Internal calculi for Separation Logics

Stéphane Demri[1] Étienne Lozes[2] **Alessio Mansutti**[1]

September 9, 2020

[1]LSV, CNRS, ENS Paris-Saclay

[2]I3S, Université Côte d'Azur

Separation Logic

'99 Logic of Bunched Implication (BI) [P. O'Hearn, D. Pym]

'02 Separation Logic [P. O'Hearn, D. Pym, J. Reynolds]

- Logic for **modular** verification of pointer programs.
- Used in state-of-the-art, industrial tools:
  - Infer (Facebook)
  - Slayer (Microsoft)
- "Why Separation Logic Works" ['18 - D. Pym et al.]

Separation Logic, with apples

‘99 Logic of Bunched Implication (BI) [P. O’Hearn, D. Pym]

‘02 Separation Logic [P. O’Hearn, D. Pym, J. Reynolds]

Multiplicative connectives (from BI):

 $\models \varphi * \psi$ *iff*  can be split into  and  s.t.
 $\models \varphi$ and  $\models \psi$.

 $\models \varphi \mathrel{-\!\!*} \psi$ *iff* for every  mergeable with ,
if  $\models \varphi$ then  $\models \psi$

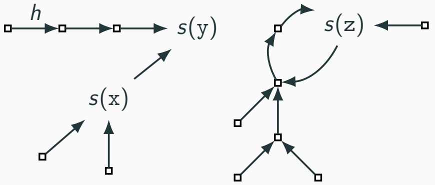
Problem: How to deal with $*$ and $\mathrel{-\!\!*}$, on concrete models and in the context of Hilbert-style axiomatisations.

Modelling the memory

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

- **store**, $s : \mathrm{VAR} \to \mathbb{N}$
- **heap**, $h : \mathbb{N} \to_{\mathsf{fin}} \mathbb{N}$

where $\mathrm{VAR} = \{\mathtt{x}, \mathtt{y}, \mathtt{z}, \dots\}$ set of variables,
$\mathbb{N}$ represents the set of addresses.



here, $h(s(\mathtt{x})) = s(\mathtt{y})$

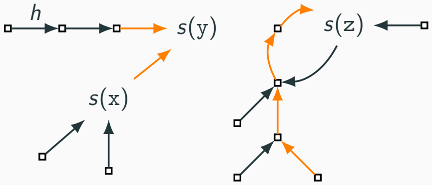
- Disjoint heaps ($h_1 \perp h_2$): $\mathrm{dom}(h_1) \cap \mathrm{dom}(h_2) = \emptyset$
- Union of disjoint heaps ($h_1 + h_2$): union of partial functions.

Modelling the memory

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

- **store**, $s : \mathrm{VAR} \to \mathbb{N}$
- **heap**, $h : \mathbb{N} \to_{\mathsf{fin}} \mathbb{N}$

where $\mathrm{VAR} = \{\mathtt{x}, \mathtt{y}, \mathtt{z}, \dots\}$ set of variables,
$\mathbb{N}$ represents the set of addresses.

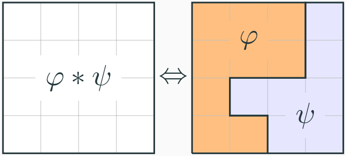


here, $h(s(\mathtt{x})) = s(\mathtt{y})$

- Disjoint heaps ($h_1 \perp h_2$): $\mathrm{dom}(h_1) \cap \mathrm{dom}(h_2) = \emptyset$
- Union of disjoint heaps ($h_1 + h_2$): union of partial functions.

The separating conjunction ($*$)

$$(s, h) \models \varphi * \psi$$



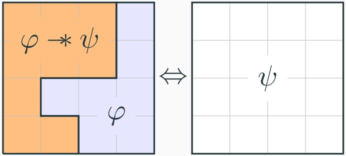
Semantics:

There are two heaps h_1 and h_2 s.t.

- $h_1 \perp h_2$ and $h = h_1 + h_2$,
- $(s, h_1) \models \varphi$,
- $(s, h_2) \models \psi$.

The separating implication ($-\!\!*$)

$$(s, h) \models \varphi -\!\!* \psi$$



Semantics:

For every heap h',

if $h' \perp h$ and $(s, h') \models \varphi$,

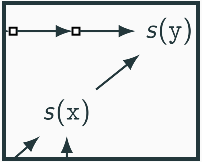
then $(s, h + h') \models \psi$.

Note: $*$ and $-\!\!*$ are adjoint operators:

$$\varphi * \psi \models \gamma \quad \text{if and only if} \quad \varphi \models \psi -\!\!* \gamma.$$

First-order Separation Logic

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



$(s, h) \models \mathtt{emp}$ *iff* $\mathrm{dom}(h) = \emptyset$,

$(s, h) \models \mathtt{x} = \mathtt{y}$ *iff* $s(\mathtt{x}) = s(\mathtt{y})$,

$(s, h) \models \mathtt{x} \hookrightarrow \mathtt{y}$ *iff* $s(\mathtt{x}) \in \mathrm{dom}(h)$ and $h(s(\mathtt{x})) = s(\mathtt{y})$,

$(s, h) \models \exists \mathtt{x}\, \varphi$ *iff* there is $n \in \mathbb{N}$ s.t. $(s[\mathtt{x} \leftarrow n], h) \models \varphi$.

Satisfiability problem: some complexity results.

Fsttcs'01 Quantifier-free $\mathtt{SL}$ ($0\mathtt{SL}$) is $\mathrm{PS}\textsc{pace}$-complete.
[C. Calcagno, P.W. O'Hearn, H. Yang]

Tocl'15 $\mathtt{SL}$ with two quantified variables ($2\mathtt{SL}$) is undecidable.
[S. Demri, M. Deters]

Fossacs'18 $0\mathtt{SL}$ + reachability predicates is undecidable.
Without $-\!\!*$ it is $\mathrm{PS}\textsc{pace}$-complete.
[S. Demri, E. Lozes, A. Mansutti]

Fsttcs'18 $1\mathtt{SL}$ + restricted reachability predicate is $\mathrm{PS}\textsc{pace}$-c.
Weakening restrictions makes it $\mathrm{T}\textsc{ower}$-hard.

Satisfiability $\approx$ Validity $\approx$ Entailment $\approx$ Model checking

Let $\varphi \mathbin{-\!\circledast} \psi \stackrel{\text{def}}{=} \neg(\varphi \mathbin{-\!\!*} \neg\psi)$.

$(s,h) \models \varphi \mathbin{-\!\circledast} \psi \quad \textit{iff} \quad \exists h' \text{ s.t. } h' \perp h,\ (s,h') \models \varphi \text{ and } (s, h{+}h') \models \psi$

Satisfiability to validity

$$\models \texttt{emp} \Rightarrow \exists \mathtt{x}_1 \ldots \exists \mathtt{x}_n(\varphi \mathbin{-\!\circledast} \top) \quad \textit{iff} \quad \exists\, s\ \exists\, h \text{ s.t. } (s,h) \models \varphi$$

where $\{\mathtt{x}_1, \ldots, \mathtt{x}_n\} = \mathsf{fv}(\varphi)$.

- Reduction can be done also without quantification, but requires exponentially many queries of validity (w.r.t. $\mathsf{fv}(\varphi)$).
- Satisfiability to validity works also for $\mathtt{0SL}$.

Undecidability implies non-axiomatisability

Validity R.E. $\rightarrow$ Satisfiability R.E. $\rightarrow$ Unvalidity R.E.
$\rightarrow$ Validity decidable.

Tocl'15: ~~$\mathtt{SL}$ with two quantified variables ($\mathtt{2SL}$) is undecidable.~~

Fossacs'18: ~~$\mathtt{0SL}$ + reachability predicates is undecidable.~~

This Talk: Hilbert-style axiomatisation for $\mathtt{SL}$s (on memory states)

- Quantifier-free Separation Logic ($\mathtt{0SL}$);
- $\mathtt{SL}$ without $-\!\!*$ and with a (novel) guarded form of quantification that can express reachability predicates.

Calculi for Bunched Implication / Separation Logics

Fsttcs'06 Hilbert-style axiomatisation of Boolean BI
[D. Galmiche, D. Larchey-Wending]

Popl'14 Axiomatisation of an hybrid version of Boolean BI
and axiomatisation of abstract separation logics
[J. Brotherston, J. Villard]

Tocl'18 Sequent calculi for abstract separation logics
[Z. Hou, R. Clouston, R. Goré, A. Tiu.]

Fossacs'18 Modular tableaux calculi for Boolean BI
[S. Docherty, D. Pym.]

On axiomatising $\mathtt{0SL}$, internally

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

Methodology:

1A. Model theoretical analysis of $\mathtt{0SL}$ (Lozes'04);

(EF-games / simulation arguments)

1B. Definition of a "normal form" for formulae of $\mathtt{0SL}$;

(Gaifman-like locality theorem for $\mathtt{0SL}$)

2. Axiomatisation specific to the formulae in this normal form;

3. Add axioms & rules to put every formula in normal form.

(similar to *reduction axioms* in dynamic epistemic logic)

What can $\mathtt{OSL}$ express?

- The heap has size at least β:

$$\mathtt{size} \geq \beta \stackrel{\text{def}}{=} \underbrace{\neg\mathtt{emp} * \ldots * \neg\mathtt{emp}}_{\beta \text{ times}}$$

- $\mathtt{x}$ corresponds to a location in the domain of the heap:

$$\mathtt{alloc}(\mathtt{x}) \stackrel{\text{def}}{=} \neg\big(\mathtt{x} \hookrightarrow \mathtt{x} \mathrel{-\!\!\circledast} \top\big)$$

Let $\mathtt{X} \subseteq_{\text{fin}} \text{VAR}$ and $\alpha \in \mathbb{N}$. We define the set of **core formulae**:

$$\mathtt{Core}(\mathtt{X}, \alpha) \stackrel{\text{def}}{=} \{\mathtt{x} = \mathtt{y},\ \mathtt{x} \hookrightarrow \mathtt{y},\ \mathtt{alloc}(\mathtt{x}),\ \mathtt{size} \geq \beta \mid \mathtt{x}, \mathtt{y} \in \mathtt{X}, \beta \in [0, \alpha]\}.$$

An indistinguishability relation for $\mathtt{OSL}$

$$(s, h) \approx_{\alpha}^{\mathtt{X}} (s', h') \quad \textit{iff} \ \forall \varphi \in \mathtt{Core}(\mathtt{X}, \alpha),\ (s, h) \models \varphi \Leftrightarrow (s', h') \models \varphi.$$

An indistinguishability relation for $\mathtt{OSL}$

$(s, h) \approx^{\mathtt{X}}_{\alpha} (s', h')$ *iff* $\forall \varphi \in \mathtt{Core}(\mathtt{X}, \alpha)$, $(s, h) \models \varphi \Leftrightarrow (s', h') \models \varphi$.

A simulation Lemma for the operator $*$

Let $(s, h) \approx^{\mathtt{X}}_{\alpha} (s', h')$.
$\forall \alpha_1, \alpha_2$ satisfying $\alpha_1 + \alpha_2 = \alpha$, $\forall h_1, h_2$ satisfying $h_1 + h_2 = h$,
$\exists h'_1, h'_2$ s.t. $h'_1 + h'_2 = h'$, $(s, h_1) \approx^{\mathtt{X}}_{\alpha_1} (s', h'_1)$ and $(s, h_2) \approx^{\mathtt{X}}_{\alpha_2} (s', h'_2)$.

Similar lemma for $-\!\!*$.

An indistinguishability relation for $\mathtt{OSL}$

$(s, h) \approx^{\mathtt{X}}_{\alpha} (s', h')$ *iff* $\forall \varphi \in \mathtt{Core}(\mathtt{X}, \alpha)$, $(s, h) \models \varphi \Leftrightarrow (s', h') \models \varphi$.

A simulation Lemma for the operator $*$

Let $(s, h) \approx^{\mathtt{X}}_{\alpha} (s', h')$.

$\forall \alpha_1, \alpha_2$ satisfying $\alpha_1 + \alpha_2 = \alpha$, $\forall h_1, h_2$ satisfying $h_1 + h_2 = h$,

$\exists h'_1, h'_2$ s.t. $h'_1 + h'_2 = h'$, $(s, h_1) \approx^{\mathtt{X}}_{\alpha_1} (s', h'_1)$ and $(s, h_2) \approx^{\mathtt{X}}_{\alpha_2} (s', h'_2)$.

This lemma hides a Spoiler/Duplicator EF-games for $\mathtt{OSL}$, and shows the existence of a winning strategy for Duplicator.

For every move of Spoiler, the Duplicator has a winning answer.

An indistinguishability relation for $\mathtt{OSL}$

$(s,h) \approx_{\alpha}^{\mathtt{X}} (s',h')$ *iff* $\forall \varphi \in \mathtt{Core}(\mathtt{X},\alpha)$, $(s,h) \models \varphi \Leftrightarrow (s',h') \models \varphi$.

A simulation Lemma for the operator $*$

Let $(s,h) \approx_{\alpha}^{\mathtt{X}} (s',h')$.
$\forall \alpha_1, \alpha_2$ satisfying $\alpha_1 + \alpha_2 = \alpha$, $\forall h_1, h_2$ satisfying $h_1 + h_2 = h$,
$\exists h'_1, h'_2$ s.t. $h'_1 + h'_2 = h'$, $(s,h_1) \approx_{\alpha_1}^{\mathtt{X}} (s',h'_1)$ and $(s,h_2) \approx_{\alpha_2}^{\mathtt{X}} (s',h'_2)$.

Similar lemma for $-\!\!*$.

A "Gaifman locality theorem" for $\mathtt{OSL}$

Every formula φ in $\mathtt{OSL}$ is logically equivalent to a Boolean combination of core formulae from $\mathtt{Core}(\mathrm{vars}(\varphi), \mathtt{size}(\varphi))$.

$$\mathtt{Core}(\mathtt{X},\alpha) \stackrel{\text{def}}{=} \{\mathtt{x} = \mathtt{y},\ \mathtt{x} \hookrightarrow \mathtt{y},\ \mathtt{alloc}(\mathtt{x}),\ \mathtt{size} \geq \beta \mid \mathtt{x},\mathtt{y} \in \mathtt{X}, \beta \in [0,\alpha]\}.$$

Normalising connectives & reasoning on core formulae

Normalisation of $*$ and $\mathrel{-\!\!*}$

$\vdash \psi_1 * \psi_2 \Leftrightarrow \psi_3$

$\vdash \psi_4 \mathrel{-\!\!*} \psi_5 \Leftrightarrow \psi_6$

Completeness for core formulae

$$\frac{\vdash \ \varphi \Leftrightarrow \psi \qquad\qquad \vdash \psi}{\vdash \varphi}$$

where φ in $\texttt{SL}$, and ψ_i, ψ are in $\bigcup_{\texttt{X},\alpha} \mathbf{Bool}(\texttt{Core}(\texttt{X}, \alpha))$.

From a simple calculus for Core formulae...

(PC) propositional calculus;

(R) $\mathtt{x} = \mathtt{x}$

(S) $\varphi \wedge \mathtt{x} = \mathtt{y} \Rightarrow \varphi[\mathtt{y} \leftarrow \mathtt{x}]$

(A) $\mathtt{x} \hookrightarrow \mathtt{y} \Rightarrow \mathtt{alloc}(\mathtt{x})$

(F) $\mathtt{x} \hookrightarrow \mathtt{y} \wedge \mathtt{x} \hookrightarrow \mathtt{z} \Rightarrow \mathtt{y} = \mathtt{z}$

(H1) $\mathtt{size} \geq \beta{+}1 \Rightarrow \mathtt{size} \geq \beta$

(H2) $\bigwedge\limits_{\mathtt{x} \in \mathtt{X}} (\mathtt{alloc}(\mathtt{x}) \wedge \bigwedge\limits_{\mathtt{y} \in \mathtt{X} \setminus \{\mathtt{x}\}} \mathtt{x} \neq \mathtt{y}) \Rightarrow \mathtt{size} \geq \mathrm{card}(\mathtt{X})$, where $\mathtt{X} \subseteq_{\mathsf{fin}}$ VAR.

$\mathtt{CoreTypes}(\mathtt{X}, \alpha)$: set of *complete*[1] conjunctions of formulae in $\mathtt{Core}(\mathtt{X}, \mathrm{card}(\mathtt{X}) + \alpha)$.

Lemma
Let $\varphi \in \mathtt{CoreTypes}(\mathtt{X}, \alpha)$. We have, $\models \neg\varphi$ *iff* $\vdash \neg\varphi$.

[1] Every $\varphi \in \mathtt{Core}(\mathtt{X}, \mathrm{card}(\mathtt{X}) + \alpha)$ appears in a literal of the conjunction.

From a simple calculus for Core formulae...

(PC) propositional calculus;

(R) $\mathtt{x} = \mathtt{x}$

(S) $\varphi \wedge \mathtt{x} = \mathtt{y} \Rightarrow \varphi[\mathtt{y} \leftarrow \mathtt{x}]$

(A) $\mathtt{x} \hookrightarrow \mathtt{y} \Rightarrow \mathtt{alloc}(\mathtt{x})$

(F) $\mathtt{x} \hookrightarrow \mathtt{y} \wedge \mathtt{x} \hookrightarrow \mathtt{z} \Rightarrow \mathtt{y} = \mathtt{z}$

(H1) $\mathtt{size} \geq \beta{+}1 \Rightarrow \mathtt{size} \geq \beta$

(H2) $\bigwedge_{\mathtt{x} \in \mathtt{X}} (\mathtt{alloc}(\mathtt{x}) \wedge \bigwedge_{\mathtt{y} \in \mathtt{X} \setminus \{\mathtt{x}\}} \mathtt{x} \neq \mathtt{y}) \Rightarrow \mathtt{size} \geq \mathrm{card}(\mathtt{X})$, where $\mathtt{X} \subseteq_{\mathsf{fin}} \mathsf{VAR}$.

$\mathtt{CoreTypes}(\mathtt{X}, \alpha)$: set of *complete*[1] conjunctions of formulae in $\mathtt{Core}(\mathtt{X}, \mathrm{card}(\mathtt{X}) + \alpha)$.

Lemma
A Boolean combination of core formulae, $\models \varphi$ *iff* $\vdash \varphi$.

[1] Every $\varphi \in \mathtt{Core}(\mathtt{X}, \mathrm{card}(\mathtt{X}) + \alpha)$ appears in a literal of the conjunction.

...to a sound and complete proof system for $\mathtt{OSL}$

(M) $\mathtt{alloc}(\mathtt{x}) * \top \Rightarrow \mathtt{alloc}(\mathtt{x})$

(N) $\neg\mathtt{alloc}(\mathtt{x}) * \neg\mathtt{alloc}(\mathtt{x}) \Rightarrow \neg\mathtt{alloc}(\mathtt{x})$

(I) $\mathtt{alloc}(\mathtt{x}) \Rightarrow (\mathtt{alloc}(\mathtt{x}) \wedge \mathtt{size} = 1) * \top$

$$\frac{\varphi \Rightarrow \gamma}{\varphi * \psi \Rightarrow \gamma * \psi}$$

Lemma
$\forall \varphi, \psi {\in} \mathsf{Bool}(\mathtt{Core}(\mathtt{X}, \alpha))\ \exists \gamma {\in} \mathsf{Bool}(\mathtt{Core}(\mathtt{X}, 2\alpha))$ s.t. $\vdash \varphi * \psi \Leftrightarrow \gamma$.

(P) $\neg\mathtt{alloc}(\mathtt{x}) \Rightarrow ((\mathtt{x} \hookrightarrow \mathtt{y} \wedge \mathtt{size} = 1) \mathrel{-\!\circledast} \top)$

$$\frac{\varphi * \psi \Rightarrow \gamma}{\varphi \Rightarrow (\psi \mathrel{-\!\!*} \gamma)}$$

Lemma
$\forall \varphi, \psi {\in} \mathsf{Bool}(\mathtt{Core}(\mathtt{X}, \alpha))\ \exists \gamma {\in} \mathsf{Bool}(\mathtt{Core}(\mathtt{X}, \alpha))$ s.t. $\vdash (\varphi \mathrel{-\!\circledast} \psi) \Leftrightarrow \gamma$.

A separation logic with path quantifiers

- We want to test our methodology on other $\mathtt{SL}$s,
- First-order quantification? Reachability predicates?
- Both extensions are undecidable, hence validity is not R.E.

We consider $\mathtt{0SL}$ + path quantifiers, w/o $-\!\!*$ (for decidability).

$$\varphi := \neg\varphi \mid \varphi_1 \land \varphi_2 \mid \mathtt{emp} \mid \mathtt{x}{=}\mathtt{y} \mid \mathtt{x}{\hookrightarrow}\mathtt{y} \mid \varphi_1 * \varphi_2 \mid \exists \mathtt{z}{:}\langle \mathtt{x}{\leadsto}\mathtt{y}\rangle\varphi$$

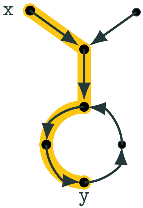
A separation logic with path quantifiers

$$(s,h) \models \exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle\ \varphi$$

iff

$$\exists \ell \in \square \text{ s.t. } (s[\mathtt{z} \leftarrow \ell], h) \models \varphi.$$

(the path must be of length at least 1 and minimal)



- $\exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle \top$ is the predicate $\mathtt{reach}^{+}(\mathtt{x},\mathtt{y})$,
- it can express the (standard) list-segment predicate ($\mathtt{ls}$),
- also cyclic structures, path of exponential length...

$$\exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle \big((\mathtt{reach}^{+}(\mathtt{x},\mathtt{z}) * \mathtt{reach}^{+}(\mathtt{z},\mathtt{z})) \land \varphi \big)$$

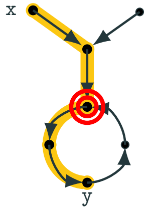
A separation logic with path quantifiers

$$(s,h) \models \exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle\ \varphi$$

iff

$$\exists \ell \in \boxed{} \text{ s.t. } (s[\mathtt{z} \leftarrow \ell], h) \models \varphi.$$

(the path must be of length at least 1 and minimal)



- $\exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle \top$ is the predicate $\mathtt{reach}^+(\mathtt{x}, \mathtt{y})$,
- it can express the (standard) list-segment predicate ($\mathtt{ls}$),
- also cyclic structures, path of exponential length...

$$\exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle \big((\mathtt{reach}^+(\mathtt{x}, \mathtt{z}) * \mathtt{reach}^+(\mathtt{z}, \mathtt{z})) \wedge \varphi\big)$$

We axiomatise $\mathsf{SL}(*, \exists{:}\rightsquigarrow)$ as done for $\mathtt{0}\mathsf{SL}$

I. With the help of simulations Lemmata for $*$ and $\exists{:}\rightsquigarrow$, we find the right set of core formulae $\mathtt{Core}(\mathtt{X}, \alpha)$.

II. We axiomatise the Boolean combination of core formulae.

III. We add axioms to treat $*$ and $\exists{:}\rightsquigarrow$, completing the system.

Normalisation of $*$ and $\exists{:}\rightsquigarrow$

$\vdash \exists \mathtt{z}{:}\langle \mathtt{x} \rightsquigarrow \mathtt{y} \rangle \psi_1 \Leftrightarrow \psi_2$

$\vdash \psi_4 * \psi_5 \Leftrightarrow \psi_6$

Completeness for core formulae

$$\frac{\vdash \varphi \Leftrightarrow \psi \qquad \vdash \psi}{\vdash \varphi}$$

From the normalisation, we also conclude that validity and satisfiability for $\mathsf{SL}(*, \exists{:}\rightsquigarrow)$ are $\mathrm{PS{\small PACE}}$-complete.

Recap

1. First axiomatisations of separation logics (on memory states),
   - quantifier-free $\mathtt{SL}$,
   - $\mathsf{SL}(*, \exists{:}\rightsquigarrow)$ (here introduced).
2. For program verification, $\exists{:}\rightsquigarrow$ is a natural form of quantification.
3. Satisfiability/validity of $\mathsf{SL}(*, \exists{:}\rightsquigarrow)$ found to be $\mathrm{PS{\small PACE}}$-complete.
4. The proof technique is quite reusable
   - Already used succesfully on two Modal Separation Logics [Jelia'19 - S. Demri, R. Fervari, A. Mansutti]