variable \( x \in \text{av}(M) \), let \( \neg x_1 \mapsto y_1, \ldots, \neg x_n \mapsto y_n \in M \) be all literals such that \( x_1 \approx x \), \( x_n \approx x \) and \( y_i \neq y_j \) for all \( i \neq j \). Then \( b(s(x)) \in \{ y \} \setminus \{ s(y_1), \ldots, s(y_n) \} \), thus \( ||U||^k \geq n + 1 = \delta_x(M) + 1 \). Since this holds for each \( x \in \text{av}(M) \), we have \( S \models |U| \geq \left( \frac{\max_{x \in \text{av}(M)} \delta_x(M) + 1}{n} \right)^k \).

Furthermore, if \( |h| \geq n_1, |h| < |U| - n_2 \) in then, since \( S \models M \), \( ||U|| - n_2 > ||h|| \geq n_1 \), thus \( ||U|| \geq n_1 + n_2 + 1 \) and \( S \models |U| \geq n_1 + n_2 + 1 \). Analogously, we obtain \( S \models |U| < n_1 + n_2 + 1 \) in the case \( |h| < n_1, |h| \geq |U| - n_2 \in M \). \( \square \)

5.2 Eliminating Spatial Connectives

We now show how to eliminate the connectives \( \ast \) and \( \rightarrow \), i.e., to transform a formula of the form \( \phi_1 \ast \phi_2 \) or \( \phi_1 \rightarrow \phi_2 \) into an equivalent boolean combination of test formulae, assuming \( \phi_1 \) and \( \phi_2 \) have already been transformed. We solve this problem by restricting ourselves to the case where \( \phi_1 \) and \( \phi_2 \) are minterms satisfying some additional properties. We first consider the separating conjunction.

**Lemma 5.16.** Let \( M_1, M_2 \) be two minterms that are footprint-consistent and E-complete for \( \text{Var}(M_1 \cup M_2) \), with atoms(\( M_1 \)) = atoms(\( M_2 \)). Then \( M_1 + M_2 \equiv \text{elim}_s(M_1, M_2) \), where

\[
\text{elim}_s(M_1, M_2) \overset{\text{def}}{=} M_1^c \wedge M_2^c \wedge M_1^f \wedge M_2^f \wedge \text{dc}(M_1)^u \wedge \text{dc}(M_2)^u \wedge \\
\bigwedge_{x \in \text{av}(M_1), y \in \text{av}(M_2)} \neg x \approx y \wedge \text{fp}_a(M_1) \wedge \text{fp}_a(M_2) \wedge \\
\text{naloc}(\text{nv}(M_1) \cap \text{nv}(M_2)) \wedge \text{npto}(M_1, M_2) \wedge \\
|h| \geq \min_{M_1} + \min_{M_2} \wedge |h| < \max_{M_1} + \max_{M_2} - 1 \\
\wedge \eta_{12} \wedge \eta_{21}
\]

**Intuitively,** if \( M_1 \) and \( M_2 \) hold separately, then all heap-independent literals from \( M_1 \cup M_2 \) must be satisfied (2), the variables allocated in \( M_1 \) and \( M_2 \) must be pairwise distinct and their footprints, relative to the allocated variables, jointly asserted (3). Moreover, unallocated variables on both sides must not be allocated and common negative points-to literals must be asserted (4). Since the heap satisfying \( \text{elim}_s(M_1, M_2) \) is the disjoint union of the heaps for \( M_1 \) and \( M_2 \), its bounds are the sum of the bounds on both sides (5) and the variables that \( M_2 \) never allocates (the set \( \text{nv}(M_2) \)) may occur allocated in the heap of \( M_1 \) and vice versa, thus the constraints \( \eta_{12} \) and \( \eta_{21} \), respectively (6).

The proof of Lemma 5.16 requires the following result:

**Proposition 5.17.** Let \( M_1, M_2 \) be two minterms that are footprint-consistent and E-complete for \( \text{Var}(M_1 \cup M_2) \) and let \( \mathcal{S} = (U, s, I, h) \) be a model of \( \text{elim}_s(M_1, M_2) \). Let \( L_i, Y_i, A_i \) be the following sets, for \( i = 1, 2 \):

\[
L_i = \{ s(x) \in \text{dom}(h) \mid x \in \text{nv}(M_{3-i}) \setminus \text{av}(M_i) \} \\
Y_i = \{ x \in \text{Var} \mid s(x) \in L_i \} \\
A_i = \{ s(x) \mid x \in \text{av}(M_i) \}
\]

Then \( L_1 \cap L_2 = \emptyset, L_i \cap (A_1 \cup A_2) = \emptyset \) (for \( i = 1, 2 \)) and \( \mathcal{S} \models \text{alloc}(Y_1) \wedge \text{alloc}(Y_2) \).

**Proof.** We have the following results:

\* \( L_1 \cap L_2 = \emptyset \). By contradiction, suppose that there exists \( \ell \in L_1 \cap L_2 \). Then \( \ell = s(y_1) = s(y_2) \) for some \( y_1 \in \text{nv}(M_1) \) and \( y_2 \in \text{nv}(M_2) \). Because \( M_1 \) is E-complete for \( \text{Var}(M_1 \cup M_2) \), exactly one of \( y_1 \approx y_2 \), \( \neg y_1 \approx y_2 \) belongs to \( M_1 \). But \( \neg y_1 \approx y_2 \in M_1 \) contradicts \( s(y_1) = s(y_2) \) and \( y_1 \approx y_2 \in M_1 \) leads to
Thus we consider from now on that \( M_1 \neq M_2 \). Since \( M_1 \) and \( M_2 \) are E-complete for \( \text{Var}(M_1 \cup M_2) \), there must exist a literal \( x \approx y \in M_i \) such that \( \neg x \approx y \in M_2 \), or vice versa. In both cases however \( M_1 \ast M_2 \equiv \text{elim}_{1}(M_1, M_2) \equiv \bot \). Thus we consider from now on that \( M_i = M_2 \).

- \( M_1 \ast M_2 \models \text{elim}_{1}(M_1, M_2) \). Let \( S = (\mathfrak{U}, s, I, b) \) be a model of \( M_1 \ast M_2 \). Then there exist disjoint heaps \( b_1 \) and \( b_2 \) such that \( b = b_1 \uplus b_2 \) and \((\mathfrak{U}, s, I, b_1) \models M_i \), for all \( i = 1, 2 \). Below we show that \( S \) is a model of the formulae (2), (3), (4), (5) and (6).

  (2) Since \((\mathfrak{U}, s, I, b_1) \models M_i^* \land M_j^* \land M_i^\# \), by Proposition 5.3, we also have \((\mathfrak{U}, s, I, b) \models M_i^* \land M_j^* \land M_i^\# \), for \( i = 1, 2 \). By Proposition 5.15, we obtain further that \((\mathfrak{U}, s, I, b) \models \text{dc}(M_i^*)^\# \), for \( i = 1, 2 \).

  (3) Since \( \text{dom}(b_1) \cap \text{dom}(b_2) = \emptyset \), for every \( x \in \text{av}(M_i) \) and \( y \in \text{av}(M_j) \), we must have \( s(x) \neq s(y) \), hence \( S \models \neg x \approx y \). Further, we have \((\mathfrak{U}, s, I, b_1) \models M_i \), thus \((\mathfrak{U}, s, I, b_1) \models \text{fp}_a(M_i) \) and, by Proposition 5.7, \((\mathfrak{U}, s, I, b) \models \text{fp}_a(M_i) \), for \( i = 1, 2 \).

  (4) Consider a variable \( x \in \text{nv}(M_i) \cap \text{nv}(M_j) \). Then there exist variables \( x_1 \) and \( x_2 \) such that \( \neg \text{alloc}(x_1) \in M_i \), \( x \approx_{M_i} x_1 \), \( \neg \text{alloc}(x_2) \in M_j \) and \( x \approx_{M_j} x_2 \). Hence \( s(x) = s(x_1) \not\in \text{dom}(b_1) \) and \( s(x) = s(x_2) \not\in \text{dom}(b_2) \), thus \( s(x) \not\in \text{dom}(b) \) and \((\mathfrak{U}, s, I, b) \models \neg \text{alloc}(x) \). Since \( x \) was chosen arbitrarily, we have \((\mathfrak{U}, s, I, b) \models \text{nlalloc} (\text{nv}(M_i) \cap \text{nv}(M_j)) \). Secondly, let \( \neg x \leftrightarrow y \in M_1 \cap M_2 \), for some \( x \not\in \text{av}(M_1 \cup M_2) \). Since \( \text{dom}(b_1) \cap \text{dom}(b_2) = \emptyset \), only the following are possible:

i. \( s(x) \in \text{dom}(b_1) \). Since \((\mathfrak{U}, s, I, b_1) \models M_i \), we must have \( b_1(s(x)) \not\in s(y) \). Then \( b(s(x)) \not\in s(y) \) thus \((\mathfrak{U}, s, I, b) \models \neg x \leftrightarrow y \).

ii. \( s(x) \not\in \text{dom}(b_1) \) and \( b_2(s(x)) \not\in s(y) \) is symmetrical.

iii. \( s(x) \not\in \text{dom}(b_1) \) and \( b_2(s(x)) \not\in s(y) \) is symmetrical.

Since \( \neg x \leftrightarrow y \in \text{npto}(M_1, M_2) \) was chosen arbitrarily, \((\mathfrak{U}, s, I, b) \models \text{npto}(M_1, M_2) \).

(5) Since \( b = b_1 \uplus b_2 \), we have \( ||b|| = ||b_1|| + ||b_2|| \), thus the first two constraints are obtained by summing up the constraints \( \min_S \leq ||b_i|| < \max_S^S \), for \( i = 1, 2 \).

(6) We prove \( S \models \eta_{12} \), the proof for \( S \models \eta_{21} \) being symmetrical. Consider a set \( Y \subseteq \text{nv}(M_i) \setminus \text{av}(M_i) \) and suppose that \((\mathfrak{U}, s, I, b) \models \text{alloc}(Y) \). For each \( y \in Y \) we must have \( s(y) \in \text{dom}(b_1) \), because \( s(y) \not\in \text{dom}(b_2) \) and \( s(y) \in \text{dom}(b_1) \). Moreover, \( s(Y) \cap s(\text{av}(M_i)) = \emptyset \) because \( Y \cap \text{av}(M_i) = \emptyset \) and \( M_i \) is E-complete for \( \text{Var}(M_1 \cup M_2) \), by Proposition 5.11. Thus \( \#_a(M_i) + ||Y||_{M_i} \leq ||b_1|| < \max_S^S + ||b_2|| \geq \max_S^S \), as required.

- \( \text{elim}_{1}(M_1, M_2) \models M_1 \ast M_2 \). Let \( S = (\mathfrak{U}, s, I, b) \) be a model of \( \text{elim}_{1}(M_1, M_2) \). We will find \( b_1 \) and \( b_2 \) such that \( b = b_1 \uplus b_2 \) and \((\mathfrak{U}, s, I, b_i) \models M_i \), for \( i = 1, 2 \). Since \( S \models \min_{M_i} + \min_{M_2} \leq ||b|| \land ||h|| \leq \max_{M_i} + \max_{M_2} - 1 \) by (5), we have, by Proposition 4.2:

\[
\min_{M_1} + \min_{M_2} \leq ||b|| < \max_{M_i}^S + \max_{M_2} - 1
\]
Let us now define the following sets, for \( i = 1, 2 \):

\[
L_i = \{ s(x) \in \text{dom}(b) \mid x \in \text{nv}(M_{3-i}) \setminus \text{av}(M_i) \}
\]

\[
Y_i = \{ x \in \text{Var} \mid s(x) \in L_i \}
\]

\[
A_i = \{ s(x) \mid x \in \text{av}(M_i) \}
\]

By Proposition 5.17, we have \( L_1 \cap L_2 = \emptyset \), \( L_1 \cap (A_1 \cup A_2) = \emptyset \), for \( i = 1, 2 \) and \( S \models \text{alloc}(Y_1) \wedge \text{alloc}(Y_2) \).

Moreover, because \((I, s, J, b) \models \eta_{12} \wedge \eta_{21}\), the following hold, for \( i = 1, 2 \):

\[
\|b\| \geq \|A_i\| + \|L_i\| + \min S_{M_{3-i}} \tag{8}
\]

\[
\|A_i\| + \|L_i\| < \max f \tag{9}
\]

We prove the following relation by distinguishing the cases below:

\[
\max(\min_{M_1} S, \|A_1\| + \|L_1\|) + \max(\min_{M_2} S, \|A_2\| + \|L_2\|) \leq \|b\| \tag{10}
\]

If \( \min_{M_1} S \geq \|A_1\| + \|L_1\| \) then we have \( \min_{M_1} S + \max(\min_{M_1} S, \|A_2\| + \|L_2\|) \leq \|b\| \) by (5) and (8). The case \( \min_{M_1} S \geq \|A_2\| + \|L_2\| \) is symmetric, and

(2) otherwise, if \( \min_{M_1} S < \|A_1\| + \|L_1\| \) and \( \min_{M_2} S < \|A_2\| + \|L_2\| \), because \( S \models \bigwedge x : \text{av}(M_1), y : \text{av}(M_2) \neg x \approx y \), the sets of locations \( L_1, L_2, A_1 \) and \( A_2 \) are pairwise disjoint and, since \( L_1 \cup L_2 \cup A_1 \cup A_2 \subseteq \text{dom}(b) \), it must be the case that \( \|b\| \geq \|A_1\| + \|L_1\| + \|A_2\| + \|L_2\| \).

Furthermore, we have \( \|b\| < \max_{M_1} S + \max_{M_2} S - 1 \) by (7) and one of the following cases holds (see Fig. 1):

(1) If \( \max_{M_1} S - 1 \leq \|b\| \) then \( \max_{M_1} S - \max(\|A_2\| + \|L_2\|, \min_{M_2} S) \) then let \( n_1 \overset{\text{def}}{=} \max_{M_1} S - \|A_1\| - \|L_1\| - 1 \) and \( n_2 \overset{\text{def}}{=} \|b\| - \max_{M_1} S - \|A_2\| - \|L_2\| + 1 \) (Fig. 1 (a)). We have that \( n_1 \geq 0 \) by (9) and \( n_2 \geq 0 \) by the hypothesis \( \max_{M_1} S - 1 \leq \|b\| - \max(\|A_2\| + \|L_2\|, \min_{M_2} S) \).

(2) Otherwise, let \( n_1 \overset{\text{def}}{=} \|b\| - \|A_1\| - \|L_1\| - \max(\|A_2\| + \|L_2\|, \min_{M_2} S) \) and \( n_2 \overset{\text{def}}{=} \|b\| - \|A_2\| - \|L_2\| \) (Fig. 1 (b)). We have \( n_1 \geq 0 \) by (10) and \( n_2 \geq 0 \) is immediate.

In both cases, the following holds, for \( i = 1, 2 \):

\[
\min_{M_i} S \leq \|A_i\| + \|L_i\| + n_i < \max_{M_i} S \tag{11}
\]

We have used the fact that \( \min_{M_i} S < \max_{M_i} S \) for \( i = 1, 2 \), which is a consequence of the fact that \( S \models \text{dc}(M_i)^u \), by (2) and Proposition 5.14.
Further, we have that \(|b| = \sum_{i=1,2} \left(\sum_{j} |A_j| + |L_i| + n_j\right)|. Moreover, there are exactly \(n_3 + n_2 \) locations in \(\text{dom}(b) \setminus (A_1 \cup L_1 \cup A_2 \cup L_2)\), thus we can partition this set into \(N_1\) and \(N_2\) such that \(|N_1| = n_3\) and define \(b_i\) to be the restriction of \(b\) to \(A_i \cup L_i \cup N_i\), for \(i = 1, 2\). It remains to be shown that \((\Gamma, s, I, b_i) \models M_i\), for \(i = 1, 2\). Below we do the proof for \(i = 1\), the case \(i = 2\) being similar.

Clearly, \((\Gamma, s, I, b_1) \models M_1^* \land M_1^0 \land M_2^0\), because \((\Gamma, s, I, b) \models M_1^* \land M_1^0 \land d(M_1)^u\), by Proposition 5.3. Further, by (11) and Proposition 4.2, we have \((\Gamma, s, I, b_1) \models |h| \geq \min M_i \land |h| < \max M_i\). There remains to show that \((\Gamma, s, I, b_1) \models M_1^0 \land M_2^0\).

\((M_2^0)\) Let \(\text{alloc}(x) \in M_2^0\) be a literal. Then \(x \in \text{av}(M_1)\), thus \(s(x) \in A_1\) and \((\Gamma, s, I, b_1) \models \text{alloc}(x)\) follows, by the definition of \(b_1\). Dually, let \(\neg \text{alloc}(x) \in M_1^0\) be a literal. Then, we have \(x \in \text{nv}(M_1)\). We distinguish the following cases:

- If \(x \in \text{av}(M_2)\) then \(s(x) \in A_2\) and since \(\text{dom}(b_1) \cap A_2 = \emptyset\), we have \(s(x) \notin \text{dom}(b_1)\), thus \((\Gamma, s, I, b_1) \models \neg \text{alloc}(x)\).

- Otherwise, \(x \in \text{nv}(M_1) \setminus \text{av}(M_2)\). Again, we distinguish the cases:
  * if \(x \in Y_2\) then \(s(x) \in L_2\) and because \(\text{dom}(b_1) \cap L_2 = \emptyset\), we obtain \(s(x) \notin \text{dom}(b_1)\), thus \((\Gamma, s, I, b_1) \models \neg \text{alloc}(x)\).
  * otherwise, \(x \notin Y_2\), thus \(s(x) \notin L_2\). But since \(x \in \text{nv}(M_1) \setminus \text{av}(M_2)\), by the definition of \(L_2\), it must be the case that \(s(x) \notin \text{dom}(b)\), thus \((\Gamma, s, I, b) \models \neg \text{alloc}(x)\) and \((\Gamma, s, I, b_1) \models \neg \text{alloc}(x)\) follows.

\((M_2^0)\) Let \(x \leftrightarrow y \in M_2^0\) be a literal. Then \(x \in \text{av}(M_1)\) and \(s(x) \in A_1\). Moreover, we have \(x \leftrightarrow y \in \text{fp}_a(M_1)\), thus \((\Gamma, s, I, b_1) \models x \leftrightarrow y\), by (3). Since \(b_1\) agrees on \(A_2\), we also have \((\Gamma, s, I, b_1) \models x \leftrightarrow y\). Dually, let \(\neg x \leftrightarrow y \in M_2^0\). If \(x \in \text{av}(M_1)\) then \(\neg x \leftrightarrow y \in \text{fp}_a(M_1)\), thus \((\Gamma, s, I, b_1) \models \neg x \leftrightarrow y\), since \(b_1\) agrees on \(A_1\). Otherwise, if \(x \notin \text{av}(M_2)\), we distinguish the cases:
  - if \(x \in \text{av}(M_2)\) then \(s(x) \in A_2\), and since \(\text{dom}(b_1) \cap A_2 = \emptyset\), we have \(s(x) \notin \text{dom}(b_1)\), thus \((\Gamma, s, I, b_1) \models \neg x \leftrightarrow y\).
  - otherwise, \(x \notin \text{av}(M_2)\), and since \(\text{atoms}(M_2^0) = \text{atoms}(M_2^0)\), we have \(\{x \leftrightarrow y, \neg x \leftrightarrow y\} \cap M_2 \neq \emptyset\). Since \(x \notin \text{av}(M_2)\), the only possibility is \(\neg x \leftrightarrow y \in M_2\), thus \(\neg x \leftrightarrow y \in \text{nto}(M_1, M_2)\) and \((\Gamma, s, I, b_1) \models \neg x \leftrightarrow y\), by (4). Since \(b_1\) is an extension of \(b_1\), we obtain that \((\Gamma, s, I, b_1) \models \neg x \leftrightarrow y\) as well.

We provide simple examples of application.

**Example 5.18.** Consider the following minterms:

\[ M_1 = E \cup \{|h| \geq 2, |h| < 4, x \leftrightarrow y, \text{alloc}(y), \neg y \leftrightarrow x, \neg z \leftrightarrow z\]  
\[ M_2 = E \cup \{|h| \geq 1, |h| < 2\]  
with \(E = \{\neg x \leftrightarrow y, \neg y \leftrightarrow z, \neg x \leftrightarrow z\}\). Then \(M_1 \land M_2 \equiv E \cup \{|h| \geq 3, |h| < 5, x \leftrightarrow y, \text{alloc}(y), \neg y \leftrightarrow x\}.

Let \(M_1^* = \{|h| \geq 0, |h| < 1, \neg x \leftrightarrow y\} \text{ and } M_2^* = \{|h| \geq 0, |h| < 3\} \equiv \{|h| \geq 0, |h| < \infty, \neg x \leftrightarrow \neg \text{alloc}(x)\}. \text{ Then } M_1^* \land M_2^* \equiv \{|h| \geq 0, |h| < \infty, \neg x \leftrightarrow \neg \text{alloc}(x)\}.\text{ Indeed, no model of } M_1^* \land M_2^* \text{ may allocate } x \text{ since the part of the heap that corresponds to } M_1^* \text{ is empty and } M_2^* \models \neg \text{alloc}(x).

**Remark 5.19.** Note that \(\text{elim}_x(M_1, M_2)\) contains negative occurrences of test formulæ \(\text{alloc}(x)\) that do not occur in \(M_1 \land M_2\). Such occurrences are introduced at Lines 4 and 6, due to the fact that we consider the closure of \(\neg \text{alloc}(x)\) formulæ w.r.t. all the equalities in \(M_1, M_2\). For example, if \(M_1 = \{\neg \text{alloc}(x), x \approx y, |h| \geq 0, |h| < \infty\} \text{ and } M_2 = \{|h| \geq 0, |h| < \infty\}, \text{ then } y \in \text{nv}(M_1) \text{ and alloc}(y) \text{ occurs at negative polarity in } \text{elim}_x(M_1, M_2)\). This is problematic because underlying results depend on the fact that the polarity of alloc(x) formulæ is preserved (Lemma 5.29). However, if alloc(x) occurs at a negative polarity in elim_x(M_1, M_2), then there exists a literal \(-\text{alloc}(x') \in M_1 \land M_2\), such that \(\text{elim}_x(M_1, M_2) \models x \approx x', \text{ making the negative occurrence of alloc(x) actually redundant. Consequently, equivalence is preserved when only the test formulæ alloc(x) such that } -\text{alloc}(x) \in M_1 \land M_2 \text{ occur at negative polarity in} \).
Next, we show a similar result for the separating implication. For technical convenience, we translate the separation \( M_1 \to M_2 \), instead of \( M_1 \not\to M_2 \), as an equivalent boolean combination of test formulæ. This is without loss of generality, because \( M_1 \not\to M_2 \equiv \neg(M_1 \to \neg M_2) \). Unlike with the case of the separating conjunction (Lemma 5.16), here the definition of the boolean combination of test formulæ depends on whether the universe is finite or infinite.

If the complement of some literal \( \ell \in \text{fp}_n(M_1) \) belongs to \( M_2 \) then no extension by a heap that satisfies \( \ell \) may satisfy \( \tilde{\ell} \). Therefore, as an additional simplifying assumption, we suppose that \( \text{fp}_n(M_1) \cap M_2 = \emptyset \), so that \( M_1 \not\to M_2 \) is not trivially unsatisfiable.

**Lemma 5.20.** Let \( M_1 \) and \( M_2 \) be footprint-consistent minterms that are E-complete for \( \text{Var}(M_1 \cup M_2) \), such that:
(a) \( M_1 \) is E-complete for \( \text{Var}(M_1 \cup M_2) \), (b) atoms\( (M_1^c \cup M_2^c) \subseteq \text{atoms}(M_1^\ell \cup M_2^\ell) \), and (c) \( \text{fp}_n(M_1) \cap M_2 = \emptyset \). Then, \( M_1 \not\to M_2 \equiv \text{elim}_{\ell}(M_1, M_2) \) and \( M_1 \to M_2 \equiv_{\text{inf}} \text{elim}_{\ell}(M_1, M_2) \), where:

\[
\text{elim}_{\ell}(M_1, M_2) \overset{\text{def}}{=} \text{pc}(M_1)^e \land M_1^f \land M_2^f \land \text{dc}(M_1)^u \land \text{dc}(M_2)^u \land \\
\text{nalloc}(\text{av}(M_1)) \land \text{fp}_{\text{nv}}(M_1) (M_2) \land \\
|h| \geq \min_{M_1} - \max_{M_1} + 1 \land |h| < \max_{M_2} - \min_{M_1} \land \\
\lambda^{\ell} \land \lambda^{\ell} \land Y \subseteq \text{Var}(M_1 \cup M_2) 
\]

with \( \lambda^{\ell} \overset{\text{def}}{=} \land_{Y \subseteq \text{Var}(M_1 \cup M_2)} \text{nalloc}(Y) \rightarrow \left( |h| < |U| - \min_{M_1} - \#_n(Y, M_1) + 1 \land \\
|U| \geq \min_{M_1} + \#_n(Y, M_1) \right) \). and \( \lambda^{\ell} \equiv \top \).

Intuitively, a heap satisfies \( M_1 \not\to M_2 \) if it has an extension, by a disjoint heap satisfying \( M_1 \), that satisfies \( M_2 \). Thus, \( \text{elim}_{\ell}(M_1, M_2) \) must entail the heap-independent literals of both \( M_1 \) and \( M_2 \) (12). Next, no variable allocated by \( M_1 \) must be allocated by \( \text{elim}_{\ell}(M_1, M_2) \), otherwise no extension by a heap satisfying \( M_1 \) is possible and, moreover, the footprint of \( M_2 \) relative to the unallocated variables of \( M_1 \) must be asserted (13). The heap’s cardinality constraints depend on the bounds of \( M_1 \) and \( M_2 \) (14) and, if \( Y \) is a set of variables not allocated in the heap, these variables can be allocated in the extension (15). Actually, this is where the finite universe assumption first comes into play. If the universe is infinite, then there are enough locations outside the heap to be assigned to \( Y \). However, if the universe is finite, then it is necessary to ensure that there are at least \( \#_n(Y, M_1) \) free locations to be assigned to \( Y \) (15). We now give the proof of Lemma 5.20.

**Proof.** If \( \text{pc}(M_1) = \bot \) then \( M_1 \not\to M_2 \equiv \text{elim}_{\ell}(M_1, M_2) \equiv \bot \). Also, since \( M_1 \) and \( M_2 \) are E-complete for \( \text{Var}(M_1 \cup M_2) \), if we suppose that \( M_1^p \neq M_2^p \) then \( M_1 \not\to M_2 \equiv \text{elim}_{\ell}(M_1, M_2) \equiv \bot \). From now on, we will assume that \( \text{pc}(M_1) = M_1 \) and \( M_1^p = M_2^p \).

\( M_1 \not\to M_2 \models \text{elim}_{\ell}(M_1, M_2) \). Let \( S = (\mathcal{U}, s, I, h) \) be a structure such that \( S \models M_1 \not\to M_2 \). Then there exists a heap \( h' \) disjoint from \( h \) such that \( (\mathcal{U}, s, I, h') \models M_1 \) and \( (\mathcal{U}, s, I, h \uplus h') \models M_2 \). Below we prove that \( S \) is a model of the formulæ (12), (13), (14) and (15), respectively.

(12) We have \( (\mathcal{U}, s, I, h') \models M_1^p \land M_2^c \land M_1^p \land M_2^c \) by Proposition 5.3, and by Proposition 5.15, we deduce that \( (\mathcal{U}, s, I, h) \models \text{pc}(M_1)^e \land \text{dc}(M_1)^u \land M_1^f \). Analogously, \( (\mathcal{U}, s, I, h) \models M_2^c \land \text{dc}(M_2)^u \land M_2^f \) follows from \( (\mathcal{U}, s, I, h \uplus h') \models M_2 \) by Propositions 5.3 and 5.15.

(13) Since \( (\mathcal{U}, s, I, h') \models M_1 \), also \( (\mathcal{U}, s, I, h') \models \text{alloc}(\text{av}(M_1)) \) and since dom\( (h') \cap \text{dom}(h) = \emptyset \), we have \( (\mathcal{U}, s, I, h) \models \text{nalloc}(\text{av}(M_1)) \). To prove that \( (\mathcal{U}, s, I, h) \models \text{fp}_{\text{nv}}(M_1) (M_2) \), we consider four cases, depending on the form of the literal:
If alloc(x) ∈ M₂ and x ∈ nv(M₁), then s(x) ∈ dom(b) ∪ dom(b’) and s(x) /∈ dom(b’), thus s(x) ∈ dom(b) and ⟨U, s, I, h⟩ |= alloc(x), by Proposition 4.2.

Finally, let V denote the set of images the allocated variable x is assigned in M₁. Thus, |s(Y ∩ nv(M₈))| ≥ |b| + min_S(M₁), since |s(Y ∩ nv(M₈))| = |Y ∩ nv(M₈)|, by Proposition 5.5, since M₁ is E-complete. Therefore, |b| ≤ |I| − min_S(M₁) ∧ #n(Y, M₁).

• elim_(M₁, M₂) |= M₁ → M₂. Let S = ⟨U, s, I, h⟩ be a structure such that S |= elim_(M₁, M₂). We build a heap b’ such that dom(b’) ∧ dom(b) = ∅, and ⟨U, s, I, h⟩ |= M₁ and ⟨U, s, I, h, b’⟩ |= M₂. First, for each variable x ∈ av(M₁) such that x’ ↦ y ∈ M₁ for some variable x’ with x ≈_M₁ x’, we add the tuple (s(x), s(y)) to b’. Since ⟨U, s, I, h⟩ |= pc(M₂), for any pair of variables x ≈_M₁ x’ if x ↦ y, x’ ↦ y’ ∈ M₁ then y’ ≈_M₁ y’, and the result is a functional relation. We define:

\[
A = \{ x ∈ av(M₁) \mid ∀x’(∀y : x ≈_M₁ x’ → x’ ↦ y ∈ M₁) \}
\]

\[
V_x = \{ \langle s(y_1), \ldots, s(y_{|k|}) \rangle \in U^{|k|} \mid x ≈_M₁ x’ \wedge x’ ↦ y ∈ M₁ \}
\]

\[
N = \{ x ∈ Var(M₁ ∪ M₂) \mid s(x) /∈ dom(b) \}
\]

Intuitively, A denotes the set of variables that must be allocated but with no constraint on their image; this set is independent of the interpretation under consideration. The set Vₓ denotes the set of images the allocated variable x cannot point to, and N denotes the set of variables that are not allocated in h.

For each x ∈ A we choose a tuple (ℓ₁, ..., ℓ_k) ∈ U^{|k|} \ V_x and let b’(s(x)) = (ℓ₁, ..., ℓ_k). Since M₁ is E-complete, we have |V_x| ≤ δ₅(M₁) for each x ∈ A, and such a choice is possible because ⟨U, s, I, h⟩ |= pc(M₁), thus |U^{|k|}| ≥ δ₅(M₁) + 1.

Since ⟨U, s, I, h⟩ |= nalloc(N), if |U| is finite, by (14) it must be the case that:

|b| ≤ |I| − min_S(M₁) ≤ #n(M₁) + 1

|I| ≥ min_S(M₂) + #n(N, M₂)

Finally, let L ⊆ U \ (dom(h) ∪ s(av(M₁)) ∪ s(nv(M₁))) be a finite set of locations of cardinality |L| = max(min_S(M₁), min_S(M₂) − |b|) − #n(M₁). Choosing such a set L is possible, because either |U| is infinite, or |U| is finite, in which case:

|I| ≥ max(min_S(M₁), |b|, min_S(M₂) + #n(N, M₁)), by (16) and (17)

≥ max(min_S(M₁), min_S(M₂) − |b|) − #n(M₁) + |b| + #n(M₁) + #n(N, M₁)

= |L| + |b| + #n(M₁) + #n(N, M₂)

≥ |L| + |dom(h) ∪ s(av(M₁)) ∪ s(nv(M₁))|
where the last inequality is a consequence of Proposition 5.5. We choose an arbitrary tuple \((\ell_1, \ldots, \ell_k) \in \mathcal{U}^k\)
and let \(b'(\ell) = (\ell_1, \ldots, \ell_k)\) for all \(\ell \in L\). Because \(\mathcal{U}\) is non-empty, such a tuple exists. Consequently, we have \(\text{dom}(b') = s(\text{av}(M_1)) \cup L\) and \(\text{dom}(b') \cap \text{dom}(b) = \emptyset\) because \(s(\text{av}(M_1)) \cap \text{dom}(b) = \emptyset\) by (13) and \(L \cap \text{dom}(b) = \emptyset\) by construction. We now prove:

- \((\mathcal{U}, s, \mathcal{T}, b') \models M_1\). Clearly \((\mathcal{U}, s, \mathcal{T}, b') \models M_1^\alpha\), observe that \(s(x) \in \text{dom}(b')\) for each \(x \in \text{av}(M_1)\), hence for each literal \(\text{alloc}(x) \in M_1\) we have
  \((\mathcal{U}, s, \mathcal{T}, b') \models \text{alloc}(x)\). Moreover, we have \(\text{dom}(b') \cap s(\text{av}(M_1)) = s(\text{av}(M_1)) \cup L\) and \(\text{dom}(b') \cap \text{av}(M_1) = \emptyset\), because \(M_1\) is footnrt and E-complete for \(\text{Var}(M_1 \cup M_2)\), by Proposition 5.11.

Thus \((\mathcal{U}, s, \mathcal{T}, b') \models \neg\text{alloc}(x)\) for each literal \(\text{alloc}(x) \in M_1^\alpha\).
For each literal \(x \leftrightarrow y\) in \(M_1\), we distinguish two cases.

* If \(x \in \text{av}(M_1)\), then \((s(y_1), \ldots, s(y_k)) \in V_x\) hence \(b'(s(x)) \neq (s(y_1), \ldots, s(y_k))\) by construction.

* If \(x \not\in \text{av}(M_1)\), then since \(M_1\) is A-complete for \(\text{Var}(M_1 \cup M_2)\), we have \(x \in \text{av}(M_1)\), thus \(s(x) \not\in \text{dom}(b') = s(\text{av}(M_1)) \cup L\).

We finally prove that \((\mathcal{U}, s, \mathcal{T}, b') \models |h| \geq \min_{M_1} \land |h| < \max_{M_1}\). Since \(\text{dom}(b') = s(\text{av}(M_1)) \cup L\) and \(s(\text{av}(M_1)) \cap L = \emptyset\), we have \(|b'| = |s(\text{av}(M_1))| + |L| = \max(\min_{M_1}^S, \min_{M_1}^S, |\{h\}|)\).
If \(|b'| = \min_{M_1}^S\) then \(|b'| = \min_{M_1}^S\) because \(S \models \text{dc}(M_1)^a\), which implies that \(\min_{M_1}^S < \max_{M_1}\), by Proposition 5.14. Otherwise \(|b'| = \min_{M_1}^S - |\{h\}| < \min_{M_1}^S - 1\), and we have thus \(|b'| < \min_{M_1}^S - 1\).

- \((\mathcal{U}, s, \mathcal{T}, b' \cup b) \models M_2\). We have \((\mathcal{U}, s, \mathcal{T}, b \cup b) \models M_2^\alpha\) because \((\mathcal{U}, s, \mathcal{T}, b) \models M_2^\alpha\) and these formul\ae\ do not depend on the heap. Next, for a given variable \(x\), let \(\alpha_x \in \{\text{alloc}(x), \neg\text{alloc}(x), x \leftrightarrow y, \neg x \leftrightarrow y\} \cap \text{M}_2\) be a literal and let \(\overline{\alpha_x}\) denote its complement. If \(x \in \text{av}(M_2)\) then

- \(x \in \text{av}(M_2)\) and \((\mathcal{U}, s, \mathcal{T}, b) \models \neg\alpha_x\) by (13). Moreover, because \(b\) and \(b'\) agree on \(s(\text{av}(M_1))\), we obtain \((\mathcal{U}, s, \mathcal{T}, \mathcal{U} \cup b') \models \neg\alpha_x\). Otherwise \(x \not\in \text{av}(M_2)\) hence \(x \in \text{av}(M_1)\) because \(M_1\) is A-complete for \(\text{Var}(M_1 \cup M_2)\), and since \(\alpha_x \in M_2^\alpha \cup M_2^\alpha\) and \(\text{atoms}(M_2^\alpha \cup M_2^\alpha) \subseteq \text{atoms}(M_2^\alpha \cup M_2^\alpha)\), we have \(\alpha_x \in \text{fp}_a(M_1)\), because the case \(\overline{\alpha_x} \in \text{fp}_a(M_1)\) is in contradiction with \(\text{fp}_a(M_1) \cap \overline{M_2} = \emptyset\) (condition (c) of the Lemma). But then \((\mathcal{U}, s, \mathcal{T}, b') \models \alpha_x\) and \((\mathcal{U}, s, \mathcal{T}, b \cup b') \models \alpha_x\) follows by Proposition 5.7.

We have thus proved that \((\mathcal{U}, s, \mathcal{T}, b' \cup b') \models (M_2^\alpha \cup M_2^\alpha)\). We are left with proving that \(\min_{M_2}^S = |b| + |b'| = \max(\min_{M_2}^f, |b|)\), hence \(\min_{M_2}^S < \max_{M_2}\). If \(\min_{M_2}^S < |b|\), the result follows from the fact that \(S \models \text{dc}(M_2)^a\), which implies \(\min_{M_2}^S < \max_{M_2}\), by Proposition 5.14. Otherwise, \(|b| + |b'| = \min_{M_2}^S + |b| > \min_{M_2}^S\), and \(|b| + |b'| < \max_{M_2}\) follows from (14).

Example 5.21. Let \(M_1 = \{\text{alloc}(x), \neg\text{alloc}(y), \neg x \leftrightarrow y\} \models (|h| \geq 1, |h| < 2), M_2 = \{\neg x \approx y, |h| \geq 3, |h| < \infty, \neg x \leftrightarrow y\} \models (|h| \geq 2, |h| < \infty, \neg\text{alloc}(x), \neg y \leftrightarrow y)\).

5.3 Translating Quantifier-free SL\(k\) into Minterms

We prove next that each quantifier-free SL\(k\) formula is equivalent to a finite disjunction of minterms. Intuitively, these disjunctions are defined by induction on the structure of the formula. The base cases and classical connectives are easy to handle. For formul\ae\ \(\psi_1 \lor \psi_2\) or \(\psi_1 \rightarrow \psi_2\), the transformation is first applied on \(\psi_1\) and \(\psi_2\), then the following equivalences are used to shift \(\ast\) and \(\rightarrow\) innermost in the formul\ae:\n
\[
\begin{align*}
(\phi_1 \lor \phi_2) \lor \phi & \equiv (\phi_1 \lor \phi) \lor (\phi_2 \lor \phi) \\
(\phi_1 \lor \phi_2) \land \phi & \equiv (\phi_1 \land \phi) \lor (\phi_2 \land \phi) \\
\phi \lor (\phi_1 \lor \phi_2) & \equiv (\phi \lor \phi_1) \lor (\phi \lor \phi_2) \\
\phi \land (\phi_1 \lor \phi_2) & \equiv (\phi \land \phi_1) \lor (\phi \land \phi_2) \\
\phi \lor \phi_1 & \equiv \phi_1 \\
\phi \land \phi_1 & \equiv \phi_1 \\
\phi \lor \phi_2 & \equiv \phi_2 \\
\phi \land \phi_2 & \equiv \phi_2 \\
\phi \lor (\phi_1 \lor \phi_2) & \equiv \phi_1 \lor (\phi_1 \lor \phi_2) \\
\phi \land (\phi_1 \lor \phi_2) & \equiv (\phi_1 \land \phi) \lor (\phi_2 \land \phi) \\
\phi \lor (\phi_1 \land \phi_2) & \equiv (\phi_1 \lor \phi) \land (\phi_2 \lor \phi) \\
\phi \land (\phi_1 \land \phi_2) & \equiv (\phi_1 \land \phi) \lor (\phi_2 \land \phi)
\end{align*}
\]

Afterwards, the operands of $\ast$ and $\rightarrow$ are minterms, and the result is obtained using the equivalences from Lemmas 5.16 and 5.20, respectively (up to a transformation into disjunctive normal form). The only difficulty is that these lemmas impose some additional conditions on the minterms (e.g., being E-complete, or A-complete).

However, the conditions are easy to enforce by case splitting, as illustrated by Example 5.22.

Example 5.22. Consider the formula $x \mapsto x \rightarrow y \mapsto y$. It is easy to check that $x \mapsto x \equiv M_1$, where $M_1 = x \leftrightarrow x \land |h| \geq 1 \land |h| < 2$ and $y \mapsto y \equiv M_2$, where $M_2 = y \leftrightarrow y \land |h| \geq 1 \land |h| < 2$. To apply Lemma 5.20, we need to ensure that $M_1$ and $M_2$ are E-complete, which may be done by adding either $x \equiv y$ or $x \not\equiv y$ to each minterm. We also have to ensure that $M_1$ is A-complete, thus for $z \in \{x, y\}$, we add either alloc($z$) or $\neg$alloc($z$) to $M_1$. Finally, we must have atoms($M_1^\ast$ $\cup$ $M_2^\ast$) $\subseteq$ atoms($M_1^\ast$ $\cup$ $M_2^\ast$), thus we add either $y \mapsto y$ or $\neg y \mapsto y$ to $M_1$.

After removing redundancies, we get (among others) the minterms: $M'_1 = x \leftrightarrow x \land |h| \geq 1 \land |h| < 2 \land x = y$ and $M'_2 = y \leftrightarrow y \land |h| \geq 1 \land |h| < 2 \land x \equiv y$. Afterwards we compute elim$_{\omega}$($M'_1, M'_2$) $= x \leftrightarrow y \land \neg$alloc($x$) $\land |h| \geq 1 \land |h| < 1$.

To describe the transformation in a more formal way, we first need to show that the conjunction of two minterms can be written as a disjunction of minterms. To this aim, given minterms $M_1$ and $M_2$, we define the sets of constraints $\text{min}(M_1, M_2)$ and $\text{max}(M_1, M_2)$ by taking the conjunction of the lower and upper bounds on the cardinality of the heap and keeping the most restrictive bounds.

**Definition 5.23.**

\[
\begin{align*}
\text{min}(M_1, M_2) & \overset{\text{def}}{=} \{ |h| \geq \max(\min_M, \min_{M_2}) \} \\
\text{if} \ \min_M, \min_{M_2} \in \mathbb{N} \\
\{ |h| \geq \min_M \land |U| < \min_M + m + 1 \} & \text{if} \ \min_M \in \mathbb{N}, \min_{M_2} = |U| - m, \ i = 1, 2 \\
\{ |h| \geq \max_M \land |U| \geq \max_M + m + 1 \} & \text{if} \ \max_M \in \mathbb{N}, \min_{M_2} = |U| - m, \ i = 1, 2 \\
\{ |h| \geq |U| - \min(m_1, m_2) \} & \text{if} \ \min_M = |U| - m, \ i = 1, 2 \\
\end{align*}
\]

\[
\begin{align*}
\text{max}(M_1, M_2) & \overset{\text{def}}{=} \{ |h| < \min(\max_M, \max_{M_2}) \} \\
\text{if} \ \max_M, \max_{M_2} \in \mathbb{N} \cup \mathbb{R} \\
\{ |h| < \max_M \} & \text{if} \ \max_{M_2} = \infty, \max_M = |U| - m, \ i = 1, 2 \\
\{ |h| < \max_M \land |U| \geq \max_M + m \} & \text{if} \ \max_{M_2} \in \mathbb{N}, \max_M = |U| - m, \ i = 1, 2 \\
\{ |h| < |U| - \max(m_1, m_2) \} & \text{if} \ \max_M = |U| - m, \ i = 1, 2 \\
\end{align*}
\]

For instance, if $M_1 = \{ |h| \geq 2, |h| < |U| - 1 \}$ and $M_2 = \{ |h| \geq 3, |h| < |U| - 2 \}$, then $\text{min}(M_1, M_2) = \{ |h| \geq 3 \}$ and $\text{max}(M_1, M_2) = \{ |h| < |U| - 2 \}$. Heterogeneous constraints are merged by performing a case split on the value of $|U|$. For example, if $M_1 = \{ |h| \geq |U| - 4 \}$ and $M_2 = \{ |h| \geq 1 \}$, then the first condition prevails if $|U| \geq 5$ yielding: $\text{min}(M_1, M_2) = \{ |h| \geq 1 \land |U| < 5 \land \text{card}(M_1, M_2) \}$. The disjunction of minterms equivalent to a conjunction of two minterms is then defined as follows:

**Definition 5.24.** For any minterms $M_1, M_2$, let $[M_1, M_2] \overset{\text{def}}{=} \{ \bigwedge_{i=1,2} M_i^\ast \land M_i^f \land M_i^2 \land M_i^p \land M_i \land \mu \land \forall \ v \mid \mu \in \text{min}(M_1, M_2), \ v \in \text{max}(M_1, M_2) \}$. We extend this notation recursively to any set of minterms of size $n > 2$:

$[M_1, M_2, \ldots, M_n] \overset{\text{def}}{=} \bigcup_{M \in [M_1, \ldots, M_{n-1}]} [M, M_n]$.

**Proposition 5.25.** Given minterms $M_1, \ldots, M_n$, we have $\land_{i=1}^n M_i \equiv \bigvee_{M \in [M_1, \ldots, M_n]} M$.

**Proof.** We prove the result for $n = 2$, the general result follows by induction. For $n = 2$, this is a consequence of the fact that $|h| \geq \min_M \land |h| \geq \min_M \equiv \bigvee_{v \in \text{min}(M_1, M_2)}$, and $|h| \leq \max_M \land |h| \leq \max_M \equiv \bigvee_{v \in \text{max}(M_1, M_2)}$. We prove the first fact in the case where $\min_M = m_1$ and $\min_M = |U| - m_2$, the other cases are similar. Consider a structure $S = (U, s, I, b)$ such that $S \models |h| \geq m_1 \land |h| \geq |U| - m_2$. Then $|h| \geq m_1$ and $|h| \geq |U| - m_2$, and we distinguish two cases.
• if \( m_1 \geq |U| - m_2 \), then necessarily \(|\| < m_1 + m_2 + 1\), so that \( S \models |h| \geq m_1 \land |U| < m_1 + m_2 + 1 \).
• otherwise, we have \(|U| \geq m_1 + m_2 + 1\), so that \( S \models |h| \geq |U| - m_2 \land |U| \geq m_1 + m_2 + 1 \).

Conversely, if \( S \) is a structure such that either \( S \models |h| \geq m_1 \land |U| < m_1 + m_2 + 1 \) or \( S \models |h| \geq |U| - m_2 \land |U| \geq m_1 + m_2 + 1 \), then it is straightforward to verify that \( S \models |h| \geq m_1 \land |h| \geq |U| - m_2 \).

The following proposition states some properties of the literals occurring in \([M_1, \ldots, M_n]\).

**Proposition 5.26.** Given minterms \( M_1, \ldots, M_n \) and \( M \in [M_1, \ldots, M_n] \), if \( \ell \in M \) is a literal then either \( \ell \in M_i \), for some \( i = 1, \ldots, n \), or \( \ell \in \{ |U| \geq m_1 + m_2, |U| < m_1 + m_2, |U| \geq m_1 + m_2 + 1, |U| < m_1 + m_2 + 1 \} \), where \( M_1 \cup \cdots \cup M_n \) contains two literals \( \ell_i \in \{ |h| \geq m_1, |h| < m_1, |h| \geq |U| - m_1, |h| < |U| - m_1 \} \), for \( i = 1, 2 \).

**Proof.** Assume that \( n = 2 \). If \( \ell \notin M_1 \cup M_2 \) then by definition of \([M_1, M_2]\), necessarily \( \ell \) occurs in \( \operatorname{minh}(M_1, M_2) \cup \operatorname{maxh}(M_1, M_2) \) and the proof is immediate, by definition of these sets. The proof for \( n \geq 2 \) goes by induction on \( n \).

For two sets \( K, L \) of literals, a *completion* of \( K \) w.r.t. \( L \) is a set of literals \( K' \) that is minimal with respect to inclusion of sets, such that \( K \subseteq K' \) and \( \operatorname{atoms}(L) \subseteq \operatorname{atoms}(K) \) (i.e., \( K \subseteq K' \) and for every \( \ell \in L, K' \) contains either \( \ell \) or \( \bar{\ell} \)). We denote by \( (K)^L \) the set of completions of \( K \) w.r.t. \( L \).

**Proposition 5.27.** If \( K \) and \( L \) are sets of literals, then \( K \equiv \bigvee_{\psi \in (K)^L} \psi \). If further \( K \) is a minterm and \( L \) contains no literals of the form \( |h| \geq t \) or \( |h| < t \), then every set \( P \in (K)^L \) is a minterm such that \( \operatorname{Var}(P) = \operatorname{Var}(K) \cup \operatorname{Var}(L) \), \( \operatorname{min}_P = \min_K \) and \( \operatorname{max}_P = \max_K \).

**Proof.** Immediate, by the definition of \( (K)^L \).

For a literal \( \ell \), let \( [\ell]^m \) be an equivalent minterm obtained from \( \ell \) by adding the missing lower/upper bounds on the cardinality of the heap, namely \(|h| \geq 0 \) if \( \ell \notin \{ |h| \geq n, |h| \geq |U| - n \mid n \in \mathbb{Z} \} \) and \(|h| < \infty \) if \( \ell \notin \{ |h| < n, |h| < |U| - n \mid n \in \mathbb{Z} \} \). We extend this notation to sets (i.e., conjunctions) of literals as \([\ell_1, \ldots, \ell_n]^m \equiv ([\ell_1]^m \ldots [\ell_n]^m)\). We have \( \ell \equiv [\ell]^m \) for any literal \( \ell \) and \( L \equiv \bigvee_{\ell \in [L]^m} M \), for any set \( L \) of literals. For a boolean combination of literals \( \phi \), we denote by \( (\phi)^m \) its disjunctive normal form. We assume from now on that the disjunctive normal form of a formula is canonical and all the conjunctions are incomparable with respect to logical entailment.

Given a formula \( \phi \) in disjunctive normal form \( \phi = \bigvee_{i=1}^n C_i \), where \( C_1, \ldots, C_n \) are conjunctions (represented by sets) of literals, we define \( [\phi]^m \equiv \bigcup_{i=1}^n [C_i]^m \). We have \( [\phi]^m = \bigvee_{M \in [\phi]^m} M \). Further, let \( E(L) \equiv \{ x \approx y \mid x, y \in \operatorname{Var}(L) \} \) and \( A(L) \equiv \{ \text{alloc}(x) \mid x \in \operatorname{Var}(L) \} \), for a set \( L \) of literals.

For each \( \ell \in \{ \text{fin}, \text{inf} \} \), we define the set of minterms \( \mu^{\ell}(\phi) \) recursively on the structure of \( \phi \):

\[
\begin{align*}
\mu^{\ell}(\text{emp}) & \equiv |h| \approx 0 \quad \mu^{\ell}(x \rightarrow y) & \equiv \{ x \rightarrow y \land |h| \approx 1 \} \\
\mu^{\ell}(x \approx y) & \equiv \{ x \approx y \land |h| \geq 0 \land |h| < \infty \} \quad \mu^{\ell}(q(x_1, \ldots, x_{\ell(q)})) & \equiv \{ q(x_1, \ldots, x_{\ell(q)}) \land |h| \geq 0 \land |h| < \infty \}
\end{align*}
\]
\[ \mu^i(\phi_1 \land \phi_2) \overset{df}{=} \bigcup_{M_i \in \mu^i(\phi_1)} [M_1, M_2] \]

\[ \mu^i(\neg \phi_1) \overset{df}{=} \bigcup_{M_i \in \mu^i(\phi_1)} \left\{ \left[ \ell_1, \ldots, \ell_n \right] \mid \ell_i \in M_i, \; i \in [1,n] \right\}, \text{ where } \mu^i(\phi_1) = \{ M_1, \ldots, M_n \} \]

\[ \mu^i(\phi_1 \lor \phi_2) \overset{df}{=} \bigcup_{M_i \in \mu^i(\phi_1)} \left\{ \left[ \begin{array}{c} \text{elim} \circ (P_1, P_2) \end{array} \right] \right\} \bigcup_{M_j \in \mu^i(\phi_2)} \left\{ \left[ \begin{array}{c} \text{elim} \circ (Q_1, Q_2) \end{array} \right] \right\} \bigcup_{N_j \in (M_j)^{E(M_j \cup M_k)}} P_j \in (N_j)^{N_{j-1}}, \; j = 1, 2 \]

\[ \mu^i(\phi_1 \Rightarrow \phi_2) \overset{df}{=} \bigcup_{M_i \in \mu^i(\phi_1)} \left\{ \left[ \begin{array}{c} \text{elim} \circ (Q_1, Q_2) \end{array} \right] \right\} \bigcup_{N_j \in (M_j)^{E(M_j \cup M_k)}} P_1 \in (N_1)^{A_{M_k} \cup M_k}, Q_1 \in (P_1)^{E_{M_k}}, \; j = 1, 2 \]

Intuitively, \( \mu^i(\phi_1 \lor \phi_2) \) and \( \mu^i(\phi_1 \Rightarrow \phi_2) \) are obtained by first recursively computing \( \mu^i(\phi_1) \) and \( \mu^i(\phi_2) \), then extending the obtained minterms in such a way that the hypotheses of Lemmas 5.16 or 5.20 are satisfied, and finally applying \( \text{elim}_i \) and \( \text{elim}_\omega \), respectively.

**Lemma 5.28.** Given a quantifier-free SL\(^k\) formula \( \phi \), the following equivalences hold: (1) \( \phi \equiv^\inf \bigvee_{M \in \mu^{m^i}(\phi)} M \), and (2) \( \phi \equiv^\inf \bigvee_{M \in \mu^{m^i}(\phi)} M \).

**Proof.** We show that \( \phi \equiv^\inf \bigvee_{M \in \mu^{m^i}(\phi)} M \) by induction on the structure of \( \phi \). The fact that \( \phi \equiv^\inf \bigvee_{M \in \mu^{m^i}(\phi)} M \) is proved in the same way. The base cases are immediate and the inductive cases are dealt with below:

- If \( \phi = \phi_1 \land \phi_2 \) and \( \phi_1 \equiv^\inf \bigvee_{M \in \mu^{m^i}(\phi_1)} M_1 \) for \( i = 1, 2 \) by the inductive hypothesis and Proposition 5.25, we have:
  \[ \phi = \bigvee_{M \in \mu^{m^i}(\phi_1)} M_1 \land \bigvee_{M \in \mu^{m^i}(\phi_2)} M_2 \]

- If \( \phi = \neg \phi_1 \) and \( \mu^{m^i}(\phi_1) = \{ M_1, \ldots, M_n \} \), \( M_i = \{ \ell_{i1}, \ldots, \ell_{in_i} \} \) for all \( i \in [1,n] \), then since \( \phi_1 \equiv^\inf \bigwedge_{j=1}^n \ell_{ij} \) by the inductive hypothesis, we have:
  \[ \neg \phi_1 \equiv^\inf \bigwedge_{i=1}^n \bigvee_{j=1}^{n_i} \ell_{ij} \]

- If \( \phi = \phi_1 \lor \phi_2 \) and \( \phi_1 \equiv^\inf \bigvee_{M \in \mu^{m^i}(\phi_1)} M \) for \( i = 1, 2 \) by the induction hypothesis, we compute successively:
  \[ (\phi_1 \lor \phi_2) \overset{\text{distributivity of } \lor \text{ with } \bigvee}{=} \bigvee_{M \in \mu^{m^i}(\phi_1), \; i=1,2} \bigvee_{M \in \mu^{m^i}(\phi_2)} \]
  \[ = \bigvee_{M \in \mu^{m^i}(\phi_1), \; i=1,2} \bigvee_{M \in \mu^{m^i}(\phi_2)} \bigvee_{N \in (M_i)^{E(M_i \cup M_k)}} N \neq^* P_i \]
  \[ = \bigvee_{M \in \mu^{m^i}(\phi_1), \; i=1,2} \bigvee_{M \in \mu^{m^i}(\phi_2)} \bigvee_{N \in (N_1)^{E(N_1)}} P_1 \neq^* P_2 \]
  \[ = \bigvee_{M \in \mu^{m^i}(\phi_1), \; i=1,2} \bigvee_{M \in \mu^{m^i}(\phi_2)} \bigvee_{N \in (N_1)^{E(N_1)}} P_1 \neq^* P_2 \]

\[ \overset{\text{See Definition 5.2 for the definition of } N^P.}{
At this point, observe that $N_i$, and thus $P_i$, are E-complete for $\text{Var}(M_1 \cup M_2)$, for $i = 1, 2$. Moreover, $\text{atoms}(P_1') = \text{atoms}(P_2')$, because $P_i \in (N_i)^{\mu_{P_i}}$, for $i = 1, 2$. We can thus apply Lemma 5.16 and infer that:

$$P_1 * P_2 \equiv (\text{elim}_{\phi}(P_1, P_2))^{\text{def}} \equiv \bigvee_{M \in [\text{elim}_{\phi}(P_1, P_2)]^{\text{def}}} M$$

• If $\phi = \phi_1 \rightarrow \phi_2$ and $\phi_2 \equiv \phi_2 \vee M \in \mu_{\phi_1}(\phi_1)$, $M$, $i = 1, 2$, by the induction hypothesis, we compute, successively:

$$\phi_1 \rightarrow \phi_2 \equiv \phi_1 \rightarrow \phi_2 \equiv (\phi_1 \rightarrow \phi_2) \text{ [distributivity of } \rightarrow \text{ with } \vee ]$$

$$\equiv \phi_1 \rightarrow (\phi_1 \rightarrow \phi_2)$$

$$\equiv \phi_1 \rightarrow (\phi_2 \rightarrow \phi_2) \equiv \phi_1 \rightarrow \phi_2$$

Observe that $N_i$ and thus $P_i$ are E-complete for $\text{Var}(M_1 \cup M_2)$, for $i = 1, 2$. Moreover, $P_1$ is A-complete for $\text{Var}(M_1 \cup M_2)$, because $P_1 \in (N_1)^{\lambda_{M_1} \cup \lambda_{M_2}}$ and $\text{atoms}(N_2^p \cup N_2^f) \subseteq \text{atoms}(Q_1^p \cup Q_1^f)$, because $Q_1 \in (P_1)^{\lambda_{P_1} \cup \lambda_{P_1}^f}$. Then we can apply Lemma 5.20 and infer that:

$$Q_1 \rightarrow N_2 \equiv \text{elim}_{\phi}(Q_1, N_2) \equiv \left(\text{elim}_{\phi}(Q_1, N_2)\right)^{\text{def}} \equiv \bigvee_{M \in [\text{elim}_{\phi}(Q_1, N_2)]^{\text{def}}} M$$

As explained in Section 4.3, boolean combinations of minterms can only be transformed into sat-equivalent BSR(FO) formulae if there is no positive occurrence of test formulae $|h| \geq |U| - n$ or alloc$\alpha(x)$ (see Definition 4.8 and the second item of Lemma 4.9). Consequently, we relate the polarity of these formulae in some minterm $M \in \mu^\op(\phi) \cup \mu^{\op}(\phi)$ with that of a separating implication within $\phi$. The analysis depends on whether the universe is finite or infinite.

**Lemma 5.29.** For any quantifier-free $S_1^L$ formula $\phi$, the following properties hold:

1. For all $M \in \mu^{\op}(\phi)$, we have $M \cap \{|h| \geq |U| - n, |h| < |U| - n\} = \emptyset$.
2. If $|h| \geq |U| - n \in M$ (resp. $|h| < |U| - n \in M$) for some minterm $M \in \mu^{\op}(\phi)$, then a formula $\psi_1 \rightarrow \psi_2$ occurs at a positive (resp. negative) polarity in $\phi$.
3. If alloc$\alpha(x) \in M$ (resp. alloc$\neg\alpha(x) \in M$) for some minterm $M \in \mu^{\op}(\phi)$, then a formula $\psi_1 \rightarrow \psi_2$, such that $x \in \text{Var}(\psi_1) \cup \text{Var}(\psi_2)$, occurs at a positive (resp. negative) polarity in $\phi$.
4. If $M \cap \{\text{alloc} \} \neq \emptyset$ for some minterm $M \in \mu^{\op}(\phi)$, then a formula $\psi_1 \rightarrow \psi_2$, such that $x \in \text{Var}(\psi_1) \cup \text{Var}(\psi_2)$, occurs in $\phi$ at some polarity $p \in \{-1, 1\}$. Moreover, alloc$\alpha(x)$ occurs at a polarity $-p$, only if alloc$\alpha(x)$ is in the scope of a $\lambda^{\text{fin}}$ subformula of a formula $\text{elim}_{\phi}(M_1, M_2)$ used to compute $\bigvee_{M \in \mu^{\op}(\phi)} M$.

*See equation (15) in Lemma 5.20.*
(1) By induction on the structure of $\phi$, one shows that no literal from $\{|h| \geq |U| - n, |h| < |U| - n \mid n \in \mathbb{N}\}$ is introduced during the construction of $\mu^\ell(\phi)$.

(2) Let $\ell \in M \cap \{|h| \geq |U| - n, |h| < |U| - n \mid n \in \mathbb{N}\}$ be a literal. The proof is by induction on the structure of $\phi$:

- The cases $\phi = \text{emp}$, $\phi = x \leftarrow y$, $\phi = q(x)$ and $\phi = x = y$ are trivial, because $\ell \notin \mu^\ell(\phi)$.
- $\phi = \phi_1 \land \phi_2$: we have $M \in [M_1, M_2]$, for some minterms $M_i \in \mu^\ell(\phi_i)$, for $i = 1, 2$. By Proposition 5.26, since $\ell \notin \{|U| \geq n, |U| < n \mid n \in \mathbb{N}\}$, we deduce that $\ell \in M_1 \cup M_2$ and the proof follows from the induction hypothesis, since any formula occurring in $\phi_i$, $i = 1, 2$, occurs at the same polarity in $\phi$.
- $\phi = \neg \phi_1$: assuming $\mu^\ell(\phi_1) = [M_1, \ldots, M_m]$, we have $M \in [\ell_1, \ldots, \ell_m] \setminus \ell_i$, for some literals $\ell_i \in M_i$, $i \in [1, m]$. By Proposition 5.26, we deduce that $\ell = \overline{\ell_i}$ for some $i = 1, \ldots, n$, because $\ell \notin \{|U| \geq n, |U| < n \mid n \in \mathbb{N}\}$.

By the induction hypothesis, there exists a formula $\psi_1 \rightarrow \psi_2$ occurring at polarity $p \in \{1, -1\}$ in $\phi_1$, where $p = 1$ if $\ell_i = |h| \geq |U| - n$ and $p = -1$ if $\ell_i = |h| < |U| - n$. Then $\ell$ occurs at polarity $-p$ in $M$ and $\psi_1 \rightarrow \psi_2$ occurs at polarity $-p$ in $\phi$.

- $\phi = \phi_1 \ast \phi_2$: for $i = 1, 2$, there exist minterms $M_i \in \mu^\ell(\phi_i)$, $N_i \in (M_i)^{E(M_1 \cup M_2)}$ and $P_i \in (N_i)^{N_p}$, such that $M \in \left(\left(\text{elim}(P_1, P_2)\right)^{\text{ad}}\right)^m$. Since by hypothesis $\ell \in \{|h| \geq |U| - n, |h| < |U| - n \mid n \in \mathbb{N}\}$, by Proposition 5.26, this literal is necessarily introduced by $\text{elim}(P_1, P_2)$ and, by inspection of $\text{elim}(P_1, P_2)$, one of the following must hold:

  - $\ell = |h| \geq \min_{M_1} + \min_{M_2}$, where $\min_{M_1}$ and/or $\min_{M_2}$ is of the form $|U| - n$. By the induction hypothesis $\phi_1$ contains a formula $\psi_1 \rightarrow \psi_2$ at polarity 1, for some $i = 1, 2$, and the proof is completed.
  - $\ell = |h| = \max_{M_1} + \max_{M_2} - 1$, where $\max_{M_1}$ and/or $\max_{M_2}$ is of the form $|U| - n$. The proof is similar, with polarity 1.

(3) Let $\ell \in M \cap \{\text{alloc}(x), \neg\text{alloc}(x) \mid x \in \text{Var}\}$ be a literal occurring in some minterm $M \in \mu^\ell(\phi)$. The proof is by induction on the structure of $\phi$:

- The cases $\phi = \text{emp}$, $\phi = x \leftarrow y$, $\phi = x = y$ and $\phi = q(x)$ are trivial, because $\ell \notin \mu^\ell(\phi)$.
- The cases $\phi = \phi_1 \land \phi_2$ are similar to point (2) of the Lemma.
- $\phi = \phi_1 \ast \phi_2$: there exist minterms $M_i \in \mu^\ell(\phi_i)$, $N_i \in (M_i)^{E(M_1 \cup M_2)}$ and $P_i \in (N_i)^{N_p}$, such that $M \in \left(\left(\text{elim}(P_1, P_2)\right)^{\text{ad}}\right)^m$. By inspection of $\text{elim}(Q_1, Q_2)$, one of the following cases must occur:

  - $\ell = |h| \geq \min_{M_1} - \max_{M_2} - 1$, where $\min_{M_1}$ is of the form $|U| - n_2$. By the induction hypothesis, $\phi_1$ contains a formula $\psi_1 \rightarrow \psi_2$ at polarity 1, and this formula also occurs at polarity 1 in $\phi$, thus the proof is completed.

  - Note that if $\max_{M_1} = |U| - n_1$, then either $\min_{M_1} = |U| - n_2$ and $|h| \geq \min_{M_1} - \max_{M_2} - 1 = |h| \geq n_1 - n_2$, or $\min_{M_2} = n_2 \in \mathbb{N}$ and $|h| \geq \min_{M_2} - \max_{M_2} - 1 = |h| = |U| + (n_1 + n_2) = \sum_{i < n_1 + n_2} |U| \approx n \rightarrow |h| \geq n_1 + n_2 - n$ by Definition 4.3, thus $|h| \geq \min_{M_1} - \max_{M_2} - 1$ contains no literal of the above form.

  - $\ell = |h| = \max_{M_1} - \min_{M_2} - 1$. The proof is similar.

  - $\ell = |h| < |U| - \min_{M_1} - y_n(Y, M_1) + 1$. In this case since $\phi_1 \rightarrow \neg \phi_2$ occurs at polarity $-1$ in $\phi$, the proof is completed.
We verify that $\psi_2 \land \phi \equiv^\text{fin} \neg \psi_1 \land \phi$:

- If $(\mathcal{H}, s, I, b) \models \psi_2 \land \phi$, then $s(x)$ is allocated in $b$ and there are exactly $n$ unallocated cells. Then the heap $b'$ whose domain is the set of unallocated cells in $b$ is disjoint from $b$ and satisfies $\neg \text{alloc}(x) \land |h| = n$, which proves that $(\mathcal{H}, s, I, b) \models \neg \psi_1$.
- If $(\mathcal{H}, s, I, b) \models \neg \psi_1 \land \phi$, then there are exactly $n$ unallocated cells in $\mathcal{H}$, and there exists a heap $b'$ disjoint from $b$ with $n$ elements in its domain, non of which is $s(x)$. Thus, $s(x)$ must occur in the domain of $b$, and $(\mathcal{H}, s, I, b) \models \psi_2$.

However, the polarity of alloc$(x)$ is positive in $\psi_2$, whereas $x$ only occurs in the scope of neutral occurrences of $\rightarrow$ in $\neg \psi_1$.

We provide another example illustrating Property 4.

**Example 5.31.** Let $M_1 = \{|h| \geq 0, |h| < 2, \neg \text{alloc}(x)|$ and $M_2 = \{|h| \geq 0, |h| < \infty, \neg x \leftrightarrow x\}$. We have $M_1 \vdash M_2 \equiv^\text{fin} \neg x \equiv y \land |h| \geq 0 \land |h| < |U| \land \neg \text{alloc}(x) \rightarrow (|U| \geq 2 \land |h| < |U| - 1)$. The last two formulæ are parts of $\lambda^\text{fin}$ in Lemma 5.20: $|h| < |U|$ ensures that there exists at least one free location (so that there exists a disjoint heap satisfying $M_1$), and if $x$ is not allocated, then there must actually exist 2 free locations, since $x$ cannot be allocated in the extension. Observe that alloc$(x)$ occurs positively in the latter formula (since it is in scope of 2 negations), whereas $x$ only occurs in the scope of negative (or neutral) occurrences of $\rightarrow$ in $M_1 \ast M_2$ (i.e., positive occurrences of $\neg$). This happens because alloc$(x)$ occurs in $\lambda^\text{fin}$.

## 5.4 Testing Membership in $\mu(\phi)$ in PSPACE

Given a quantifier-free $\text{SL}^k$ formula $\phi$, the number of minterms occurring in $\mu^\text{fin}(\phi)$ (resp. $\mu^\text{inf}(\phi)$) is exponential in the size of $\phi$, in the worst case. Therefore, an optimal decision procedure cannot generate and store these sets explicitly, but rather must enumerate minterms lazily. We show that (i) the size of the minterms in $\mu^\text{fin}(\phi)$ is bounded by a polynomial in the size of $\phi$, and that (ii) the problem "given a minterm $M$, does $M$ occur in $\mu^\text{fin}(\phi)$ (resp. in $\mu^\text{inf}(\phi)$)?" is in PSPACE. To this aim, we define a measure on a quantifier-free formula $\phi$, which bounds...
the size of the minterms in the sets \( \mu^{\text{in}}(\phi) \) and \( \mu^{\text{inf}}(\phi) \), inductively on the structure of the formulae:

\[
\begin{align*}
M(\top) & \overset{\text{def}}{=} 0 \\
M(x \approx y) & \overset{\text{def}}{=} 0 \\
M(\text{emp}) & \overset{\text{def}}{=} 1 \\
M(\neg \phi_1) & \overset{\text{def}}{=} M(\phi_1) \\
M(\phi_1 \cdot \phi_2) & \overset{\text{def}}{=} \sum_{i=1}^{\ell} (M(\phi_i) + ||\text{Var}(\phi_i)||) \\
M(\phi_1 \rightarrow \phi_2) & \overset{\text{def}}{=} \max(M(\phi_1), M(\phi_2)) \\
M(\phi_1 \leftarrow \phi_2) & \overset{\text{def}}{=} \sum_{i=1}^{\ell} (M(\phi_i) + ||\text{Var}(\phi_i)||)
\end{align*}
\]

The intuition is that \( M(\phi) \) is an upper bound on natural number occurring in the test formulae in \( \mu^{\text{in}}(\phi) \cup \mu^{\text{inf}}(\phi) \), when viewed as linear inequalities on \( |U| \) and \( |h| \). For instance, \( M(\text{emp}) \) is 1, because \( \text{emp} \equiv |h| < 1 \), whereas \( M(x \rightarrow y) \) is 2, because \( M(x \rightarrow y) \equiv x \rightarrow y \land |h| \geq 1 \land |h| < 2 \). The extension to the standard connectives is straightforward, but the handling of the separating connectives is more involved: first, the combination of two inequalities may increase the bound (for instance, \( |h| \geq 1 \ast |h| \geq 2 \equiv |h| \geq 3 \)) and second, the elimination of these connectives yields additional inequalities (see Lemma 5.16 and Lemma 5.20).

**Proposition 5.32.** For any \( n \in \mathbb{N} \), we have:

\[
\begin{align*}
M(|h| \geq n) &= M(|U| \geq n) = n \\
M(|h| \geq |U| - n) &= n + 1
\end{align*}
\]

**Proof.** By induction on \( n \geq 0 \).

Note that, because \( |h| < \infty \) is a shorthand for \( \top \), we have \( M(|h| < \infty) = 0 \).

**Definition 5.33.** A minterm \( M \) is \( M \)-bounded by a formula \( \phi \), if for each literal \( \ell \in M \), the following hold:

(i) \( M(\ell) \leq M(\phi) \) if \( \ell \in [|h| \geq \min_{M_i}, |h| < \max_{M_i}] \); (ii) \( M(\ell) \leq 2M(\phi) + 1 \), if \( \ell \in [|U| \geq n, |U| < n \mid n \in \mathbb{N}| \).

**Proposition 5.34.** Given minterms \( M_1, \ldots, M_n \) all \( M \)-bounded by \( \phi \), each minterm \( M \in [M_1, \ldots, M_n] \) is also \( M \)-bounded by \( \phi \).

**Proof.** This is an immediate corollary of Proposition 5.26.

The following lemma provides the required result:

**Lemma 5.35.** Given a quantifier-free SL^k formula \( \phi \), each minterm \( M \in \mu^{\text{in}}(\phi) \cup \mu^{\text{inf}}(\phi) \) is \( M \)-bounded by \( \phi \).

**Proof.** We prove that each \( M \in \mu^{\text{in}}(\phi) \) is \( M \)-bounded by \( \phi \). The proof for \( M \in \mu^{\text{inf}}(\phi) \) follows from the observation that, because of the definition of \( \text{elim}^{\text{inf}} \), for each \( M \in \mu^{\text{inf}}(\phi) \) there exists \( M' \in \mu^{\text{in}}(\phi) \) such that \( M(M') = M(M') \). By induction on the structure of \( \phi \):

- If \( \phi = \text{emp} \) then \( \mu^{\text{in}}(\phi) = [|h| \geq 0 \land |h| < 1], M(|h| \geq 0) = 0, M(|h| < 1) = M(|h| \geq 1) = 1 \) and \( M(\text{emp}) = 1 \), by definition.
- If \( \phi = x \leftarrow y \) then \( \mu^{\text{in}}(\phi) = \{x \leftarrow y \land |h| \geq 1 \land |h| < 2\}, M(|h| \geq 1) = 1, M(|h| < 2) = 2 \) and \( M(x \rightarrow y) = 2 \), by definition.
- If \( \phi = q(y) \) with \( q \in \mathcal{T} \) then \( \mu^{\text{in}}(\phi) = \{q(y) \land |h| \geq 0 \land |h| < \infty\}, M(|h| \geq 0) = 0, M(|h| < \infty) = 0 \) and \( M(q(y)) = 0 \), by definition.
- If \( \phi = x \equiv y \) then \( \mu^{\text{in}}(\phi) = \{x \equiv y \land |h| \geq 0 \land |h| < \infty\} \) and \( M(|h| \geq 0) = M(|h| < \infty) = 0 \), by definition.
- If \( \phi = \phi_1 \land \phi_2 \), let \( \ell \in M \) be a literal, where \( M \in \mu^{\text{in}}(\phi_1 \land \phi_2) \) is a minterm. Then \( M \in [M_1, M_2] \), for some minterms \( M_i \in \mu^{\text{in}}(\phi_i), i = 1, 2 \). and the proof follows from Proposition 5.34, because \( M_i \) is \( M \)-bounded by \( \phi_i \) and \( M(\phi_i) \leq M(\phi_i) \), so that \( M_i \) is \( M \)-bounded by \( \phi_i \), for \( i = 1, 2 \).
If $\phi = \neg \phi_1$ assume that $\mu^a(\phi_1) = \{M_1, \ldots, M_m\}$. Let $\ell \in M$ be a literal, where $M \in \mu^a(\neg \phi_1)$ is a minterm. Then $M \in \{t_i^{m_1}, \ldots, t_i^{m_n}\}$, for some literals $t_i \in M_i$, $i \in [1, m]$. By the induction hypothesis, $\ell_i$ is $M$-bounded by $\phi$, for every $i \in 1, \ldots, n$, thus the same holds for $\ell_i$. Since $M([h] \geq 0) = M([h] < \infty) = 0$, we deduce that $[t_i^{m_1}]$ is $M$-bounded by $\phi$, and the proof follows from Proposition 5.34.

If $\phi = \phi_1 * \phi_2$, let $\ell \in M$ be a literal, where $M \in \mu^a(\phi_1 * \phi_2)$. Then there exist minterms $M_i \in \mu^a(\phi_1)$, $N_i \in (M_i)_{\{M_i \cup M_j\}}$, and $P_i \in (N_i)_{\{M_i \cup M_j\}}$, such that $M \in \{\text{elim}((P_i, P_j))\}^{\mu^a}$ for $i = 1, 2$. First assume that $\ell$ is of the form $[h] \geq t$ or $[h] < t$. We only consider the case where $\ell$ occurs in elim,$(P_1, P_2)$, the rest of the cases follow from Proposition 5.34. We distinguish the following:

If $\phi = \phi_1 * \phi_2$, let $\ell \in M$ be a literal, where $M \in \mu^a(\phi_1 * \phi_2)$. Then there exist minterms $M_i \in \mu^a(\phi_1)$, $N_i \in (M_i)_{\{M_i \cup M_j\}}$, and $P_i \in (N_i)_{\{M_i \cup M_j\}}$, such that $M \in \{\text{elim}((P_i, P_j))\}^{\mu^a}$ for $i = 1, 2$. First assume that $\ell$ is of the form $[h] \geq t$ or $[h] < t$. We only consider the case where $\ell$ occurs in elim,$(P_1, P_2)$, the rest of the cases follow from Proposition 5.34. We distinguish the following:

- $\ell$ is a subformula of $[h] \geq \min_{P_i} + \min_{P_j} = [h] \geq \min_{M_{i, j}}$, because $\min_{P_i} = \min_{M_i}$, for $i = 1, 2$, by Proposition 5.27. By the inductive hypothesis we have $M([h] \geq \min_{M_{i, j}}) \leq M(\phi_i)$, for $i = 1, 2$. If $\min_{M_i} \in \mathbb{N}$ for $i = 1, 2$ then $\ell = [h] \geq \min_{M_{i, j}}$, and we have:

\[
M(\ell) = M([h] \geq \min_{M_{i, j}} + \min_{M_j}) = M([h] \geq \min_{M_{i, j}}) + M([h] \geq \min_{M_j}) \leq M(\phi_1) + M(\phi_2) \leq M(\phi).
\]

If $\min_{M_i} = |U| - n_i$ and $n_i, \min_{M_{i, j}} \in \mathbb{N}$, then $\ell = [h] \geq \min_{M_{i, j}} + \min_{M_j}$ and we obtain:

\[
M(\ell) = M([h] \geq \min_{M_{i, j}} + \min_{M_j}) = M([h] \geq |U| - (n_i - \min_{M_{i, j}})) \leq M([h] \geq |U| - n_i) \leq M(\phi_1) \leq M(\phi).
\]

Otherwise, $\min_{M_i} = |U| - n_i$, for $i = 1, 2$, where $n_1, n_2 \in \mathbb{N}$, thus by Definition 4.3:

\[
|h| \geq \min_{M_{i, j}} + \min_{M_j} = |h| \geq 2 \cdot |U| - n_1 - n_2 \leq |U| < 1 + n_1 + n_2 \land \\
\land_{1 \leq n \leq n_1 + n_2} [U] = n \to |h| \geq 2n - n_1 - n_2
\]

and

\* either $\ell \in [|U| \geq n, |U| < n + 1]$ for some $n \in [1, n_1 + n_2]$, and we have $M(\ell) \leq n + 1 \leq n_1 + n_2 + 1 \leq 2(M(\phi_1) + M(\phi_2)) + 1 = 2M(\phi) + 1$;

\* or $\ell = |h| \geq 2n - n_1 - n_2$ for some $n \in [1, n_1 + n_2]$, and we have $M(\ell) = 2n - n_1 - n_2 \leq n_1 + n_2 = M(\phi_1) + M(\phi_2) = M(\phi)$.

- The proof in the case where $\ell$ is a subformula of $[h] < \max_{M_{i, j}} + \max_{M_j} - 1$ is analogous.

$\ell = |h| \geq \#_a(P_i) + |Y|_{P_i} + \min_{P_i}$, where $Y \subseteq \text{av}(P_{i-1}) \setminus \text{av}(P_i)$, for some $i = 1, 2$. Because $Y \cap \text{av}(P_i) = \emptyset$, we have (Definition 5.6 and Proposition 5.27): $\#_a(P_i) + |Y|_{P_i} \leq ||\text{Var}(P_i)|| + ||\text{Var}(P_{i-1})|| \leq ||\text{Var}(\phi_1)|| + ||\text{Var}(\phi_2)||$ and thus $M(\ell) \leq M([h] \geq \min_{M_{i, j}} + |\text{Var}(\phi_1)| + |\text{Var}(\phi_2)|) \leq M(\phi)$. Now assume $\ell \in [|U| \geq m, |U| < m + 1] \land [M \in \mathbb{N}]$. Then one of the following holds:

- $\ell \in \text{dc}(P_i)^\mu$, for some $i = 1, 2$, and we have two cases:

\* $\ell \in [|U| \geq n_1 + n_2 + 1, |U| < n_1 + n_2$, where $\min_{P_i} = \min_{M_i} = n_1$ and $\max_{P_i} = \max_{M_i} = |U| - n_2$. By the induction hypothesis, we have $n_1, n_2 \leq M(\phi_i)$, thus $M(\ell) \leq 2M(\phi_i) + 1 \leq 2M(\phi) + 1$.

\* $\ell = |U| \geq \sqrt{\max_{x \in \text{av}(M_i)}(\delta_x(P_i) + 1)}$, in which case either $\text{Var}(M_i) \cup \text{Var}(M_2) = \emptyset$ so that we have

\[
\ell \leq |U| \geq \sqrt{\max_{x \in \text{av}(M_i)}(\delta_x(P_i) + 1)} = 0 \text{ and the proof is immediate, or we have } M(\ell) \leq \sqrt{|\text{Var}(M_i)|} + 1 \leq 2M(\phi) + 1.
\]

- $\ell = |U| > n_1 + n_2$, where $Y \subseteq \text{av}(M_{i-1}) \setminus \text{av}(M_i)$ and $\max_{M_i} = |U| - n_i$, for some $i = 1, 2$. Because $Y \cap \text{av}(P_i) = \emptyset$, we have $\#_a(P_i) + |Y|_{P_i} \leq ||\text{Var}(P_i)|| + ||\text{Var}(P_{i-1})|| \leq ||\text{Var}(\phi_1)|| + ||\text{Var}(\phi_2)||$ and thus $M(\ell) \leq M(\phi_1) + |\text{Var}(\phi_2)| \leq 2M(\phi) + 1$.

- $\ell = \phi = \phi_1 \circ \phi_2$, consider a literal $\ell \in M$, where $M \in \mu^a(\phi_1 \circ \phi_2)$. Then there exist minterms $M_i \in \mu^a(\phi_1)$ and $N_i \in (M_i)_{\{M_i \cup M_j\}}$, for $i = 1, 2$, and minterms $P_i \in (N_i)_{\{M_i \cup M_j\}}$ and $Q_i \in (P_i)_{\{M_i \cup M_j\}}$, such that
$M \in \left( \text{elim}^{\mu}(Q_1, N_3) \right)^{\text{def}}_{m}$. We only consider the case where $\ell$ occurs in $\text{elim}^{\mu}(Q_1, N_3)$, in the remaining cases, the result follows directly from Proposition 5.34. If $\ell$ is of the form $|h| \geq t$ or $|h| < t$ then either:

- $\ell$ is a subformula of $|h| \geq \min_{N_1} - \max_{Q_1} - 1 = |h| \geq \min_{M_1} - \max_{M_2} - 1$, because $\min_{N_1} = \min_{M_1}$ and $\max_{Q_1} = \max_{P_1} = \max_{N_1} = \max_{M_1}$ by Proposition 5.27. Then $\min_{M_1} \in \{n_2, |U| - n_2\}$ and $\max_{M_2} \in \{n_1, |U| - n_1\}$ with $n_1, n_2 \in \mathbb{N}_{\text{ord}}$, and by the induction hypothesis $n_i \leq M(\varphi_i)$. If $\max_{M_2} = n_1$ or $\min_{M_1} \neq n_2$, then by an inspection of the different cases and using Proposition 5.32, we have $\ell = |h| \geq \min_{M_2} - \max_{M_1} + 1,$ thus:

$$M(\ell) = M(|h| \geq \min_{M_2} - \max_{M_1} + 1) \leq n_1 + n_2 \leq M(\varphi_1) + M(\varphi_2) \leq M(\varphi).$$

Otherwise, $\min_{M_2} = n_2$ and $\max_{M_1} = |U| - n_1$ hence:

* either $\ell \in \{n_1, |U| - n_1\}$ for some $n \in [1, n_1 + n_2 - 1]$ and we have $M(\ell) \leq n + 1 \leq n_1 + n_2 \leq 2(M(\varphi_1) + M(\varphi_2)) + 1 = M(\varphi)$;

* or $\ell = |h| \geq n_1 + n_2 - n$, for some $n \in [1, n_1 + n_2 - 1]$ and we have $M(\ell) = n_1 + n_2 - n \leq n_1 + n_2 - 1 \leq M(\varphi_1) + M(\varphi_2) = M(\varphi)$.

- The case $\ell = |h| < \min_{N_1} - \max_{Q_1}$ is proved in a similar way.

- If $\ell = |h| < |U| - \min_{N_1} - \max_{Q_1} + 1$, for some $Y \subseteq \text{Var}(Q_1 \cup N_2)$. Because $n_{\forall}(Q_1) \subseteq n_{\forall}(P_1) \subseteq n_{\forall}(\varphi_1) \cup n_{\forall}(\varphi_2)$, we have $n_{\forall}(Y, Q_1) \leq |\text{Var}(\varphi_1)| + |\text{Var}(\varphi_2)|$. Moreover, $\min_{N_1} = \min_{M_1}$ by Proposition 5.27. We distinguish the following cases:

  * If $\min_{N_1} \in \mathbb{N}$, we compute:

    $$M(\ell) = \min_{M_1} + n_{\forall}(Y, Q_1) - 1,$$

    by Proposition 5.32

    $$\leq M(\varphi_1) + |\text{Var}(\varphi_1)| + |\text{Var}(\varphi_2)|$$

    since $M(|h| \geq \min_{M_1}) \leq M(\varphi_1)$, by the inductive hypothesis.

    $$\leq M(\varphi).$$

  * Otherwise, $\min_{M_1} = |U| - n_1$, for some $n_1 \in \mathbb{N}$, thus $\ell = |h| < n_1 - n_{\forall}(Y, Q_1) + 1$. By Proposition 5.32, we have $M(|h| \geq \min_{M_1}) = n_1 + 1$ and $M(\ell) = n_1 - n_{\forall}(Y, Q_1) + 1$, therefore:

    $$M(\ell) = M(|h| \geq \min_{M_1}) - n_{\forall}(Y, Q_1)$$

    $$\leq M(\varphi_1) \leq M(\varphi).$$

- If $\ell$ is of the form $|U| \geq m$ or $|U| < m$, with $m \in \mathbb{N}$, then either:

  - if $\ell \in \text{dc}(Q_1) \cup \text{dc}(N_2)$ the argument is similar to the previous case $\phi = \phi_1 * \phi_2$.

  - otherwise, $\ell = |U| \geq n_{\forall}(Y, Q_1)$ and either $\min_{M_1} \in \mathbb{N}$ in which case $M(\ell) = \min_{N_1} + n_{\forall}(Y, Q_1)$.

Since $M(\varphi)$ is polynomially bounded by $\text{size}(\varphi)$, this entails that it is possible to check whether $M \in \mu(\varphi)$ (resp. $\mu(\varphi)$) using space bounded also by a polynomial in $\text{size}(\varphi)$).

**Proposition 5.36.** Given a quantifier-free $\text{SL}_k$ formula $\phi$ and a minterm $M \in \mu(\varphi)$, we have $\text{size}(M) = O(\text{size}(\varphi)^2)$. As a consequence, $N(\bigvee M(\mu(\varphi))) = O(\text{size}(\varphi)^2)$ and $N(\bigvee M(\mu^m(\varphi))) = O(\text{size}(\varphi)^2)$.

**Proof.** We give the proof for $M \in \mu(\varphi)$, the case $M \in \mu^m(\varphi)$ being similar. Let $\ell \in M$ be a literal. We distinguish the following cases, based on the form of $\ell$:

- $\ell \in \{\text{alloc}(x), \neg\text{alloc}(x) \mid x \in \text{Var}\}$: $\ell$ must occur in $\varphi$ or has been introduced by $\mu(\varphi)$, in which case, at most $2 \cdot |\text{Var}(\varphi)|$ such literals are introduced.

- $\ell \in \{x \leftarrow y, \neg x \rightarrow y \mid x \in \text{Var}, y \in \text{Var}^k \cup \{q(z) \mid q \in \mathcal{F}, z \in \text{Var}^m(q)\}$: $\ell$ occurs in $\varphi$, since $\mu(\varphi)$ does not introduce literals of this form.
The size of the above formula is of the order of \( O(\text{size}(\phi)^2) \). This second result follows immediately.

**Proposition 5.37.** Let \( L \) be a set of literals and \( \phi \) be a boolean combination of literals. The problem of deciding whether \( L \in (\phi)_{\text{inf}} \) is in \( \text{NSPACE}(\text{size}(L) + \text{size}(\phi)) \).

**Proof.** W.l.o.g., we may assume that \( \phi \) is in negation normal form. The algorithm is nondeterministic and proceeds recursively on the structure of \( \phi \):

- If \( \phi = \ell \) is a literal: then \( (\phi)_{\text{inf}} = \{ \ell \} \) hence it suffices to verify whether \( L = \{ \ell \} \), using \( O(\text{size}(L) + \text{size}(\phi)) \) space.
- If \( \phi = \phi_1 \lor \phi_2 \): then \( (\phi)_{\text{inf}} = (\phi_1)_{\text{inf}} \cup (\phi_2)_{\text{inf}} \) and we check that one of \( L \in (\phi_1)_{\text{inf}} \) and \( L \in (\phi_2)_{\text{inf}} \) holds. By the induction hypothesis, checking \( L \in (\phi_1)_{\text{inf}} \) can be done using \( O(\text{size}(L) + \text{size}(\phi_1)) \) space. Since the working space used for \( L \in (\phi_2)_{\text{inf}} \) can be reused for \( L \in (\phi_1)_{\text{inf}} \), the entire check takes \( O(\text{size}(L) + \text{size}(\phi)) \) space.
- If \( \phi = \phi_1 \land \phi_2 \): then \( L \in (\phi)_{\text{inf}} \Leftrightarrow L = L_1 \cup L_2 \) with \( L_1 \in (\phi_1)_{\text{inf}} \) and \( L_2 \in (\phi_2)_{\text{inf}} \), thus we guess two subsets \( L_1 \) and \( L_2 \) with \( L_1 \cup L_2 = M \) and check that \( L_1 \in (\phi_1)_{\text{inf}} \), using \( O(\text{size}(L_1) + \text{size}(\phi_1)) \) space, for \( i = 1, 2 \). Since we must store \( L_2 \) during the check \( L_1 \in (\phi_1)_{\text{inf}} \) and the working space can be reused for \( L_2 \in (\phi_2)_{\text{inf}} \), the entire check takes \( O(\text{size}(L) + \text{size}(\phi)) \) space.

**Proposition 5.38.** Let \( L \) be a set of literals and let \( M_1, M_2 \) be minterms. Checking whether \( L \in ((\text{elim}_1(M_1, M_2))_{\text{inf}} \) is in \( \text{NSPACE}(\text{size}(L) + \text{size}(M_1) + \text{size}(M_2)) \).

**Proof.** The algorithm proceeds by induction on the structure of \( (\text{elim}_1(M_1, M_2))_{\text{inf}} \) as in the proof of Proposition 5.37. The only difference concerns the subformulæ \( \eta_{ij} \) (Line 6 in Lemma 5.16) which cannot be constructed explicitly since they are of exponential size. However, \( \eta_{ij} \) is of positive polarity, and to check that \( L \in (\eta_{ij})_{\text{inf}} \), it suffices to guess a set of variables \( Y \subseteq \av(M_j) \setminus \av(M_i) \) and check whether:

\[
L \in \left( \text{alloc}(Y) \rightarrow (|h| \geq \#_a(M_i) + |Y|_{M_i} + \min_{M_j} \land \#_a(M_i) + |Y|_{M_i} < \min_{M_i} \right)_{\text{inf}}
\]

The size of the above formula is of the order of \( O(\text{size}(M_1) + \text{size}(M_2)) \), thus \( L \in ((\text{elim}_1(M_1, M_2))_{\text{inf}} \) can be checked in \( \text{NSPACE}(\text{size}(L) + \text{size}(M_1) + \text{size}(M_2)) \), by Proposition 5.37.

**Proposition 5.39.** Let \( L \) be a set of literals and let \( M_1, M_2 \) be minterms. The problems whether \( L \in ((\text{elim}_n(M_1, M_2))_{\text{inf}} \) and \( L \in ((\text{elim}_n(M_1, M_2))_{\text{inf}} \) are both in \( \text{NSPACE}(\text{size}(L) + \text{size}(M_1) + \text{size}(M_2)) \).

**Proof.** The proof is similar to that of Proposition 5.38 (again, the formula \( \lambda_1 \) is exponential, but does not have to be constructed explicitly).

**Proposition 5.40.** Checking whether \( M \in [M_1, \ldots, M_n] \), where \( M, M_1, \ldots, M_n \) are minterms, \( n \geq 2 \), is in \( \text{NSPACE}(\text{size}(M) + \text{size}(M_1) + \ldots + \text{size}(M_n)) \).

**Proof.** The proof is by induction on \( n \geq 2 \). If \( n = 2 \) then by definition of \( [M_1, M_2] \) it suffices to check that

\[
M = M_1 \lor M_2 \land M_1 \land M_2 \land M_1 \land M_2 \land M_1 \land M_2 \land M_1 \land M_2 \land \mu \land \nu \text{ for some } \mu \in \text{min}(M_1, M_2), \nu \in \text{max}(M_1, M_2).
\]

By definition, the size of each formula in \( \text{min}(M_1, M_2) \) \( \cup \max(M_1, M_2) \) is of the order of \( O(\text{size}(M_1) + \text{size}(M_2)) \), thus the algorithm requires \( O(\text{size}(M) + \text{size}(M_1) + \text{size}(M_2)) \) space.
If \( n > 2, M \in [M_1, \ldots, M_n] \) \( \Rightarrow \) \( M \in [M', M_n] \), where \( M' \in [M_1, \ldots, M_{n-1}] \). By Proposition 5.26, the literals in \( M' \) are either literals from \( M_1, \ldots, M_{n-1} \) or occur in \([U] \geq m_1 + m_2, [U] < m_1 + m_2, [U] \geq m_1 + m_2 + 1, [U] < m_1 + m_2 + 1 \), where \( M_1 \uplus \cdots \uplus M_{n-1} \) contains two literals \( \ell_1 \) and \( \ell_2 \) and \( \ell_1 \) is of the form \( [h] \geq m_i, [h] < m_i, [h] \geq [U] - m_i \) or \( [h] < [U] - m_i \), for \( i = 1, 2 \). Thus \( \text{size}(M') \leq \sum_{i=1}^{n-1} \text{size}(M_i) \). The nondeterministic algorithm guesses and stores a minterm \( M_i' \) of size at most \( \sum_{i=1}^{n-1} \text{size}(M_i) \) and checks that \( M \in [M', M_n] \) and that \( M_i' \in [M_1, \ldots, M_{n-1}] \).

According to the base case \( n = 2 \), the first check takes up \( \text{O}(\text{size}(M) + \text{size}(M_1') + \text{size}(M_n)) = \text{O}(\text{size}(M) + \sum_{i=1}^{n} \text{size}(M_i)) \) space, and the second check takes space \( \text{O}(\text{size}(M_1') + (\sum_{i=1}^{n-1} \text{size}(M_i))^2) = \text{O}((\sum_{i=1}^{n} \text{size}(M_i))^2) \), by the induction hypothesis. Because we only need to store \( M_i' \) between the two checks, the algorithm takes \( \text{O}(\text{size}(M) + (\sum_{i=1}^{n} \text{size}(M_i))^2) \) space.

\[\square\]

**Proposition 5.41.** Let \( M \) be a minterm and let \( L \) be a set of literals. The problem of checking whether \( M = [L]^n \) is in \( \text{NSPACE}(\text{size}(M) + \sum_{i \in L} \text{size}(l)) \).

**Proof.** By definition, \( [L]^n = [[\ell_1]^m, \ldots, [\ell_n]^m] \), with \( L = \{\ell_1, \ldots, \ell_n\} \), and each minterm \( [\ell_i]^m \) is of size \( \text{O}(\text{size}(\ell_i)) \), thus the proof follows immediately from Proposition 5.40.

**Lemma 5.42.** Given a minterm \( M \) and an \( \text{SL}^k \) formula \( \phi \), the problems of checking whether \( M \in \mu^m(\phi) \) and \( M \in \mu^m(\phi) \) are in \( \text{PSPACE} \).

**Proof.** We show the existence of a nondeterministic algorithm that decides \( M \in \mu^m(\phi) \) in space \( \text{O}(\text{size}(M) + \text{size}(\phi)^k) \). The PSPACE upper bound is by an application of Savitch’s Theorem [21]. We only give the proof for \( M \in \mu^m(\phi) \), the proof for \( M \in \mu^m(\phi) \) is similar and omitted. By induction on the structure of \( \phi \), we distinguish the following cases:

- \( \phi = \text{emp} \): we check \( M = [h] \approx 0 \) in space \( \text{O}(\text{size}(M) + \text{size}(\phi)) \).
- \( \phi = x \mapsto y \): we check \( M = [x \mapsto y \land [h] = 1] \) in space \( \text{O}(\text{size}(M) + \text{size}(\phi)) \).
- \( \phi = q(x_1, \ldots, x_{\#(q)}) \): we check \( M = \{q(x_1, \ldots, x_{\#(q)}) \land [h] \geq 0 \land [h] < \infty \} \) in space \( \text{O}(\text{size}(M) + \text{size}(\phi)) \).
- \( \phi = \phi_1 \land \phi_2 \): \( M \in \mu^m(\phi) \) \( \Rightarrow \) \( M \in [M_1, M_2] \) with \( M_i \in \mu^m(\phi_i) \), for every \( i = 1, 2 \). Since, by Proposition 5.36, \( \text{size}(M_i) = \text{O}(\text{size}(\phi_i)^2) = \text{O}(\text{size}(\phi)^2) \), for \( i = 1, 2 \), it suffices to guess two such minterms \( M_1 \) and \( M_2 \), check that \( M_i \in \mu^m(\phi_i) \), \( i = 1, 2 \), and that \( M \in [M_1, M_2] \). By the induction hypothesis, checking \( M_i \in \mu^m(\phi_i) \) requires space \( \text{O}(\text{size}(M_i) + \text{size}(\phi_i)^2) \), for each \( i = 1, 2 \), and by the proof of Proposition 5.40 in the case \( n = 2 \), checking \( M \in [M_1, M_2] \) requires space \( \text{O}(\text{size}(M) + \text{size}(M_1) + \text{size}(M_2)) = \text{O}(\text{size}(M) + \text{size}(\phi)) \).
- \( \phi = \neg \phi_1 \): \( M \in \mu^m(\phi) \) if and only if \( M \in \left[ [\ell_1]^m, \ldots, [\ell_m]^m \right] \), for some literals \( \ell_i \in M_i \), \( i \in [1, m] \), where \( \mu^m(\phi) = [M_1, \ldots, M_m] \). For any \( i \in [1, m] \), we distinguish the following cases:
  - if \( \ell_i \in \{x \mapsto y, \neg x \mapsto y \mid x \in \text{Var}, y \in \text{Var}^k\} \) then \( \ell_i \) occurs in \( \phi_1 \), thus there are at most size(\( \phi_1 \)) such literals.
  - if \( \ell_i \in \{x \approx y, \neg x \approx y \mid x, y \in \text{Var}\} \) then there are at most \( 2 ||\text{Var}(\phi)||^2 \) such literals.
  - if \( \ell_i \in [U] \geq n, [U] < n \mid n \in \mathbb{N} \), by Lemma 5.35, \( M(\ell_i) \leq 2M(\phi_1) + 1 \), thus there are at most \( 2M(\phi_1) + 1 = \text{O}(\text{size}(\phi_1)^2) \) such literals.

Summing up, we obtain that \( \|\|\ell_i\| \mid i \in [1, m]\| = \text{O}(\text{size}(\phi_1)^2) \). Thus it suffices to guess a set \( \{\ell'_1, \ldots, \ell'_n\} \) of literals and a set of minterms \( \left\{ M'_1, \ldots, M'_n \right\} \) such that \( \ell'_i \in M'_i \), where \( n = \text{O}(\text{size}(\phi_1)^2) \) and size(\( M'_i \)) = \( \text{O}(\text{size}(\phi_1)^2) \), for all \( i \in [1, n] \). Then we can check that:
- \( M'_i \in \mu^m(\phi_1) \), which can be done in space \( \text{O}(\text{size}(M'_i) + \text{size}(\phi_1)^2) = \text{O}(\text{size}(\phi_1)^2 + \text{size}(\phi_1)^2) = \text{O}(\text{size}(\phi_1)^2) \), by the inductive hypothesis,
we carve two nontrivial fragments of $\phi$

This section contains the results concerning decidability of the (in)finite satisfiability problems within the BSR($\text{SL}^k$) fragment. First, we show that, contrary to BSR(FO), the satisfiability of BSR($\text{SL}^k$) is undecidable for $k \geq 2$. Second, we carve two nontrivial fragments of BSR($\text{SL}^k$), for which the infinite and finite satisfiability problems are both PSPACE-complete, defined based on restrictions of (i) polarities of the occurrences of the separating implication, and (ii) occurrences of universally quantified variables in the scope of separating implications. These results draw a rather precise chart of decidability within the BSR($\text{SL}^k$) fragment.

6.1 Undecidability of BSR($\text{SL}^k$)

**Theorem 6.1.** The finite and infinite satisfiability problems are both undecidable for formulæ in BSR($\text{SL}^k$) even if the formulæ contain no uninterpreted predicates.

**Proof.** Let $\phi = \forall x. \phi$ be a formulæ in BSR($\text{SL}^k$), where $\phi$ is quantifier-free, contains no predicate symbol, one variable $x$, one constant symbol $c$ and two monadic function symbols $f$ and $g$ of sort $U$. It is known that the finite satisfiability problem is undecidable for such formulæ, by Proposition 2.3. We reduce this problem to the infinite and finite satisfiability problems for BSR($\text{SL}^k$) formulæ. We proceed by first flattening each term in $\phi$ consisting of nested applications of $f$ and $g$. The result is an equivalent sentence $\phi_{\text{flat}} = \forall x_1 \ldots \forall x_n. \phi_{\text{flat}}$, in which the only terms are $x_i, c, f(x_i), g(x_i), f(c)$ and $g(c)$, for $i \in [1, n]$. For example, the formulæ $\forall x. f(g(x)) \approx c$ is flattened into $\forall x_1 \forall x_2. (g(x_1)) \neq x_2 \lor f(x_2) \approx c$. The formal construction is standard and thus omitted. We define the following BSR($\text{SL}^k$) formulæ, for $\hat{f} \in \{\text{fin, inf}\}$:

$$\phi_{d,\hat{f}} \overset{\text{def}}{=} \forall x \land x_c \leftarrow (y_c, z_c) \land \forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n \forall z_1 \ldots \forall z_n \ldots \bigwedge_{i=1}^{n} (x_i \leftarrow (y_i, z_i) \rightarrow \phi_0)$$

where $\alpha^{\text{fin}} \overset{def}{=} |h| \geq |U| - 0$, $\alpha^{\text{inf}} \overset{def}{=} \forall x \forall y \forall z \cdot x \leftrightarrow (y, z) \rightarrow \text{alloc}(y) \land \text{alloc}(z)$ and $\phi_u$ is obtained from $\phi_{\text{flat}}$ by replacing each occurrence of $c$ by $x_c$, each term $f(c)$ (resp. $g(c)$) by $y_c$ (resp. $z_c$) and each term $f(x_i)$ (resp. $g(x_i)$) by $y_i$ (resp. $z_i$). Next, we show that the following statements are equivalent:

1. $\phi_{\text{flat}}$ has a finite model $(\mathcal{I}, s, I)$,
2. $\phi_u^{\text{fin}}$ has a finite model $(\mathcal{I}, s', \ell, b)$, and
3. $\phi_u^{\text{inf}}$ has an infinite model $(\mathcal{I}^{\infty}, s', I, b)$.

“$(1) \Rightarrow (2)$” We define the store $s'' \overset{def}{=} s[x_c \leftarrow c^f, y_c \leftarrow f^f(c^f), z_c \leftarrow g^f(c^f)]$ and the heap $h$ such that dom($h$) = $\mathcal{I}$ and $b(\ell) = (f^f(\ell), g^f(\ell))$, for all $\ell \in \mathcal{I}$. By construction, we have $(\mathcal{I}, s', h) \models \alpha^{\text{fin}} \land x_c \leftarrow (y_c, z_c)$, because dom($h$) = $\mathcal{I}$ and $b(\ell) = (f^f(\ell), g^f(\ell))$. Consider a store $s''' \overset{def}{=} s''[x_i \leftarrow \ell_i, y_i \leftarrow \ell_i, z_i \leftarrow \ell'_i | i = 1, \ldots, n]$, for an arbitrary set $\{\ell_i, \ell'_i | i \in [1, n]\} \subseteq \mathcal{I}$ and assume that $(\mathcal{I}, s'', h) \models \land_{i=1}^n x_i \leftarrow (y_i, z_i)$. Then by definition of $h$, for all $i \in [1, n]$, we have $\ell'''_i = f^f(\ell_i)$ and $\ell''_i = g^f(\ell_i)$; hence, $(\mathcal{I}, s'', h) \models \phi_u$. Since $\ell_i, \ell'_i$ and $\ell''_i$ are arbitrary, for $i \in [1, n]$, this proves that $(\mathcal{I}, s', b)$ is a finite model of $\phi_u^{\text{fin}}$.

“$(2) \Rightarrow (3)$” We define $\mathcal{I}^{\infty} \overset{def}{=} \mathcal{I} \cup L$, where $L$ is an infinite set of locations not in $\mathcal{I}$. Clearly $(\mathcal{I}^{\infty}, s'', h) \models \alpha^{\text{inf}}$, because $x \leftrightarrow (y, z)$ is false for any extension of $s''$ with a pair of the form $[x \leftarrow \ell, [y \leftarrow \ell] \cup [z \leftarrow \ell]$, where $\ell \in \mathcal{I}$. Furthermore, the valuation of $x_c \leftarrow (y_c, z_c)$ is unchanged between $(\mathcal{I}, s', h)$ and $(\mathcal{I}^{\infty}, s', h)$. Consider a store $s''' \overset{def}{=} s''[x_i \leftarrow \ell_i, y_i \leftarrow \ell_i, z_i \leftarrow \ell''_i | i = 1, \ldots, n]$, for an arbitrary set $\{\ell_i, \ell'_i, \ell''_i | i \in [1, n]\} \subseteq \mathcal{I}$ and assume that $(\mathcal{I}, s'''', h) \models \land_{i=1}^n x_i \leftarrow (y_i, z_i)$. Then necessarily, $\{\ell_i, \ell'_i, \ell'''_i | i \in [1, n]\} \cap L = \emptyset$. Next, we show that $(\mathcal{I}, s'', h) \models \phi_u$ by induction on the structure of $\phi_u$. Since $(\mathcal{I}, s', h) \models \phi_u$ by the hypothesis, we have $(\mathcal{I}^{\infty}, s', h) \models \phi_u$, thus $(\mathcal{I}^{\infty}, s, h) \models \phi_u^{\text{inf}}$.

“$(3) \Rightarrow (1)$” Let $\mathcal{I}^{\infty} \overset{def}{=} \text{dom}(h) \cup \{\ell_1, \ell_2 | \exists \ell \in \mathcal{I}^{\infty}. b(\ell) = (\ell_1, \ell_2)\}$. Since $h$ is finite, so is $\mathcal{I}$. Let $\mathcal{I}$ be an arbitrary\(^{10}\) store on $\mathcal{I}$ and define $\mathcal{I}''$ such that:

- $c^f = s'(x_c)$, and,
- for each $\ell \in \mathcal{I}$, such that $b(\ell) = (\ell', \ell'')$, we have $f^f(\ell) = \ell'$ and $g^f(\ell) = \ell''$.

Note that $f^f(\ell) \in \mathcal{I}$, because by hypothesis $(\mathcal{I}^{\infty}, s', h) \models x_c \leftarrow (y_c, z_c)$, hence $s'(x_c) \in \text{dom}(h)$. Similarly, $f^f(\ell), g^f(\ell) \in \mathcal{I}$, for each $\ell \in \mathcal{I}$, by the definition of $\mathcal{I}$. Moreover, since $(\mathcal{I}^{\infty}, s', h) \models \alpha^{\text{inf}}$ we obtain that $\mathcal{I}^{\infty}$ and $\mathcal{I}$ are well-defined total functions. For each set $\{\ell_i | i = 1, \ldots, n\} \subseteq \mathcal{I}$, the function $s'' = [x_i \leftarrow \ell_i, y_i \leftarrow f^f(\ell_i), z_i \leftarrow g^f(\ell_i) | i = 1, \ldots, n]$ is a store on $\mathcal{I}^{\infty}$ such that $(\mathcal{I}^{\infty}, s''', h) \models x_i \leftarrow (y_i, z_i)$ for every $i \in [1, n]$, hence $(\mathcal{I}^{\infty}, s'', h) \models \phi_u$. By induction on the structure of $\phi_{\text{flat}}$, one shows that $(\mathcal{I}^{\infty}, s'', h) \models \phi_u = (\mathcal{I}, s', I) \models \phi_{\text{flat}}$. Since $(\mathcal{I}^{\infty}, s', h) \models \phi_u$, we have $(\mathcal{I}, s, I) \models \phi_{\text{flat}}$, \(\square\)

Note that, by the previous proof, the undecidability result still holds for finite satisfiability if a single occurrence of $\rightarrow$ is allowed, in a ground formula (indeed, we may take $\alpha^{\text{fin}} = ([h] \geq |U| - 0) = (\neg \text{emp} \leftrightarrow \bot)$. For infinite satisfiability one occurrence of $\rightarrow$ is still sufficient, however there must be a universally quantified variable within the scope of $\rightarrow$.

The reductions (18) use positive occurrences of test formula $[h] \geq |U| - n$ and alloc($x$), where $x$ is universally quantified. We obtain decidable subsets of BSR(SL$^k$) by devising conditions that are sufficient to discard positive occurrences of such formula from $\mu(\phi)$, where $\phi \in \{\text{fin}, \text{inf}\}$ and $\forall y_1, \ldots, y_m \cdot \phi$ is a BSR(SL$^k$) formula. Note that $\mu^{\text{inf}}(\phi)$ contains no formulae of the form $[h] \geq |U| - n$ (as such test formulae are trivially false in all infinite structures) which explains why slightly less restrictive conditions are needed for infinite structures. As we shall see (Proposition 6.5), these conditions are sufficient to ensure that the formula $\forall y_1, \ldots, y_m \cdot \forall m \in \mu^{\text{inf}}(\phi) M$ is BSR-compatible (but not that $\forall y_1, \ldots, y_m \cdot \forall M \in \mu^{\text{inf}}(\phi) M$ is BSR-compatible, see Section 6.2.3 for details).

---

\(^{9}\)Note that an equivalent definition of $\alpha^{\text{fin}} = \alpha^{\text{fin}} \overset{def}{=} \forall x \cdot \text{alloc}(x)$.

\(^{10}\)The store is arbitrary because $\phi$ contains no free variables.
6.2 Decidability Proofs

6.2.1 Model Checking. We first show that the first-order model checking problem, considering the translation of
minterms to FO, is in PSPACE. We first recall the following well-known result, proved for instance in [22].

**Proposition 6.2.** Let \( S \) be an FO-structure and let \( \phi \) be an FO formula. The problem of testing whether \( S \models \phi \) is in PSPACE.

Proposition 6.2 does not by itself entail the desired result since \(|\mu(M)|\) is exponential w.r.t. size(\( \phi \)). We need the following:

**Lemma 6.3.** Given a finite FO-structure \( S = (U, s, I) \) and an SL formula \( \forall y_1 \ldots \forall y_m \cdot \phi \) where \( \phi \) is quantifier-free, the problem \( S \models \tau(\forall y_1 \ldots \forall y_m \cdot \\sqrt{M_{\mu(\phi)}} M) \) is in PSPACE, for each \( \tau \in \{\text{fin}, \text{inf}\}. \)

**Proof.** Since PSPACE is closed under complement (see, e.g., [2, Corollary 4.21]), we consider instead the problem \( S \models \neg \tau(\forall y_1 \ldots \forall y_m \cdot \\sqrt{M_{\mu(\phi)}} M) \). Because \( \tau(\cdot) \) is homomorphic w.r.t. the propositional connectives, we have the equivalences:

\[
\neg \tau(\forall y_1 \ldots \forall y_m \cdot \\sqrt{M_{\mu(\phi)}} M) \equiv \neg \forall y_1 \ldots \forall y_m \cdot \tau(\\sqrt{M_{\mu(\phi)}} M) \\
\equiv \exists y_1 \ldots \exists y_m \cdot \neg \tau(\\sqrt{M_{\mu(\phi)}} M) \\
\equiv \exists y_1 \ldots \exists y_m \cdot \\sqrt{M_{\mu(\phi)}} \neg \tau(M)
\]

To check that \( S \not\models \tau(\forall y_1 \ldots \forall y_m \cdot \\sqrt{M_{\mu(\phi)}} M) \), we may thus guess locations \( \ell_1, \ldots, \ell_m \in U \), and check that

\( (\langle \forall \rangle, s[\ell_1 \leftarrow \ell_1] \ldots [\ell_m \leftarrow \ell_m], I) \models \sqrt{M_{\mu(\phi)}} \neg \tau(M) \).

There remains to prove that the latter test in is PSPACE.

To this aim, we consider again the complement problem (\( (\langle \forall \rangle, s[\ell_1 \leftarrow \ell_1] \ldots [\ell_m \leftarrow \ell_m], I) \not\models \sqrt{M_{\mu(\phi)}} \neg \tau(M) \)). We guess a minterm \( M \) that is \( M \)-bounded by \( \phi \), then check that \( M \in \mu(M) \) and that \( (\langle \forall \rangle, s[\ell_1 \leftarrow \ell_1] \ldots [\ell_m \leftarrow \ell_m], I) \models \tau(M) \). The first check is in PSPACE, by Lemma 5.42. The second check is also in PSPACE, by Proposition 6.2.

**Remark 6.4.** Note that the size of an FO-structure \( S = (U, s, I) \) is exponential w.r.t. the arity of the symbols in \( \mathcal{F} \). In our context, the arity of all symbols is bounded by a constant, except that of the special symbol \( \forall \) that encodes the heap. Further, in the following (see for instance the proof of Theorem 6.11), we will only consider structures that satisfy the formula Heap in Definition 4.6, so that \( \forall^\mathcal{F} \) is a partial function and \(|\mu^\mathcal{F}| \leq |U||\). Hence we may assume that the size of \( S \) is polynomial in \(|U| + k + \text{dom}(s)|\).

6.2.2 Infinite Satisfiability (BSR\text{inf}(SL_k)). We start by showing decidability, in PSPACE, of the infinite satisfiability problem for the BSR\text{inf}(SL_k) fragment. We first establish the following result:

**Proposition 6.5.** Let \( \varphi = \forall y_1 \ldots \forall y_m \cdot \phi \) be a formula in BSR\text{inf}(SL_k), where \( \phi \) is quantifier-free. The formula

\( \chi \overset{\text{def}}{=} \sqrt{M_{\mu(\phi)}} M \) is BSR-compatible.

**Proof.** By Lemma 5.29 (1), no formula of the form \(|h| \geq |U| - i \) occurs positively in \( \chi \). Furthermore, if alloc\((x)\) positively occurs in \( \chi \), then it must occur in a minterm in \( \mu(\phi) \), and by Lemma 5.29 (3), \( x \) necessarily occurs in the scope of a positive occurrence of \( \neg \), which entails by definition of BSR\text{inf}(SL_k) that \( x \notin \{y_1, \ldots, y_m\} \).

Consequently, \( \chi \) is BSR-compatible.

**Proposition 6.5**, together with Lemma 5.28, ensures that a reduction from BSR\text{inf}(SL_k) to BSR(FO) is feasible. However, we also have to ensure that the cardinality of the universe is infinite and that the cardinality of the heap is finite, which cannot be expressed in FO.

To this aim, we rely on existing results about the cardinality of models of BSR(FO) formulæ. The definition and theorem below are from [12] (they have been slightly adapted to handle formulæ containing free variables).
The intuition is that, due to the above condition, the interpretation of the predicate symbols in an \( \mathcal{U} \) of \( \phi \) always has an infinite model. To ensure that the domain of the predicate \( \sigma \) is finite, necessarily \( \mathcal{U} \) is in \( \mathcal{PSPACE} \).

Definition 6.6. Let \( \mathcal{S} = (\mathcal{U}, s, I) \) be an FO-structure. Let \( A \overset{\text{def}}{=} \{ s(x) \mid x \in \text{dom}(s) \} \cup \{ c \overset{\mathcal{F}}{\mathcal{F}}, \sigma(c) = 0, \sigma(c) \in \mathcal{U} \} \) and \( B := \mathcal{U} \setminus A \). The structure \( \mathcal{S} \) is \( m \)-repetitive if \( ||B|| \geq m \) and there exists a total order \( \prec \) on \( \mathcal{U} \) such that for every \( n \leq m \) and strictly increasing sequences \( e_1 < \cdots < e_n \) and \( e'_1 < \cdots < e'_n \) of elements in \( B \), for every predicate symbol \( q \in \mathcal{F} \) and every \( d_1, \ldots, d_{\#(q)} \in A \cup \{ e_1, \ldots, e_n \} \) the following holds:

\[
(d_1, \ldots, d_{\#(q)}) \in q^\mathcal{F} \Leftrightarrow (d'_1, \ldots, d'_{\#(q)}) \in q^\mathcal{F}, \quad \text{where } d'_i \overset{\text{def}}{=} \begin{cases} e'_j & \text{if } d_i = e_j \\ d_i & \text{otherwise} \end{cases}
\]

The following theorem, proved in [12], characterizes the existence of an infinite model of a BSR(FO) formula.

**Theorem 6.7.** Consider a BSR(FO) formula \( \phi \) containing \( n \) free variables and constants, no existential quantifier and \( m \) distinct universally quantified formulae. The formula \( \phi \) has an infinite model if and only if it has an \( m \)-repetitive model \( (\mathcal{U}, s, I) \) such that \( ||\mathcal{U}|| \leq n + m \).

**Proof.** See [12, Theorems 4 and 5]. The addition of free variables is not problematic as they can be handled as constants.

Definition 6.8. Testing whether a first-order structure \( \mathcal{S} = (\mathcal{U}, s, I) \) is \( m \)-repetitive for a given \( m \in \mathbb{N} \) is in \( \mathcal{PSPACE} \).

**Proof.** The algorithm is straightforward: it is clear that \( A \) and \( B \) can be computed in polynomial time, then it suffices to guess some total order \( \prec \) on \( \mathcal{U} \), to iterate over the increasing sequences \( (e_1, \ldots, e_n), (e'_1, \ldots, e'_n) \in B^n \), with \( n \leq m \), over the predicate symbols \( q \in \mathcal{F} \) and elements \( d_1, \ldots, d_{\#(q)} \in A \cup \{ e_1, \ldots, e_n \} \), to compute in each case the elements \( d'_1, \ldots, d'_{\#(q)} \) according to Definition 6.6 and to check that the equivalence \( (d_1, \ldots, d_{\#(q)}) \in q^\mathcal{F} \Leftrightarrow (d'_1, \ldots, d'_{\#(q)}) \in q^\mathcal{F} \) holds.

Theorem 6.7 and Proposition 6.8 provide an effective method to decide whether a formula \( \phi \) in BSR(FO) has an infinite model. To ensure that the domain of the predicate \( \phi \) encoding the heap is finite we rely on the following definition and result:

Definition 6.9. Let \( \phi \) be a BSR(FO) formula. We denote by \( \overline{\phi} \) the formula:

\[
\forall x_1, \ldots, x_{k+1} : \bigwedge_{i=1}^{k+1} \bigwedge_{z \in \text{Var}(\phi) \cup \text{Const}(\phi)} \neg x_i \approx z \rightarrow \neg \phi(x_1, \ldots, x_{k+1})
\]

where \( x_1, \ldots, x_{k+1} \) are pairwise distinct variables not occurring in \( \phi \).

**Proposition 6.10.** Let \( \phi \) be a BSR(FO) formula. The two following assertions are equivalent.

- \( \phi \) has an infinite model \( (\mathcal{U}, s, I) \) such that \( \nu^\mathcal{F} \) is finite.
- \( \phi \wedge \overline{\phi} \) has an infinite model.

**Proof.** Assume that \( \phi \) admits a model \( (\mathcal{U}, s, I) \) such that \( ||\mathcal{U}|| = \infty \) and \( ||\nu^\mathcal{F}|| \in \mathbb{N} \). Let \( A \overset{\text{def}}{=} \{ s(x) \mid x \in \text{Var}(\phi) \cup \{ c \overset{\mathcal{F}}{\mathcal{F}}, \sigma(c) \in \mathcal{U} \} \} \) and \( B = \mathcal{U} \setminus A \). Since \( \nu^\mathcal{F} \) is finite, necessarily \( B \) and \( \mathcal{U} \) are both infinite. By Proposition 2.2, the restriction \( (\mathcal{U}', s', I') \) of \( (\mathcal{U}, s, I) \) to \( \mathcal{U}' \) validates \( \phi \), since \( \phi \) is a BSR(FO) formula and \( A \subseteq \mathcal{U}' \). It is clear that \( (\mathcal{U}', s', I') \models \phi \), since by
definition, \( U^{k+1} \cap p^F = A^{k+1} \cap p^F \). Conversely, let \( S = (U, s, I) \) be an infinite model of \( \varphi \land \varphi_0 \). Then by definition of \( \varphi_0 \) for every \((\ell_1, \ldots, \ell_{k+1})\) such that \((\ell_1, \ldots, \ell_{k+1}) \in p^F\), and for every \( i \in [1, k+1] \), either \( \ell_i = c^F \) for some \( c \in \mathcal{F}(\varphi) \), or \( \ell_i = s(x) \) for some \( x \in \text{Var}(\varphi) \). Since \( \mathcal{F}(\varphi) \) and \( \text{Var}(\varphi) \) are both finite, \( p^F \) is also finite.

Putting all results together, we obtain the first decidability result of this paper:

**Theorem 6.11.** The infinite satisfiability problem for \( BS \text{R}^{\text{inf}}(SL^k) \) is PSPACE-complete.

**Proof.** PSPACE-hardness is an immediate consequence of the fact that the quantifier-free fragment of \( SL^k \), without the separating implication, but with the separating conjunction and negation, is PSPACE-hard \([7, Proposition 5]\).

To show membership in PSPACE, let \( \varphi = \forall y_1 \ldots \forall y_m \cdot \phi \) be a formula in \( BS \text{R}^{\text{inf}}(SL^k) \), where \( \phi \) is quantifier-free and \( \text{Var}(\varphi) = \{x_1, \ldots, x_n\} \). Let \( \phi' \overset{\text{def}}{=} \forall y_1 \ldots \forall y_m \cdot \chi \), with \( \chi \overset{\text{def}}{=} \bigvee_{M \in p^{\text{inf}}(\phi)} M \) and let \( \psi \overset{\text{def}}{=} \tau(\phi') \land \mathcal{A}(\phi') \). By Lemma 5.28, \( \phi \equiv_{\text{inf}} \phi' \). By Proposition 6.5, \( \chi \) is BSR-compatible and we deduce by Lemma 4.9 that \( \phi' \) (and hence \( \varphi \)) has an infinite model iff \( \psi \) has an infinite model where the interpretation of \( p \) is finite.

We now show how to solve the latter problem. By Proposition 4.10, \( \psi \) is a BSR(FO) formula with no existential variable and contains \( k \cdot n + (k + 6) \cdot N(\chi) + 5 \) constants. By Proposition 5.36, \( N(\chi) = O(\text{size}(\phi)^2) \), thus we deduce that \( \psi \) is a BSR(FO) formula, with \( O(k \cdot \text{size}(\phi)^2) \) constants and free variables. By Proposition 6.10, \( \psi \) has an infinite model where the interpretation of \( p \) is finite iff \( \psi \land \psi_0 \) has an infinite model. By Theorem 6.7, \( \psi \land \psi_0 \) has an infinite model iff it has an \( m \)-repetitive model \((U, s, I)\) of cardinality \( ||U|| = O(k \cdot \text{size}(\phi)^2) \), because \( \psi_0 \) is a BSR(FO) formula with no existential variable and contains no constant or free variable other than those in \( \psi \).

The algorithm is then defined as follows. We guess an FO-structure \((U, s, I)\) such that \( ||U|| = O(k \cdot \text{size}(\phi)^2) \) and \((U, s, I) \models \text{Heap} \) (where \( \text{Heap} \) is the formula in Definition 4.6). Note that since \( k \) may depend on the input, \( U^k \) is of exponential size, hence in principle the interpretation of \( p \) may be exponential. However, since we assume that \((U, s, I) \models \text{Heap} \), for every element \( x \in U \), there is at most one vector \( y \in U^k \) such that \((x, y) \in p^F \), hence \( ||p^F|| \leq ||U|| \). To ensure that \( \text{Heap} \) holds, it suffices to guess a subset of \( U \) (the set of allocated locations), and choose for every element \( x \) in this subset one vector \( y \in U^k \) such that \((x, y) \in p^F \). Moreover, the arity of each predicate symbol in \( \varphi \) that are different from \( p \) is bounded by a constant, thus their interpretation is polynomial w.r.t. \( U \). Then we check that \((U, s, I) \equiv_{\text{def}} \tau(\phi') \land \mathcal{A}(\phi') \land \psi_0 \). This test is feasible in PSPACE:

- the problem of testing whether \((U, s, I) \equiv_{\text{def}} \tau(\phi') \) is in PSPACE by Proposition 6.8.
- the problem \((U, s, I) \models \tau(\phi') \) is in PSPACE by Lemma 6.3.
- the problems \((U, s, I) \models \mathcal{A}(\phi') \) and \((U, s, I) \models \psi_0 \) are both in PSPACE, by Proposition 6.2.

**Remark 6.12.** The algorithm given in the proof of Theorem 6.11 is based on guessing some structure of size \( s \), with \( s = O(k \cdot \text{size}(\phi)^2) \). To apply the algorithm one needs of course to know an upper bound of \( s \). Because our aim in the present paper is only to prove the existence of such an algorithm, we do not bother to give this bound explicitly, as this would only hinder readability, and we only state that it exists. However, the bound can easily be extracted from the above proofs, if needed. Similarly, an explicit bound on the size of the minterms in \( p^{\text{inf}}(\phi) \) could be extracted from the proof of Lemma 5.42.

**6.2.3 Finite Satisfiability \((BSR^{\text{fin}}(SL^k))\).** We now prove that finite satisfiability is PSPACE-complete for the class \( BSR^{\text{fin}}(SL^k) \), defined as the set of formulæ with no positive occurrence of separating implications. Even with this stronger restriction, the previous proof based on a translation to first-order logic cannot be carried over without any additional argument, because Proposition 6.5 does not hold for \( BSR^{\text{fin}}(SL^k) \). The problem is that, in the case of a finite universe, alloc(\( x \)) test formulæ may occur at a positive polarity, even if every \( \phi_1 \equiv \phi_2 \) subformula occurs at a negative polarity, due to the positive occurrences of alloc(\( x \)) within the subformula \( \lambda^\text{fin} \) in
the definition of $\text{elim}_\approx(M_1, M_2)$ (Equation (15) in Lemma 5.20), see also Example 5.30. As previously discussed, positive occurrences of $\text{alloc}(x)$ hinder the translation into BSR(FO), because of the existential quantifiers that may occur in the scope of a universal quantifier.

The solution is to distinguish a class of finite structures $(U, s, I, h)$. Given $\alpha \in \mathbb{N}$, we consider the so-called $\alpha$-controlled structures, for which there exists a set of locations $\ell_1, \ldots, \ell_\alpha$, such that every location $\ell \in U \setminus \{\ell_1, \ldots, \ell_\alpha\}$ points to a tuple from the set $(\ell_1, \ldots, \ell_\alpha, \ell)$. An example of a 3-controlled structure is given in Figure 2.

![Fig. 2. A finite 3-controlled SL\(^2\) structure.](image)

**Definition 6.13.** An SL-structure $S$ is $\alpha$-controlled if $S \models \exists \overline{x}_1, \ldots, \overline{x}_\alpha \cdot C(\alpha)$, with

$$C(\alpha) \triangleq \forall x \cdot \bigwedge_{i=1}^\alpha x \approx \overline{x}_i \lor \bigwedge_{y \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_\alpha, x)} x \leftarrow y$$

where $\text{vect}^k(x_1, \ldots, x_n)$ is the set of $k$-tuples of symbols in $\{x_1, \ldots, x_n\}$, and $\overline{x}_1, \ldots, \overline{x}_\alpha, x$ are pairwise distinct variables. Analogously, an FO-structure $S$ is $\alpha$-controlled if $S \models \exists \overline{x}_1, \ldots, \overline{x}_\alpha \cdot \tau(C(\alpha))$, with

$$\tau(C(\alpha)) = \forall x \cdot \bigwedge_{i=1}^\alpha x \approx \overline{x}_i \lor \bigwedge_{y \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_\alpha, x)} p(x, y)$$

Any $\alpha$-controlled SL-structure is finite, since $|U| = \text{dom}(h) \cup \{s(\overline{x}_1), \ldots, s(\overline{x}_\alpha)\}$, but its cardinality is not bounded. Furthermore, if $|U| \leq \alpha$, then $(U, s, I, h)$ is necessarily $\alpha$-controlled.

**Overview of the Proof for Finite Satisfiability.** For a formula $\phi = \forall y_1, \ldots, y_m \cdot \phi$ in $\text{BSR}^\text{fin}(\text{SL}^k)$, we distinguish the following cases:

1. If $\phi$ has an $\alpha$-controlled model $S$, the formula obtained by replacing each occurrence of an $\text{alloc}(x)$ with $\bigwedge_{i=1}^\alpha (x \approx \overline{x}_i \rightarrow \text{alloc}(\overline{x}_i))$ in $\forall y_1, \ldots, y_m \forall M \in \text{BSR}(\phi)$ is satisfied by $S$ (as stated by Proposition 6.15).
2. Otherwise, each finite model of $\phi$ is non-$\alpha$-controlled and we can build a model $S$, with a sufficiently large universe, such that each test formula $\theta \in \{U\} \geq n, |h| < |U| - n \mid n \in \mathbb{N}\}$ becomes true in $S$. Assume $\text{alloc}(x)$ occurs positively in a $\lambda^\text{fin}$ subformula of some formula $\text{elim}_\approx(M_1, M_2)$. The latter must have been generated by the elimination of a separating implication from $\phi$, hence alloc($x$) occurs in a disjunction with a formula of the form $|h| < |U| - n \land |U| \geq n_2$; its truth value in $S$ can thus be ignored and the entire subformula deleted.

In both cases, we obtain an equisatisfiable universally quantified boolean combination of test formulae with no positive occurrence of alloc($y$) formulae. We translate this into an equisatisfiable BSR(FO) formula, for which finite satisfiability is decidable and apply a similar argument to that for the infinite case, to obtain the PSPACE upper bound.

**The Case of Controlled Structures.** We first consider the case where the considered models are $\alpha$-controlled.

**Proposition 6.14.** Let $S = (U, s, I)$ be an FO-structure. The problem of testing whether $S \models \tau(C(\alpha))$ is in $P$.

**Proof.** Note that the size of $C(\alpha)$ is exponential w.r.t. $k$. However, to test that $S \models \tau(C(\alpha))$, it suffices to check that for every $u \in U \setminus s(\{\overline{x}_1, \ldots, \overline{x}_\alpha\})$, there exist $u_1, \ldots, u_k$ such that $(u, u_1, \ldots, u_k) \in p^I$ and $u_1, \ldots, u_k \subseteq \{u\} \cup s((\overline{x}_1, \ldots, \overline{x}_\alpha))$, which can be done in time polynomial in size($S$).  \[\square\]
Proposition 6.15. If \( x \) is a variable distinct from \( \overline{x}_1, \ldots, \overline{x}_n \), then:

\[
C(\alpha) \models \forall x \cdot \langle \text{alloc}(x) \leftrightarrow \bigwedge_{i=1}^a (x \approx \overline{x}_i \to \text{alloc}(\overline{x}_i))\rangle
\]

Proof. This is immediate, since \( C(\alpha) \) entails that every element distinct from \( \overline{x}_1, \ldots, \overline{x}_a \) is allocated. \( \Box \)

Lemma 6.16. Given a formula \( \varphi \in \text{BSR}^\text{fin}(\text{SL}^k) \) and a number \( \alpha \in \mathbb{N} \) encoded in unary, the problem of checking whether \( \varphi \) has an \( \alpha \)-controlled model is in \( \text{PSPACE} \).

Proof. We assume that the formula \( \varphi \) is of the form \( \forall y_1, \ldots, y_m \cdot \phi \), with \( \text{Var}(\varphi) = \{x_1, \ldots, x_n\} \). Let \( \overline{x}_1, \ldots, \overline{x}_a \) be pairwise distinct variables not occurring in \( \{x_1, \ldots, x_n, y_1, \ldots, y_m\} \). It is clear that \( \varphi \) admits an \( \alpha \)-controlled model iff it admits a model that also validates \( C(\alpha) \). Let \( \chi \equiv \bigvee_{M \models \mu^{\text{fin}}(\phi)} M \), let \( \chi' \) the formula obtained from \( \chi \) by replacing every formula \( \text{alloc}(x) \) with \( \bigwedge_{i=1}^a (x \approx \overline{x}_i \to \text{alloc}(\overline{x}_i)) \) and let \( \varphi' = \forall y_1 \ldots \forall y_m \cdot \chi' \). By Proposition 6.15, \( C(\alpha) \models \chi \leftrightarrow \chi' \), thus, since any model of \( C(\alpha) \) is finite, we deduce by Lemma 5.28 that \( C(\alpha) \models \varphi \leftrightarrow \varphi' \).

Consequently \( \varphi \) has an \( \alpha \)-controlled model iff \( \varphi' \land C(\alpha) \) has a model. By Lemma 5.29 (2), \( \chi' \) contains no test formulae \( |h| \geq |U| - 1 \) at positive polarity, thus the same holds for \( \chi' \). Moreover, the formula \( \chi' \) contains no occurrence of \( \text{alloc}(y_i) \), since by definition the only test formulae \( \text{alloc}(x) \) occurring in \( \chi' \) are such that \( x \in \{\overline{x}_1, \ldots, \overline{x}_a\} \). Hence \( \chi' \) is \( \text{BSR} \)-compatible, and we deduce by Lemma 4.9 that \( \varphi \) has an \( \alpha \)-controlled model iff \( \varphi \equiv \tau(\varphi' \land C(\alpha)) \land \mathcal{A}(\varphi' \land C(\alpha)) \) has a finite model.

We now show how to solve the latter problem. Since \( \varphi' \) and \( C(\alpha) \) contain no cardinality constraints other than those in \( \chi \), we have \( N(\varphi' \land C(\alpha)) \leq N(\chi) \), thus by Proposition 5.36, we deduce that \( N(\varphi' \land C(\alpha)) = O(k \cdot \text{size}(\varphi)^2 + k \cdot \alpha) \) constants and free variables (since \( \text{Var}(\varphi' \land C(\alpha)) = \text{Var}(\varphi') \cup \text{Var}(C(\alpha)) = \{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_a\} \)). By Proposition 2.2, \( \psi \) has a finite model iff it has a model \( \langle U, s, I \rangle \), with \( ||U|| = O(k \cdot \text{size}(\varphi)^2 + k \cdot \alpha) \).

The algorithm is defined as follows (see the proof of Theorem 6.11 for details). We first guess a structure \( \langle U, s, I \rangle \) such that \( ||U|| = O(k \cdot \text{size}(\varphi)^2 + k \cdot \alpha) \) and \( \langle U, s, I \rangle \models \text{Heap} \). Then we check that \( \langle U, s, I \rangle \models \tau(\varphi') \) (as done in the proof of Lemma 6.3, except that all formulae \( \text{alloc}(x) \) are replaced by \( \bigwedge_{i=1}^a (x \approx \overline{x}_i \to \text{alloc}(\overline{x}_i)) \)), that \( \langle U, s, I \rangle \models \tau(C(\alpha)) \) (using Proposition 6.14) and that \( \langle U, s, I \rangle \models \mathcal{A}(\varphi' \land C(\alpha)) \) (using Proposition 6.2). \( \Box \)

The General Case. To handle the case where no \( \alpha \)-controlled model exists, the following results are used.

Proposition 6.17. Let \( \langle U, s, I \rangle \) be a non-\( \alpha \)-controlled FO-structure satisfying the (Heap) axiom, defined on page 11. Let \( E \subseteq U \), with \( ||E|| \leq \alpha \). There exists an element \( u \in U \setminus E \) such that either \( u \) is not allocated, or there exist \( u_1, \ldots, u_k \in U \) and \( j \in [1, k] \) such that \( \langle u, u_1, \ldots, u_k \rangle \in p^f \) and \( u_j \notin E \cup \{u\} \).

Proof. Because \( \langle U, s, I \rangle \) is not \( \alpha \)-controlled, we have

\[
\langle U, s, I \rangle \models \forall \overline{x}_1, \ldots, \overline{x}_a \cdot \exists x \cdot \bigwedge_{i=1}^a \neg x \approx \overline{x}_i \land \bigwedge_{y \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_a, x)} \neg p(x, y).
\]

Let \( s' \) be any extension of \( s \) to \( \overline{x}_1, \ldots, \overline{x}_a \) such that \( s'((\overline{x}_1, \ldots, \overline{x}_a)) = E \) (such as store necessarily exists since \( ||E|| \leq \alpha \)). We have \( \langle U, s', I \rangle \models \exists x \cdot \bigwedge_{i=1}^a \neg x \approx \overline{x}_i \land \bigwedge_{y \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_a, x)} \neg p(x, y) \) hence \( \exists t \) contains an element \( u \notin E \) such that \( \langle U, s'[x \leftarrow u], I \rangle \models \bigwedge_{y \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_a, x)} \neg p(x, y) \). If \( u \) is not allocated then the proof is completed. Otherwise, let \( \langle u, u_1, \ldots, u_k \rangle \in p^f \) and assume that \( v_j \in [1, k], v_j \in E \cup \{u\} \). Since \( s'((\overline{x}_1, \ldots, \overline{x}_a)) = E \), this entails that for every \( j \in [1, k] \), there exists \( y_j \in \{x_1, \ldots, x_n, x\} \) such that \( v_j = s'(y_j) \). But then \( \langle x, y_1, \ldots, y_k \rangle \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_a, x) \) and \( \langle U, s'[x \leftarrow u], I \rangle \models p(x, y_1, \ldots, y_k) \), which contradicts the fact that \( \langle U, s'[x \leftarrow u], I \rangle \models \bigwedge_{y \in \text{vect}^k(\overline{x}_1, \ldots, \overline{x}_a, x)} \neg p(x, y) \). \( \Box \)

LEMMA 6.18. Let \( n \in \mathbb{N} \). Consider a BSR(FO) formula \( \varphi \), let \( m \overset{df}{=} ||\text{Var}(\varphi)|| + ||\text{Const}(\varphi)|| \) and let \( \alpha \geq (k+2)\cdot n + m \).

If \( \varphi \cup \{\text{Heap}\} \) has a non-\( \alpha \)-controlled model \( S \) then there is a restriction of \( S \) that also validates \( \varphi \cup \{\text{Heap}\} \) and has at least \( n \) unallocated elements.

**Proof.** The result is trivial if \( n = 0 \), since \( S \) is a restriction of itself and trivially contains at least 0 unallocated elements. Thus we assume that \( n > 0 \). Let \( S = (\mathbb{U}, s, J) \) be a non-\( \alpha \)-controlled model of \( \varphi \cup \{\text{Heap}\} \). Let \( A = s(\text{Var}(\varphi)) \cup \{c^f \mid c \in \text{Const}(\varphi)\} \). Note that by definition, \( ||A|| \leq m \). We construct a sequence of pairwise distinct elements \( x_1, \ldots, x_n \in \mathbb{U} \) and a sequence of sets of elements \( Y_0 \subseteq Y_1 \subseteq \cdots \subseteq Y_n \subseteq \mathbb{U} \setminus A \) such that \( ||Y_i|| \leq i \), \( x_1, \ldots, x_i \notin Y_i \) and for every \( j \in [1, i] \), either \( x_j \) is unallocated or points to a vector containing an element of \( Y_i \). The sequence is constructed inductively as follows. Let \( Y_0 \overset{df}{=} \emptyset \). Assume that \( x_1, \ldots, x_i, Y_1, \ldots, Y_i \) have been constructed, for some \( i \in [0, n-1] \). Let \( X = \{x_1, \ldots, x_i\} \), \( E = \{z_1, \ldots, z_k \mid (x_j, z_1, \ldots, z_k) \in p^f, 1 \leq j \leq i\} \). Because \( S \models \text{Heap} \), for every \( j \in [1, i] \) there is at most one vector \( (z_1, \ldots, z_k) \) such that \( (x_j, z_1, \ldots, z_k) \in p^f \), hence \( ||E|| \leq k \cdot i \leq k \cdot n \). Further, \( ||X|| = i \leq n \) and \( ||Y_i|| \leq i \leq n \). Thus \( ||E \cup U X \cup Y_i|| \leq ||E|| + ||A|| + ||X|| + ||Y_i|| \leq k \cdot n + m + 2 \cdot n \leq \alpha \). Thus, since \( \varphi \) is not \( \alpha \)-controlled, by Proposition 6.17, there exists an element \( x_{i+1} \notin E \cup U X \cup Y_i \) such that either \( x_{i+1} \) is not allocated, or there exists a (unique) vector \( z_i \) such that \( (x_{i+1}, z_i) \in p^f \) and \( z \in Y_{i+1} \). Thus \( X = \{x_1, \ldots, x_n\} \) is arbitrary, this entails that \( (\mathbb{U}, s, J) \) is a non-\( \alpha \)-controlled model of \( \varphi \cup \{\text{Heap}\} \) satisfying \( \varphi \). Thus, \( \varphi \) has a finite model.

Theorem 6.20. The finite satisfiability problem for BSR(\( \mathbb{N} \))(SL\( ^k \)) is PSPACE-complete.

**Proof.** PSPACE-hardness is proved using the same argument as in the proof of Theorem 6.11, which does not rely on the infiniteness of the universe.

Let \( \varphi \overset{df}{=} \bigvee y_1, \ldots, y_m \). \( \varphi \) is a formula in BSR(\( \mathbb{N} \))(SL\( ^k \)), where \( \varphi \) is quantifier-free and \( \text{Var}(\varphi) = \{x_1, \ldots, x_n\} \).

Let \( \chi \overset{df}{=} \bigvee_{\text{M} \in \varphi} M \) and \( \alpha = (k + 2) \cdot (N(\chi) + 1) + (k + 1) \cdot n + (k + 6) \cdot N(\chi) + 5 \). We first test whether \( \varphi \) admits an \( \alpha \)-controlled model, which can be done in PSPACE, by Lemma 6.16 since, by Proposition 5.36, \( N(\chi) = O(\text{size}(\varphi)^2) \), thus \( \alpha = O(k \cdot \text{size}(\varphi)^2) \). In this case, \( \varphi \) has a finite model, and otherwise \( \varphi \) has a finite model iff it has a non-\( \alpha \)-controlled finite model. We now prove that \( \varphi \) does not have any \( \alpha \)-controlled model.

Let \( \varphi' \overset{df}{=} \bigvee y_1, \ldots, y_m \cdot \chi' \), where \( \chi' \) is obtained from \( \chi \) by replacing all positive occurrences of a formula \( \text{alloc}(x) \), where \( x \in \{x_1, \ldots, x_n, y_1, \ldots, y_m\} \), by \( \bot \). We prove that \( \varphi' \) has a finite model iff \( \varphi \) has a finite model.
By Lemma 5.28, $\phi \equiv^\text{fin} \forall y_1, \ldots, y_m \cdot \chi$. Because the replaced occurrences of alloc($x$) are all positive, it is clear that $\chi' \models \chi$, thus $\phi' = \forall y_1, \ldots, y_m \cdot \chi' \equiv^\text{fin} \phi$ and the direct implication holds. Now, assume that $\phi$ admits a finite model. Note that the above assumption this model is necessarily non-$\alpha$-controlled.

The formula $\chi$ can be written in cnf as $\chi_1 \land \chi_2$ where $\chi_1$ is a conjunction of clauses not containing any literal alloc($x$) and $\chi_2$ is a conjunction of clauses containing at least one such literal. It is clear that $N(\chi_1), N(\chi_2) \leq N(\chi)$ and $\phi \equiv \forall y_1, \ldots, y_m \cdot \chi_1$, thus $\forall y_1, \ldots, y_m \cdot \chi_1$ has a non-$\alpha$-controlled model. By Proposition 4.10, the formula $\xi = \tau(\forall y_1, \ldots, y_m \cdot \chi_1) \land A(\forall y_1, \ldots, y_m \cdot \chi_1)$ is a BSR($\text{FO}$) formula with at most $n$ free variables and $k \cdot n + (k + \beta) \cdot N(\chi) + 5$ constants, since $N(\chi_1) \leq N(\chi)$. Furthermore, by Proposition 6.19, $\xi$ admits a non-$\alpha$-controlled FO-model such that the interpretation of $\phi$ is finite, since $\forall y_1, \ldots, y_m \cdot \chi_1$ has a non-$\alpha$-controlled SL-model. By Lemma 6.18, and by definition of $\alpha$, this entails that there exists an FO-model of $\xi$ with strictly more than $N(\chi)$ unallocated elements and such that the interpretation of $\phi$ is finite. By Lemma 5.29 (2), the formula $\chi$ (hence also $\chi_1$) contains no positive occurrence of a formula of the form $|h| \geq |U| - i$, and by definition, $\chi_1$ contains no positive occurrence of a formula alloc($x$). Thus $\forall y_1, \ldots, y_m \cdot \chi_1$ is BSR-compatible. By Lemma 4.9 (2), we deduce that $\forall y_1, \ldots, y_m \cdot \chi_1$ admits an SL-model $S = (\Omega, s, I, b)$ with strictly more than $N(\chi)$ unallocated elements. Assume that $S \not\models \forall y_1, \ldots, y_m \cdot \chi^C_1$. This entails that there exist $e_1, \ldots, e_m \in \Omega$ and a clause $C$ in $\chi_2$ such that $(\Omega, s^C, I, b) \not\models C'$, where $s^C = s[x_1 \leftarrow e_1 | 1 \leq i \leq m]$ and $C'$ is obtained from $C$ by removing all the literals alloc($x$). By definition $C$ must contain at least one literal alloc($x$). Because all occurrences of $\leftarrow$ in $\phi$ are negative or neutral, by Lemma 5.29 (4), every literal alloc($x$) occurs within a subformula $\chi^{\text{fin}}$ of some formula elim$^\text{fin}$($M_1, M_2$), hence inside a formula of the form alloc($x$) $\lor (|h| < |U| - q \land |U| \geq r)$. Thus $C$ (hence $C'$) contains either $|h| < |U| - q$ or $|U| \geq r$, and necessarily, $q, r \leq N(\chi_2) \leq N(\chi)$. But $S$ has more than $N(\chi)$ unallocated elements, hence $S \models (|h| < |U| - q \land |U| \geq r)$. Therefore, $(\Omega, s^C, I, b) \models C'$, which contradicts our previous assumption.

Consequently, the initial problem boils down to testing whether $\phi'$ has a finite model. It is clear that $\phi'$ is BSR-compatible (since by definition all positive occurrences of alloc($x$) have been removed), hence by Lemma 5.28, it is sufficient to test whether $\tau(\phi') \land A(\phi')$ has a finite model. By Proposition 4.10, the formula $\tau(\phi') \land A(\phi')$ is equivalent to a formula in BSR($\text{FO}$). We have $N(\phi') \leq N(\forall y_1, \ldots, y_m \cdot \chi)$, hence, using Propositions 2.2, 4.10 and 5.36 we deduce as it is done in the proof of Theorem 6.11, that $\tau(\phi') \land A(\phi')$ has a finite model if it has a model $(\Omega, s, I)$, with $||\Omega|| = O(k \cdot \text{size($\phi'$)})$.

The algorithm is then defined as follows (see the proof of Theorem 6.11 for details). We guess an FO-structure $(\Omega, s, I)$ satisfying Heap such that $||\Omega|| = O(k \cdot \text{size($\phi'$)})$ and check in polynomial space that $(\Omega, s, I) \models \tau(\phi')$ (this is done as in Lemma 6.3, except that the test formula alloc($x$) are replaced by $\top$) and that $(\Omega, s, I) \models A(\phi')$ (using Proposition 6.2).

7 CONCLUSION

We have studied the decidability problem for SL formulae with quantifier prefix in the language $\exists^* \forall^*$, denoted as BSR($\text{SL}^k$), for finite and infinite universes, in the presence of uninterpreted predicate symbols. Although both problems were found to be undecidable, we identified two non-trivial subfragments for which the infinite and finite satisfiability are PSPACE-complete. These fragments are defined by restricting the polarity of occurrences of separating implications as well as the occurrence of universally quantified variables within the scope of separating implications. In both cases, the number of record fields $k$ may be part of the input, but we assume that the arity of the uninterpreted predicates is bounded by a constant. If the latter condition does not hold, then the provided algorithms run in exponential space, and the problem is NEXPTIME-complete. Note that the PSPACE-completeness results for BSR$^{\text{fin}}$(SL$^k$) and BSR$^{\text{inf}}$(SL$^k$) allow us to (re-)establish the PSPACE-membership of the satisfiability problem for quantifier-free formulae of SL$^k$, both in finite and infinite domains. Indeed, every
The Bernays-Schönfinkel-Ramsey Class of Separation Logic with Uninterpreted Predicates

quantifier-free formula \( \phi \) is sat-equivalent to a formula \( \phi \rightarrow \top \) that is both in \( \text{BSR}^{\text{fin}} (\text{SL}^k) \) and \( \text{BSR}^{\text{inf}} (\text{SL}^k) \), since the left-hand side of \( \rightarrow \) has neutral polarity.

Future work includes the implementation of an effective procedure for testing satisfiability of \( \text{BSR}(\text{SL}) \) formulæ in the above fragments. Since a non deterministic algorithm based on a guess-and-check approach is not practical, such a procedure could rely either on an encoding in QBF based on the finite model property derived in the present paper, or on some compact computational representations of boolean combinations of test formulæ. The bottleneck of the approach is certainly the computation of equivalent boolean combinations of test formulæ. To make the transformation more efficient, refined versions of Lemmas 5.16 and 5.20 could be derived, getting rid of some hypotheses such as E-completeness or A-completeness (as enforcing these hypotheses yield an exponential blow-up). Instead, the needed test formulæ could be added on demand, only if needed.

An extension of the presented results to formulæ containing inductively defined predicates (such as singly-linked lists) or interpreted predicates or functions (such as arithmetic symbols) will also be considered. This would allow us to extend existing approaches to test satisfiability of such formulæ \([5?]\) to formulæ containing negation.

ACKNOWLEDGMENTS

The authors wish to acknowledge the contributions of Stéphane Demri and Étienne Lozes to the insightful discussions during the early stages of this work.

REFERENCES


Mnacho Echenim, Radu Iosif, and Nicolas Peltier


