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 proof 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 $\mathtt{P}$*".*

The system is made of **deduction schemas**, e.g.

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

The rule ENT states 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 we would like a rule to reason locally on the memory model, e.g.

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

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

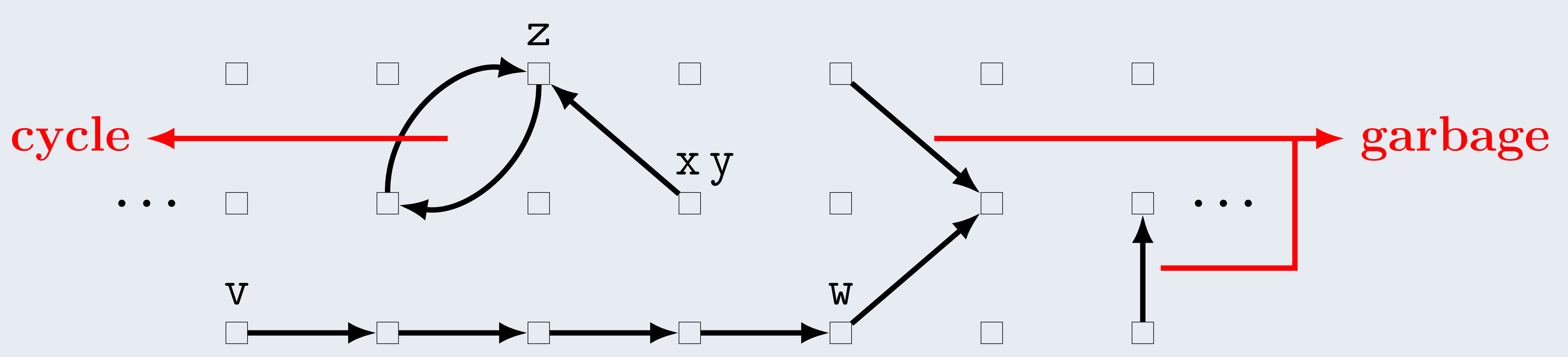
$$\frac{\{\ \mathtt{x} \hookrightarrow 1\ \}\ \ {}^{*}\mathtt{x} \leftarrow 0\ \{\ \mathtt{x} \hookrightarrow 0\ \}}{\{\ \mathtt{x} \hookrightarrow 1 \wedge \mathtt{y} \hookrightarrow 1\ \}\ \ {}^{*}\mathtt{x} \leftarrow 0\ \{\ \mathtt{x} \hookrightarrow 0 \wedge \mathtt{y} \hookrightarrow 1\ \}}$$

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, by using the separating connectives introduced in Bunched Implication Logic (P. O'Hearn, D. Pym).

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:

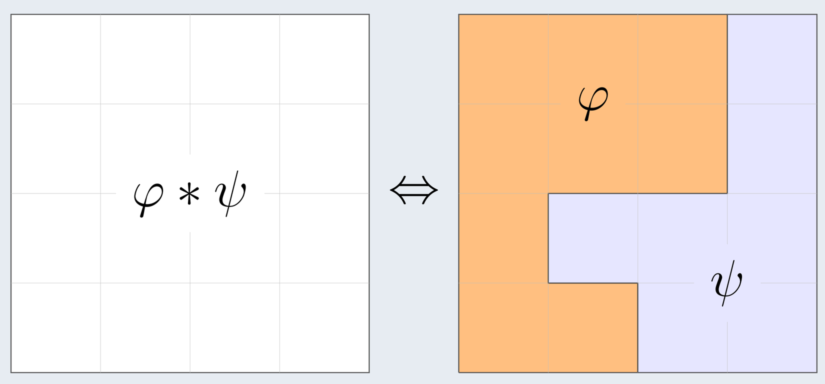


- infinite set of locations (□)
- infinite set of variables
- $\mathbf{s}$: variables $\mapsto$ locations
- $\mathbf{h}$: finite heap (□⟶□)
- a model is def. as $(\mathbf{s}, \mathbf{h})$.

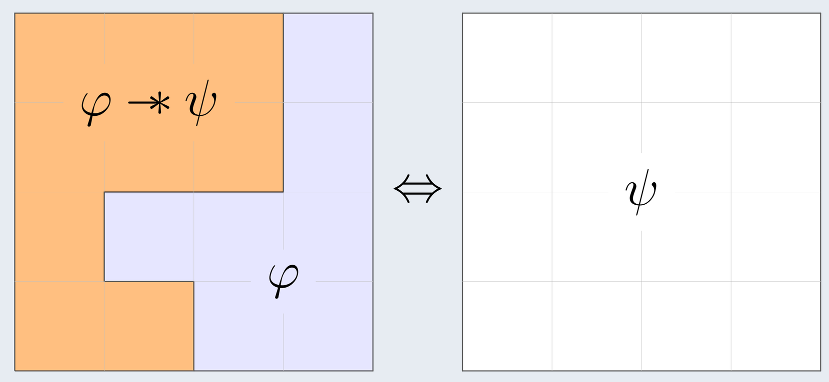
Separation Logic adds two new spatial connectives to reason **modularly** about the memory

$$\ast \qquad\qquad \mathrel{-\!\!\ast}$$

$(\mathbf{s}, \mathbf{h}) \models \varphi \ast \psi$ 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 $(\mathbf{s}, \mathbf{h} + \mathbf{h}_1) \models \psi$



We focus on...

(1) Satisfiability and **entailment** of SL and its extensions, as they are used in the Hoare system (see ENT).

(2) Deriving procedures to decide various **robustness properties**, such as 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?"

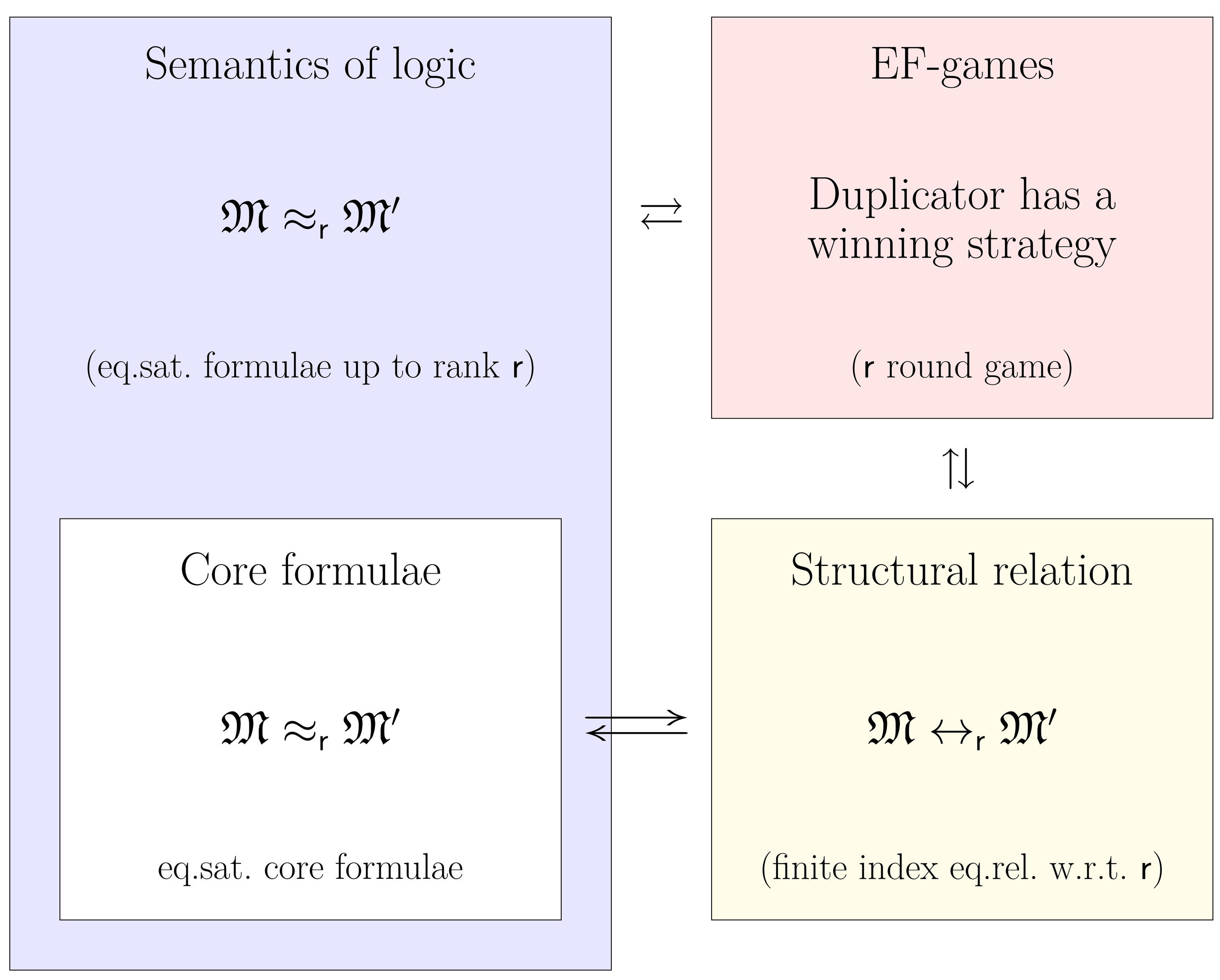
These properties are crucial in program analysis.

(3) Internal calculi for SL, as this logic is also interesting from a proof-theoretical point of view. For example, because of the $\mathrel{-\!\!\ast}$ operator, the satisfiability problem can be reduced to both validity and model checking.

General approach: Core formulae

We tame the $\ast$ and $\mathrel{-\!\!\ast}$ operators by defining **core formulae**:

- A set of formulae of SL where $\ast$ and $\mathrel{-\!\!\ast}$ appear with specific patterns.
- We show that Boolean combinations of core formulae are as expressive as SL.
- Similar to Gaifman's Theorem for FO logic.



Axiomatisation

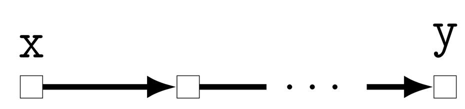
By relying on the core formulae, we define Hilbert-style axiomatisations for:

- Propositional SL and a guarded fragment of First-Order SL [4]
- SL with modalities [3]

Complexity results

We add to Propositional SL reachability predicates

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



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

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

In [2] (with the core formulae) we show that, under syntactical restrictions, this logic

- admits a PSPACE-complete satisfiability/entailment problem
- is more expressive that all known PSPACE-complete extensions of SLs
- can encode acyclicity and garbage freedom as queries of entailment:
  - $\varphi \models \forall \mathtt{u}\ \neg \mathtt{reach}^{+}(\mathtt{u}, \mathtt{u})$
  - $\varphi \models \forall \mathtt{u}\ (\mathtt{u} \hookrightarrow \mathtt{u} \mathrel{-\!\!\ast} \bot) \Rightarrow \bigvee_{\mathtt{x} \in \mathsf{fv}(\varphi)} \mathtt{reach}^{+}(\mathtt{x}, \mathtt{u}) \vee \mathtt{x} = \mathtt{u}$).
- Weakening even slightly these restrictions leads to TOWER-hard logics [5].
- Without restrictions, the satisfiability problem of this logic is undecidable [1].

What's next?

We are currently implementing our decision procedures in QBF/SMT solvers.

References

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

[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)*.

[5] A. Mansutti. An auxiliary logic on trees *(submitted)*.