Undecidability of propositional separation logic and its neighbours

James Brotherston\textsuperscript{1} and Max Kanovich\textsuperscript{2}

\textsuperscript{1}Imperial College London
\textsuperscript{2}Queen Mary University of London

LICS-25, University of Edinburgh, 12 July 2010
Separation logic (Reynolds, O’Hearn)

- Separation logic is a formalism for reasoning about memory.
Separation logic (Reynolds, O’Hearn)

- **Separation logic** is a formalism for reasoning about memory.
- **Separation models** are cancellative partial commutative monoids $\langle H, \circ, E \rangle$ ($E \subseteq H$ is a set of units).
Separation logic (Reynolds, O’Hearn)

- **Separation logic** is a formalism for reasoning about memory.
- **Separation models** are cancellative partial commutative monoids \( \langle H, \circ, E \rangle \) (\( E \subseteq H \) is a set of units).
- Propositional formulas combine standard Boolean connectives with “multiplicatives” \( * \), \( \neg * \) and \( I \).
Separation logic (Reynolds, O’Hearn)

- **Separation logic** is a formalism for reasoning about memory.
- **Separation models** are cancellative partial commutative monoids $\langle H, \circ, E \rangle$ ($E \subseteq H$ is a set of units).
- Propositional formulas combine standard Boolean connectives with “multiplicatives” $\ast$, $\ast\ast$ and $I$.
- **Separating conjunction** $F \ast G$ defined by:

  $$h \models_\rho F_1 \ast F_2 \iff h = h_1 \circ h_2 \text{ and } h_1 \models_\rho F_1 \text{ and } h_2 \models_\rho F_2$$
Separation logic (Reynolds, O’Hearn)

- **Separation logic** is a formalism for reasoning about memory.
- **Separation models** are cancellative partial commutative monoids $\langle H, \circ, E \rangle$ ($E \subseteq H$ is a set of units).
- Propositional formulas combine standard Boolean connectives with “multiplicatives” $\ast$, $\ast\ast$ and I.
- **Separating conjunction** $F \ast G$ defined by:
  \[ h \models_\rho F_1 \ast F_2 \iff h = h_1 \circ h_2 \text{ and } h_1 \models_\rho F_1 \text{ and } h_2 \models_\rho F_2 \]
- **Archetypal heap models** are $\langle H, \circ, \{e\} \rangle$, where $H = L \rightarrow_{\text{fin}} RV$ is a set of heaps, $e$ is the empty heap, and $\circ$ is (partial) union of disjoint heaps.
  *(Variations: stacks-and-heaps, heaps with permissions)*
Validity: concrete models vs. classes of models

- \( F \) is valid in \( \langle H, \circ, E \rangle \) if \( h \models \rho F \) for all \( h \in H \) and for all valuations \( \rho \) of propositional variables.
Validity: concrete models vs. classes of models

- $F$ is valid in $\langle H, \circ, E \rangle$ if $h \models_\rho F$ for all $h \in H$ and for all valuations $\rho$ of propositional variables.
- Applications of separation logic are typically based on a fixed, heap-like model.
Validity: concrete models vs. classes of models

- \( F \) is valid in \( \langle H, \circ, E \rangle \) if \( h \models_{\rho} F \) for all \( h \in H \) and for all valuations \( \rho \) of propositional variables.
- Applications of separation logic are typically based on a fixed, heap-like model.
- Validity in such a model is a subtler problem than validity in classes of models:
Validity: concrete models vs. classes of models

- $F$ is valid in $\langle H, \circ, E \rangle$ if $h \models_\rho F$ for all $h \in H$ and for all valuations $\rho$ of propositional variables.
- Applications of separation logic are typically based on a fixed, heap-like model.
- Validity in such a model is a subtler problem than validity in classes of models:
  - Normally, to show a property $Q$ given that $F$ is valid in a class of models $C$, one chooses some model $M \in C$ such that $(F \text{ valid in } M) \rightarrow Q;$
Validity: concrete models vs. classes of models

- $F$ is valid in $\langle H, \circ, E \rangle$ if $h \models \rho F$ for all $h \in H$ and for all valuations $\rho$ of propositional variables.

- Applications of separation logic are typically based on a fixed, heap-like model.

- Validity in such a model is a subtler problem than validity in classes of models:
  - Normally, to show a property $Q$ given that $F$ is valid in a class of models $\mathcal{C}$, one chooses some model $M \in \mathcal{C}$ such that $(F$ valid in $M) \rightarrow Q$;
  - but, when $M$ is given in advance, we have no such freedom!
Axiomatisations of separation logic

- BI, which is intuitionistic logic plus the MILL axioms and rules for I, * and ¬*;
Axiomatisations of separation logic

• **BI**, which is *intuitionistic* logic plus the **MILL** axioms and rules for I, * and −*;
• **BBI**, which is **BI** plus −−A ⊢ A;
Axiomatisations of separation logic

- **BI**, which is intuitionistic logic plus the MILL axioms and rules for I, ✷ and ◐;
- **BBI**, which is **BI** plus ¬¬A ⊩ A;
- **BBI + eW** where eW is I ∧ (A ✷ B) ⊩ I ∧ A, which says “you can’t split the empty heap into two non-empty heaps”;
Axiomatisations of separation logic

- **BI**, which is intuitionistic logic plus the MILL axioms and rules for $I$, $*$ and $\neg*$;
- **BBI**, which is **BI** plus $\neg\neg A \vdash A$;
- **BBI+eW** where $eW$ is $I \wedge (A * B) \vdash I \wedge A$, which says "you can’t split the empty heap into two non-empty heaps";
- **BBI+W** where $W$ is $A * B \vdash A$. This system collapses into classical logic!
Axiomatisations of separation logic

- **BI**, which is *intuitionistic* logic plus the **MILL** axioms and rules for I, *, and — *
- **BBI**, which is **BI** plus ¬¬ A ⊬ A
- **BBI+eW** where eW is I ∧ (A * B) ⊬ I ∧ A, which says "you can’t split the empty heap into two non-empty heaps"
- **BBI+W** where W is A * B ⊬ A. This system collapses into classical logic!

NB.

1. BI ⊂ BBI ⊂ BBI+eW ⊂ BBI+W, and both BI, BBI+W are decidable;
Axiomatisations of separation logic

- **BI**, which is intuitionistic logic plus the MILL axioms and rules for I, *, and ¬*;
- **BBI**, which is BI plus ¬¬A ⊬ A;
- **BBI+eW** where eW is I ∧ (A * B) ⊬ I ∧ A, which says “you can’t split the empty heap into two non-empty heaps”;
- **BBI+W** where W is A * B ⊬ A. This system collapses into classical logic!

NB.

1. BI ⊂ BBI ⊂ BBI+eW ⊂ BBI+W, and both BI, BBI+W are decidable;
2. BBI, BBI+eW are (obviously) incomplete wrt. validity in particular concrete models.
Undecidability

Machine $M$ terminates from configuration $C$.

($M$ is a non-deterministic, 2-counter Minsky machine.)
Undecidability

\[ \text{machine } M \text{ terminates from configuration } C \]

\[ \text{Thm. 3.1} \]

\[ \mathcal{F}_{M,C} \text{ provable in Minimal BBI} \]

\( M \) is a non-deterministic, 2-counter Minsky machine.
Undecidability

\( F_{M,C} \) valid in any chosen heap-like model

\[ \text{Thm. 4.2} \]

machine \( M \) terminates from configuration \( C \)

\[ \text{Thm. 3.1} \]

\( F_{M,C} \) provable in Minimal BBI

\( M \) is a non-deterministic, 2-counter Minsky machine.
Undecidability

(\(M\) is a non-deterministic, 2-counter Minsky machine.)
Undecidability

\( F_{M,C} \) valid in any chosen heap-like model

Thm. 4.2

machine \( M \) terminates from configuration \( C \)

Thm. 3.1

\( F_{M,C} \) valid in all CBI-models with indivisible units

Thm. 7.1

\( F_{M,C} \) valid in all separation models with indivisible units

Prop. 7.1

\( F_{M,C} \) provable in Minimal BBI

Prop. 7.1

\( F_{M,C} \) provable in \( BBI + \epsilon W \)

Prop. 7.1

\( F_{M,C} \) provable in CBI

(\( M \) is a non-deterministic, 2-counter Minsky machine.)
Undecidability is intimately related to infinite valuations of the propositional variables (as sets of model elements):

**Theorem**

There is a sequent $\mathcal{F}_M,\mathcal{C}$ such that, for any heap-like model $M$:

- $\mathcal{F}_M,\mathcal{C}$ is not valid in $M$, but;
- $\mathcal{F}_M,\mathcal{C}$ is valid in $M$ under every finite valuation!
Finite valuations

Undecidability is intimately related to infinite valuations of the propositional variables (as sets of model elements):

**Theorem**

There is a sequent $F_M, C$ such that, for any heap-like model $M$:

- $F_M, C$ is not valid in $M$, but;
- $F_M, C$ is valid in $M$ under every finite valuation!

So, to obtain decidable fragments of separation logic, one could:

1. give up infinite valuations (Calcagno et al., FSTTCS’01);
Undecidability is intimately related to infinite valuations of the propositional variables (as sets of model elements):

**Theorem**

There is a sequent $F_{\mathcal{M},\mathcal{C}}$ such that, for any heap-like model $M$:

- $F_{\mathcal{M},\mathcal{C}}$ is not valid in $M$, but;
- $F_{\mathcal{M},\mathcal{C}}$ is valid in $M$ under every finite valuation!

So, to obtain decidable fragments of separation logic, one could:

1. give up infinite valuations (Calcagno et al., FSTTCS’01);
2. restrict the formula language (Berdine et al., FSTTCS’04).
Summary

For the purely propositional fragment of separation logic, we have the following new results:
Summary

For the purely propositional fragment of separation logic, we have the following new results:

- validity in any given heap-like model is undecidable;
Summary

For the purely propositional fragment of separation logic, we have the following new results:

- validity in any given heap-like model is undecidable;
- validity in such a model cannot be approximated by finite valuations for propositional variables (which imposes restrictions on decidable fragments);
Summary

For the purely propositional fragment of separation logic, we have the following new results:

- validity in any given heap-like model is undecidable;
- validity in such a model cannot be approximated by finite valuations for propositional variables (which imposes restrictions on decidable fragments);
- validity in various classes of models is undecidable;
For the purely propositional fragment of separation logic, we have the following new results:

- validity in any given heap-like model is undecidable;
- validity in such a model cannot be approximated by finite valuations for propositional variables (which imposes restrictions on decidable fragments);
- validity in various classes of models is undecidable;
- and provability in various axiomatisations ($\text{BBI}$, $\text{BBI}+\text{eW}$, $\text{CBI}$, $\text{CBI}+\text{eW}$, $\ldots$) is undecidable too.
Separation logic vs. linear logic

Separation logic obeys two principles which are highly unorthodox from the perspective of linear logic:

1. The usual distributivity law

\[ A \land (B \lor C) = (A \land B) \lor (A \land C) \]

2. The exact equality

\[ \llbracket A \ast B \rrbracket = \llbracket A \rrbracket \cdot \llbracket B \rrbracket \]

(In linear logic we typically have \( \llbracket A \ast B \rrbracket \not\subseteq \llbracket A \rrbracket \cdot \llbracket B \rrbracket \).)

These two facts are entirely responsible for the undecidability of separation logic!