# The Effects of Adding Reachability Predicates in Propositional Separation Logic

**A. Mansutti**[1] S. Demri[1] E. Lozes [2]

[1]LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, Cachan, France

[2]I3S, Université Côte d'Azur, Nice, France

# Motivations

- Many tools support Separation Logic as an assertion language;
- Growing demand to consider more powerful extensions:
  - inductive predicates;
  - magic wand operator $\mathrel{-\!\!*}$;
  - closure under boolean connectives.

# Our work

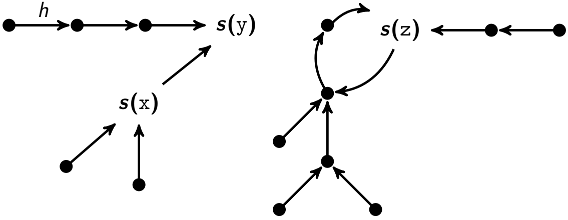
We study the satisfiability problem of $\mathtt{SL}(*, \mathrel{-\!\!*}, \mathtt{ls})$: Propositional Separation Logic enriched with the list segment predicate $\mathtt{ls}$.

# Memory states with one record field

Separation Logic is interpreted over **memory states** $(s, h)$ where:

- $s : \mathtt{VAR} \to \mathtt{LOC}$ is called store;
- $h : \mathtt{LOC} \to_{\mathsf{fin}} \mathtt{LOC}$ is called heap.

where $\mathtt{VAR} = \{\mathrm{x}, \mathrm{y}, \mathrm{z}, \dots\}$ set of (program) variables;
$\mathtt{LOC}$ set of locations (typically $\mathtt{LOC} \cong \mathbb{N} \cong \mathtt{VAR}$).



# Propositional Separation Logic $\mathtt{SL}(*, \mathrel{-\!\!*})$

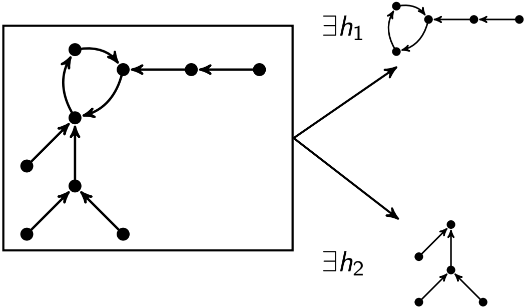
$$\varphi := \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathtt{x} = \mathtt{y} \mid \mathtt{emp} \mid \mathtt{x} \hookrightarrow \mathtt{y} \mid \varphi_1 * \varphi_2 \mid \varphi_1 \mathrel{-\!\!*} \varphi_2$$

## Semantics

- standard for $\wedge$ and $\neg$;
- $(s, h) \models \mathtt{x} = \mathtt{y} \iff s(\mathtt{x}) = s(\mathtt{y})$
- $(s, h) \models \mathtt{emp} \iff \mathrm{dom}(h) = \emptyset$
- $(s, h) \models \mathtt{x} \hookrightarrow \mathtt{y} \iff h(s(\mathtt{x})) = s(\mathtt{y})$

# Separating conjunction ($*$)

$(s, h) \models \varphi_1 * \varphi_2$ if and only if



$\exists h_1$ $(s, h_1) \models \varphi_1$

and

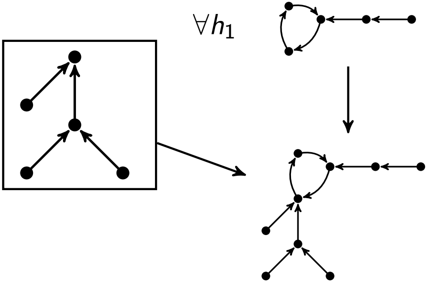
$\exists h_2$ $(s, h_2) \models \varphi_2$

There is a way to split the heap into two so that, together with the store, one part satisfies $\varphi_1$ and the other satisfies $\varphi_2$.

# Separating implication ($-\!\!*$)

$(s, h) \models \varphi_1 -\!\!* \varphi_2$ if and only if

$\forall h_1$



$$\mathrm{dom}(h) \cap \mathrm{dom}(h_1) = \emptyset$$
$$(s, h_1) \models \varphi_1$$
$$\Downarrow$$
$$(s, h + h_1) \models \varphi_2$$

Whenever a (disjoint) heap that, together with the store, satisfies $\varphi_1$ is added, the resulting memory state satisfies $\varphi_2$.

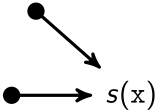
# $\mathtt{SL}(*, \mathrel{-\!\!*})$ + list segment predicate ($\mathtt{ls}$)

$(s, h) \models \mathtt{ls}(\mathtt{x}, \mathtt{y})$ if and only if

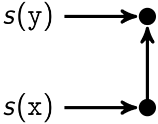
$$s(\mathtt{x}) \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow s(\mathtt{y})$$

$s(\mathtt{x})$ reaches $s(\mathtt{y})$ and all elements in $\mathrm{dom}(h)$ are necessary for this to hold.

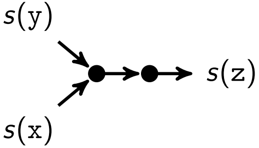
# Expressible properties in $\mathtt{SL}(*, -\!*, \mathtt{ls})$



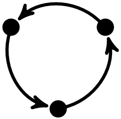

Number of predecessors




Next-points




Common paths



Loops

# Decidable status of related logics

First-order $\mathtt{SL}(\cdot \hookrightarrow (\cdot, \cdot))$

First-order $\mathtt{SL}(\mathrel{-\!\!*})$

undecidable

decidable

First-order $\mathtt{SL}(*)$
$\textsc{tower}$-C.

Prop. $\mathtt{SL}(*, \mathrel{-\!\!*})$
$\textsc{PSpace}$-C.

Symbolic Heaps
$\textsc{PTime}$

# Main results

- The satisfiability problem for $\mathtt{SL}(*, -\!\!*, \mathtt{ls})$ is undecidable.
- Several variants of $\mathtt{SL}(*, -\!\!*, \mathtt{ls})$ are also concluded undecidable.
- The satisfiability problem for $\mathtt{SL}(*, \mathtt{ls})$ (i.e. $\mathtt{SL}(*, -\!\!*, \mathtt{ls})$ without $-\!\!*$) is $\mathrm{PS{\scriptstyle PACE}}$-complete.
- The satisfiability problem for Boolean combinations of formulae in $\mathtt{SL}(*, \mathtt{ls}) \cup \mathtt{SL}(*, -\!\!*)$ is $\mathrm{PS{\scriptstyle PACE}}$-complete.

# Decidability status of $\mathtt{SL}(*, -\!\!*, \mathtt{ls})$

First-order $\mathtt{SL}(\cdot \hookrightarrow (\cdot,\cdot))$

First-order $\mathtt{SL}(-\!\!*)$

$\boxed{\textbf{Prop. } \mathtt{SL}(*, -\!\!*, \mathtt{ls})}$

undecidable

decidable

First-order $\mathtt{SL}(*)$
TOWER-C.

Prop. $\mathtt{SL}(*, -\!\!*)$
PSPACE-C.

Symbolic Heaps
PTIME

# Reduction of First-order $\mathtt{SL}(\multimap)$ to $\mathtt{SL}(*, \multimap, \mathtt{ls})$

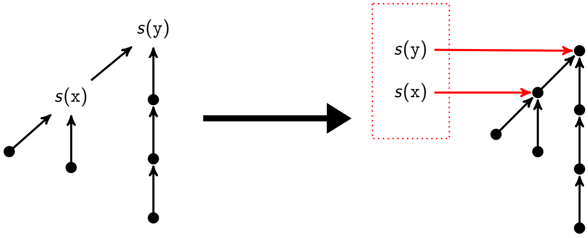
- We consider the first-order extension of SL($\multimap$)

$$(s, h) \models \forall \mathtt{x}.\varphi \iff \text{for all } \ell \in \mathtt{LOC}, (s[\mathtt{x} \leftarrow \ell], h) \models \varphi$$

- The satisfiability problem for First-order $\mathtt{SL}(\multimap)$ is undecidable. [IC, 2012].

- Idea for the translation: use the heap to mimic the store.

# Heaps simulate stores




- Given $\mathtt{V} \subseteq_{\mathsf{fin}} \mathtt{VAR}$, take $s|_{\mathtt{V}} + h : \mathtt{VAR} + \mathtt{LOC} \to_{\mathsf{fin}} \mathtt{LOC}$ and translate it inside the heap domain $[\mathtt{LOC} \to_{\mathsf{fin}} \mathtt{LOC}]$;

- A finite set of locations is used to simulate a finite portion of the store, effectively splitting the domain $\mathtt{LOC}$.

# Expressive power of SL($*$,$-\!\!*$,$\mathtt{ls}$)

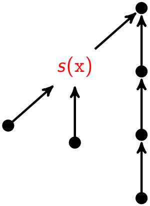
- $\mathtt{size} \geq \beta \iff \mathrm{dom}(h)$ has at least $\beta$ locations
- $\mathtt{alloc}(\mathtt{x}) \iff s(\mathtt{x}) \longrightarrow \bullet$
- $\mathtt{alloc}^{-1}(\mathtt{x}) \iff \bullet \longrightarrow s(\mathtt{x})$
- $n(\mathtt{x}) = n(\mathtt{y}) \iff s(\mathtt{x}) \longrightarrow \bullet \longleftarrow s(\mathtt{y})$
- $n(\mathtt{x}) \hookrightarrow n(\mathtt{y}) \iff s(\mathtt{x}) \rightarrow \bullet \longrightarrow \bullet \leftarrow s(\mathtt{y})$

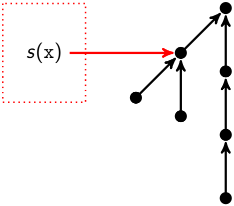
# Some bits of the translation

- $\mathtt{translation}_{\mathtt{V}}(\mathtt{x} = \mathtt{y}) \stackrel{\text{def}}{=} n(\mathtt{x}) = n(\mathtt{y})$;
- $\mathtt{translation}_{\mathtt{V}}(\mathtt{x} \hookrightarrow \mathtt{y}) \stackrel{\text{def}}{=} n(\mathtt{x}) \hookrightarrow n(\mathtt{y})$.

# Universal quantifier – $\forall \mathtt{x}.\varphi$

$$(\mathtt{alloc}(\mathtt{x}) \wedge \mathtt{size} = 1) \mathrel{-\!\!*} (\mathtt{safe}(\mathtt{V}) \implies \mathtt{translation}_{\mathtt{V}}(\varphi))$$



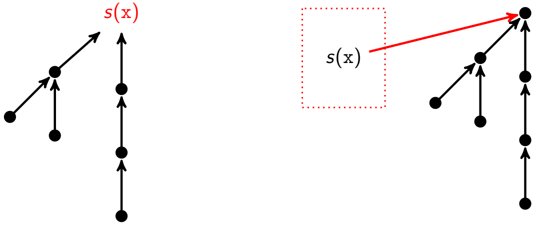

Where $\mathtt{safe}(\mathtt{V})$ states the sanity conditions to encode the store.

# Some bits of the translation

- $\texttt{translation}_{\texttt{V}}(\texttt{x} = \texttt{y}) \stackrel{\text{def}}{=} n(\texttt{x}) = n(\texttt{y})$;
- $\texttt{translation}_{\texttt{V}}(\texttt{x} \hookrightarrow \texttt{y}) \stackrel{\text{def}}{=} n(\texttt{x}) \hookrightarrow n(\texttt{y})$.

# Universal quantifier – $\forall \texttt{x}.\varphi$

$$(\texttt{alloc}(\texttt{x}) \land \texttt{size} = 1) \mathbin{-\!\!*} (\texttt{safe}(\texttt{V}) \implies \texttt{translation}_{\texttt{V}}(\varphi))$$




Where $\texttt{safe}(\texttt{V})$ states the sanity conditions to encode the store.

# Equisatisfiability

The translation of $\varphi \mathrel{-\!\!*} \psi$ requires the introduction of a copy $\overline{\mathtt{x}}$ for every variable $\mathtt{x}$ occurring in the formula.

## Theorem

Let $\varphi$ be a closed formula with variables in $\{\mathtt{x}_1, \ldots, \mathtt{x}_q\}$ and let $\mathtt{V} = \{\mathtt{x}_1, \ldots, \mathtt{x}_q, \overline{\mathtt{x}_1}, \ldots, \overline{\mathtt{x}_q}\}$.

$$\varphi \text{ is satisfiable}$$

$$\Updownarrow$$

$$\neg\mathtt{alloc}(\mathtt{V}) \land \mathtt{safe}(\mathtt{V}) \land \mathtt{translation}_{\mathtt{V}}(\varphi) \text{ is satisfiable.}$$

# Undecidability results

The following fragments have undecidable satisfiability problem:

- $\mathsf{SL}(*, \mathrel{-\!\!*}) + n(\mathtt{x}) = n(\mathtt{y})$, $n(\mathtt{x}) \hookrightarrow n(\mathtt{y})$ and $\mathtt{alloc}^{-1}(\mathtt{x})$;
- $\mathsf{SL}(*, \mathrel{-\!\!*}) + \mathtt{reach}(\mathtt{x}, \mathtt{y}) = 2$ and $\mathtt{reach}(\mathtt{x}, \mathtt{y}) = 3$;
- $\mathsf{SL}(*, \mathrel{-\!\!*}, \mathtt{ls})$.

# Complexity of $\mathtt{SL}(*, \mathtt{ls})$

First-order $\mathtt{SL}(\cdot \hookrightarrow (\cdot, \cdot))$

First-order $\mathtt{SL}(\mathrel{-\!\!*})$

undecidable

decidable

$\boxed{\textbf{Prop. } \mathtt{SL}(*, \mathtt{ls})}$

PSPACE-C.

First-order $\mathtt{SL}(*)$

TOWER-C.

$\boxed{\mathtt{SL}(*, \mathrel{-\!\!*})}$

PSPACE-C.

Symbolic Heaps

PTIME

# Deciding $\mathtt{SL}(*, \mathtt{ls})$ thanks to the test formulae approach

- Study basic properties that can be expressed in $\mathtt{SL}(*, \mathtt{ls})$;
- Define (test) formulae for these properties;
- $*$ elimination: show that each formula of $\mathtt{SL}(*, \mathtt{ls})$ is captured by a boolean combination of test formulae;
- Show a small-model property for the logic of test formulae.

# Deciding $\mathtt{SL}(*, \mathtt{ls})$ thanks to the test formulae approach

- Study basic properties that can be expressed in $\mathtt{SL}(*, \mathtt{ls})$;
- D
- $*$ captured
  by

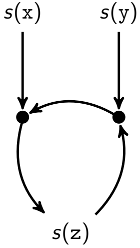
For $\mathtt{SL}(*, -\!\!*)$: each formula is equivalent to a boolean combinations of formulae of the form

$$x = y,\ \mathtt{alloc}(x),\ x \hookrightarrow y,\ \mathtt{size} \geq \beta.$$

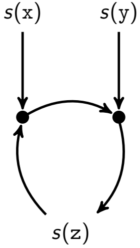
- Show a small-model property for the logic of test formulae.

# $\mathtt{SL}(*, \mathtt{ls})$: Searching for Test Formulae

For example, we can show that



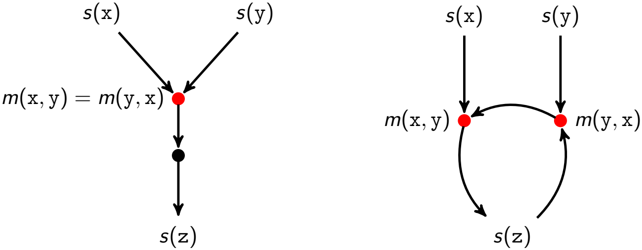

and




can be distinguished in the logic.

# Meet–points

To capture this and other properties, we introduce meet–points.




# Interpretation

$[\![m(\mathrm{x},\mathrm{y})]\!]_{s,h}$ is the first location reachable from $s(\mathrm{x})$ that is also reachable from $s(\mathrm{y})$.

# Test formulae

Given $\{\mathtt{x}_1, \ldots, \mathtt{x}_q\} \subseteq \mathtt{VAR}$ and $\alpha \in \mathbb{N}^+$, we define $\mathsf{Test}(q, \alpha)$ as the set of following test formulae:

$$v = v' \quad v \hookrightarrow v' \quad \mathtt{alloc}(v) \quad \mathtt{sees}_q(v, v') \geq \beta + 1 \quad \mathtt{sizeR}_q \geq \beta,$$

where $\beta \in [1, \alpha]$ and $v, v'$ are variables $\mathtt{x}_i$ or meet–points $m(\mathtt{x}_i, \mathtt{x}_j)$, with $i, j \in [1, q]$.

# Indistinguishability Relation

$(s, h) \approx_\alpha^q (s', h')$ whenever $(s, h)$ and $(s', h')$ satisfy the same test formulae of $\mathsf{Test}(q, \alpha)$.

# Test formulae: $\mathtt{sees}_q$

$$(s, h) \models \mathtt{sees}_q(v, v') \geq \beta + 1$$

if and only if there is a path path from $[\![v]\!]_{s,h}$ to $[\![v']\!]_{s,h}$

- of length at least $\beta + 1$
- that does not traverse labelled locations

$$[\![v]\!]_{s,h} \longrightarrow \underbrace{\bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet}_{\text{not labelled}} \longrightarrow [\![v']\!]_{s,h}$$

where $[\![\mathtt{x}]\!]_{s,h} = s(\mathtt{x})$.

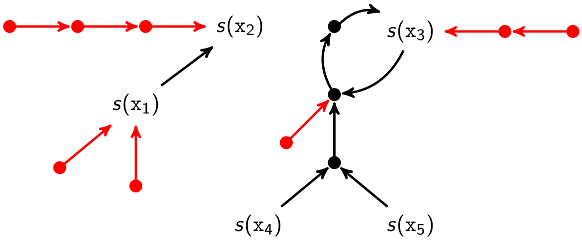
# Test formulae: $\mathtt{sizeR}_q$

$$(s, h) \models \mathtt{sizeR}_q \geq \beta$$

if and only if the number of locations in $\mathrm{dom}(h)$ that

- are not corresponding to variables
- are not in the path between two variables

is greater or equal than $\beta$



## Expressive power characterisation

Let $\varphi$ with variables $\mathtt{x}_1, \ldots, \mathtt{x}_q$ and let $\alpha \geq |\varphi|$.

- If $(s, h) \approx_{\alpha}^{q} (s', h')$ then we have $(s, h) \models \varphi$ iff $(s', h') \models \varphi$.
- $\varphi$ is logically equivalent to a Boolean combination of test formulae from $\mathsf{Test}(q, \alpha)$.

## Small model property

Let $\varphi$ be a satisfiable $\mathtt{SL}(*, \mathtt{ls})$ formula built over $\mathtt{x}_1, \ldots, \mathtt{x}_q$.
There is $(s, h)$ such that $(s, h) \models \varphi$ and

$$\mathrm{card}(\mathrm{dom}(h)) \leq \mathrm{card}(\mathsf{Test}(q, |\varphi|))$$

## Complexity upper bound

The satisfiability problem for $\mathtt{SL}(*, \mathtt{ls})$ is $\mathrm{PS{\scriptsize PACE}}$-complete.

## Recap

- SL($*$, $-\!\!*$, `ls`) admits an undecidable satisfiability problem, but
- if `ls` is not in the scope of $-\!\!*$ then the problem is decidable
- and it is PSPACE-complete if $-\!\!*$ is removed.

## Ongoing work

- SL($*$, $-\!\!*$, `ls`) where `ls` does not occur on the right side of $-\!\!*$ (PSPACE-complete)
- SL($-\!\!*$) $+$ $n(x) = n(y)$, $n(x) \hookrightarrow n(y)$ and $\texttt{alloc}^{-1}(x)$ (undecidable)

## Future Work

- Decidable fragments with `ls` in the scope of $-\!\!*$;
- Generalisation of the test formulae approach.