Decision procedures for Separation Logic

Alessio Mansutti

LSV, CNRS, E.N.S. Paris-Saclay, France

under the supervision of Stéphane Demri and Étienne Lozes

Hoare Logic for program analysis

In the 1960s, Floyd and Hoare introduced a fundamental technique for deductive verification: a logical system where **judgements** are of the form

$$\{\ \varphi\ \}\ \mathtt{P}\ \{\ \psi\ \};\ \text{read as:}$$

"Every model $\mathfrak{M}$ that satisfies φ, will satisfy ψ after being modified by the program $\mathtt{P}$*".*

A **model** is a mathematical structure that abstract the resources that the program uses. For example, it could be a set of variables with their content.

$$\{\ \mathtt{x} = n\ \}\ \mathtt{y} \leftarrow \mathtt{factorial}(\mathtt{x})\ \{\ \mathtt{y} = n!\ \}$$

A last ingredient are the **inference rules**, e.g.

$$\frac{\varphi \models \varphi' \qquad \{\ \varphi'\ \}\ \mathtt{P}\ \{\ \psi'\ \} \qquad \psi' \models \psi}{\{\ \varphi\ \}\ \mathtt{P}\ \{\ \psi\ \}}\ (\mathsf{ENT})$$

stating that judgements retain validity when considering stronger preconditions (φ) or weaker postconditions (ψ). $\varphi \models \varphi'$ is the logical **entailment**.

Why Separation Logic?

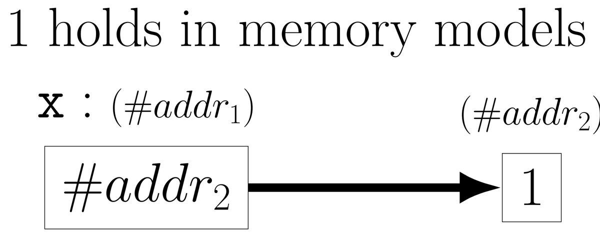
To **analyse large programs it is vital to reason locally** on the memory model. We would like:

$$\frac{\{\ \varphi\ \}\ \mathtt{P}\ \{\ \psi\ \}}{\{\ \varphi \land \chi\ \}\ \mathtt{P}\ \{\ \psi \land \chi\ \}}$$

but this rule is not valid when considering the standard heap/RAM memory, containing pointers:

$$\frac{\{\ \mathtt{x} \hookrightarrow 1\ \}\ {}^{*}\mathtt{x} \leftarrow 0\ \{\ \mathtt{x} \hookrightarrow 0\ \}}{\{\ \mathtt{x} \hookrightarrow 1 \land \mathtt{y} \hookrightarrow 1\ \}\ {}^{*}\mathtt{x} \leftarrow 0\ \{\ \mathtt{x} \hookrightarrow 0 \land \mathtt{y} \hookrightarrow 1\ \}}$$

does not hold whenever $\mathtt{x}$ and $\mathtt{y}$ are in aliasing.
Here, $\mathtt{x} \hookrightarrow 1$ holds in memory models such that:

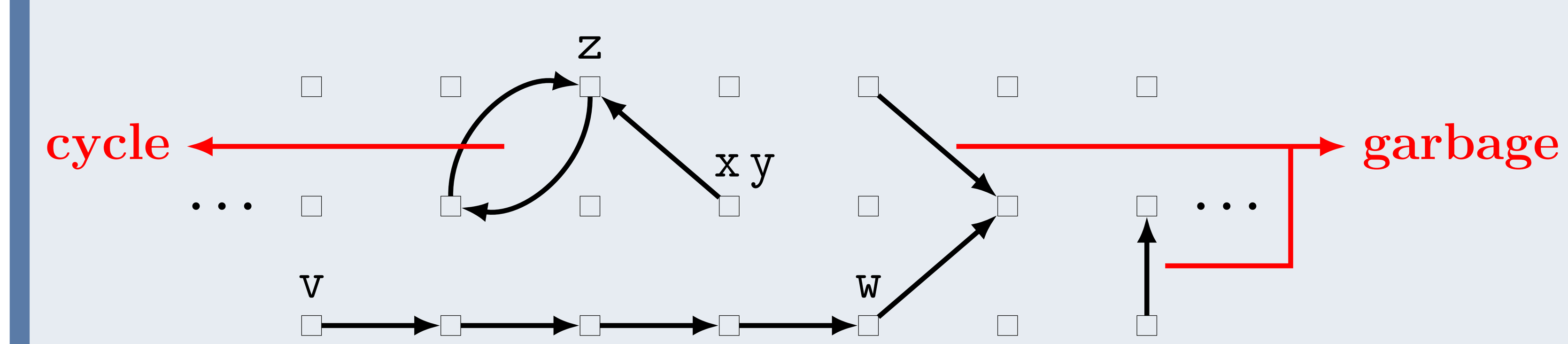


Separation Logic solves this problem elegantly, with its separating connectives.
This led to numerous tools using Separation Logic:

- Infer (Facebook)
- SLAyer (Microsoft)
- Verifast
- SeLoger

Separation Logic

(Propositional) Separation Logic (**SL**) – by J. Reynolds, P. O'Hearn et al. – reasons about programs with dynamic data structures. **Models** of SL are abstractions of the heap/RAM model:

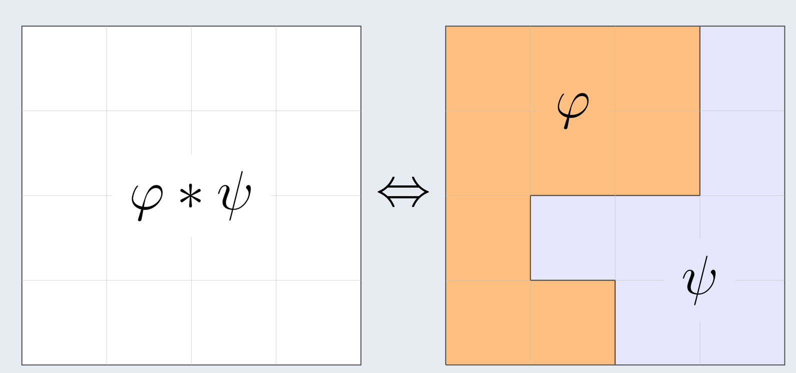


- infinite set of locations (□)
- infinite set of variables
- $\mathbf{s}$: variables $\mapsto$ locations
- $\mathbf{h}$: finite heap (□→□)
- a model is def. as $(\mathbf{s}, \mathbf{h})$.

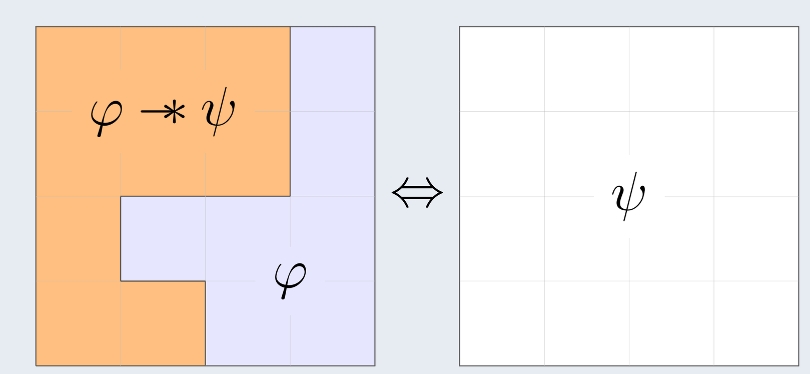
Separation Logic adds two new spatial connectives to reason **modularly** about the memory

$$\ast \qquad\qquad -\!\!\ast$$

$(\mathbf{s}, \mathbf{h}) \models \varphi \ast \psi$ iff the heap $\mathbf{h}$ can be partitioned into $\mathbf{h}_1$ and $\mathbf{h}_2$ so that $(\mathbf{s}, \mathbf{h}_1) \models \varphi$ and $(\mathbf{s}, \mathbf{h}_2) \models \psi$



$(\mathbf{s}, \mathbf{h}) \models \varphi \mathrel{-\!\!\ast} \psi$ iff for every $\mathbf{h}_1$ disjoint from $\mathbf{h}$, if $(\mathbf{s}, \mathbf{h}_1) \models \varphi$ then $(\mathbf{s}, \mathbf{h} + \mathbf{h}_1) \models \psi$



Decision Procedures

(1) Hoare Logic requires to solve **satisfiability** and **entailment** of formulae.

(2) Other crucial **robustness properties** of program analysis, like the **acyclicity** property

"Is every model satisfying φ acyclic?"

and the **garbage freedom** property

"In every model satisfying φ, are all allocated cells reachable from variables in φ?"

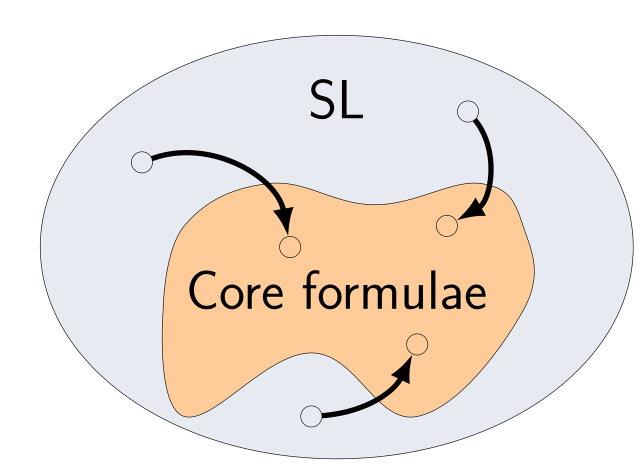
reduce to entailment as soon as we consider more powerful extensions of SL.

Therefore, it is important to:

- study the **complexity** of satisfiability and entailment, especially for SLs that can express robustness properties;
- derive calculi for these decision problems. To do this, an **internal axiomatisation** of separation logic can be helpful.

Core formulae Technique

We tame the $\ast$ and $-\!\!\ast$ operator by defining **core formulae**: a subset of SL formulae where spatial connectives appear with specific patterns.



Similarly to Gaifman's Theorem for first-order logic, we show that Boolean combinations of core formulae are as expressive as SL. We use this to derive

- **expressive power** and **complexity** of SL
- Hilbert-style **axiomatisation** with
  - axioms that eliminate $\ast$ and $-\!\!\ast$
  - axioms for the logic of core formulae.

Complexity results

We add reachability predicates to SL

$$(\mathbf{s}, \mathbf{h}) \models \mathtt{reach}^{+}(\mathtt{x}, \mathtt{y}) \text{ iff } \overset{\mathtt{x}}{\square} \to \square \to \cdots \to \overset{\mathtt{y}}{\square}$$

and only one quantified variable name ($\mathtt{u}$)

$$(\mathbf{s}, \mathbf{h}) \models \exists \mathtt{u}.\varphi \text{ iff } \exists \ell \text{ s.t. } (\mathbf{s}[\mathtt{u} \leftarrow \ell], \mathbf{h}) \models \varphi$$

With the core formulae, in [2] we show that under syntactical restrictions, this logic is in PSPACE and

- generalise all known PSPACE-complete SLs
- can encode acyclicity and garbage freedom:
  - $\varphi \models \forall \mathtt{u}\ \neg \mathtt{reach}^{+}(\mathtt{u}, \mathtt{u})$
  - $\varphi \models \forall \mathtt{u}\ ((\mathtt{u} \hookrightarrow \mathtt{u} \mathrel{-\!\!\ast} \bot) \Rightarrow \bigvee_{\mathtt{x} \in \mathsf{fv}(\varphi)} \mathtt{reach}^{+}(\mathtt{x}, \mathtt{u}) \lor \mathtt{x} = \mathtt{u})$
- weakening even slightly these restrictions leads to TOWER-hard logics.
- Full propositional SL + reachability up to paths of length 3 is already undecidable [1].

Axiomatisation

With the core formulae we define internal calculi for:

- Propositional Separation Logic [4]
- A guarded fragment of first-order SL [4]
- SL with modalities [3]

What's next?

- Use the axiomatisation to better the encoding of SL into SAT/SMT solvers.
- Improve the core formulae technique.

References

- J.C. Reynolds. Separation logic: a logic for shared mutable data structures. In LICS'02
- Peter W. O'Hearn, John C. Reynolds, Hongseok Yang. Local Reasoning about Programs that Alter Data Structures. In CSL'01

[1] S. Demri, É. Lozes, and A. Mansutti. The effects of adding reachability predicates in propositional separation logic. In FOSSACS'18

[2] A. Mansutti. Extending propositional separation logic for robustness properties. In FSTTCS'18

[3] S. Demri, R. Fervari, and A. Mansutti. Axiomatising logics with separating conjunction and modalities. In JELIA'19

[4] S. Demri, É. Lozes, and A. Mansutti. Internal calculi for Separation Logics *(submitted)*.