Undecidability of propositional separation logic and its neighbours

James Brotherston

Computer Science Seminar
Institute of Cybernetics, Tallinn University of Technology
17 Nov 2011

Outline

1. An overview of propositional separation logic

Outline

1. An overview of propositional separation logic
2. Undecidability of separation logic

Outline

1. An overview of propositional separation logic
2. Undecidability of separation logic
3. Decidable fragments: finite vs. infinite valuations

Outline

1. An overview of propositional separation logic
2. Undecidability of separation logic
3. Decidable fragments: finite vs. infinite valuations
4. Additional results

Outline

1. An overview of propositional separation logic
2. Undecidability of separation logic
3. Decidable fragments: finite vs. infinite valuations
4. Additional results

This is joint work with Prof. Max Kanovich, Queen Mary University of London. This talk is based on the paper of the same name (in Proc. LICS'10).

Part I

Propositional separation logic

Separation models

Separation logic is well established as a formalism for expressing and reasoning about properties of memory.

Separation models

Separation logic is well established as a formalism for expressing and reasoning about properties of memory.

Definition

A separation model is a cancellative partial commutative monoid $\langle H, \circ, E \rangle$.

Separation models

Separation logic is well established as a formalism for expressing and reasoning about properties of memory.

Definition

A separation model is a cancellative partial commutative monoid $\langle H, \circ, E \rangle$. We define:

$$X \cdot Y =_{\text{def}} \{x \circ y \mid x \in X, y \in Y\}$$

whence $E \subseteq H$ is a set of units such that $X \cdot E = X$.

Separation models

Separation logic is well established as a formalism for expressing and reasoning about properties of memory.

Definition

A separation model is a cancellative partial commutative monoid $\langle H, \circ, E \rangle$. We define:

$$X \cdot Y =_{\text{def}} \{x \circ y \mid x \in X, y \in Y\}$$

whence $E \subseteq H$ is a set of units such that $X \cdot E = X$.

Definition

$\langle H, \circ, E \rangle$ has indivisible units if $h_1 \circ h_2 \in E$ implies $h_1, h_2 \in E$.

(**NB.** All models of practical interest have indivisible units!)

Practical examples of separation models (I)

- Heap models $\langle H, \circ, \{e\}\rangle$, where $H = L \rightharpoonup_{\text{fin}} RV$ is the set of *heaps* (L is infinite). e is the function with empty domain, and:

$$h_1 \circ h_2 = \begin{cases} h_1 \cup h_2 & \text{if } \mathtt{dom}(h_1), \mathtt{dom}(h_2) \text{ disjoint} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Practical examples of separation models (I)

- Heap models $\langle H, \circ, \{e\}\rangle$, where $H = L \rightharpoonup_{\text{fin}} RV$ is the set of *heaps* (L is infinite). e is the function with empty domain, and:

$$h_1 \circ h_2 = \begin{cases} h_1 \cup h_2 & \text{if } \mathtt{dom}(h_1), \mathtt{dom}(h_2) \text{ disjoint} \\ \text{undefined} & \text{otherwise} \end{cases}$$

- A basic example of the above: the RAM-domain model $\langle \mathcal{D}, \circ, \{e_0\}\rangle$ where $\mathcal{D}$ is the class of finite subsets of $\mathbb{N}$, the operation $\circ$ is the union of disjoint sets, and the unit e_0 is $\emptyset$.

Practical examples of separation models (II)

- Heap-with-permissions models $\langle H, \circ, E \rangle$, where $H = L \rightharpoonup_{\text{fin}} (RV \times P)$ is a set of *heaps with permissions.* $h_1 \circ h_2$ is defined as before, except that for heaps with the same value at overlapping locations, we add the permissions.

Practical examples of separation models (II)

- Heap-with-permissions models $\langle H, \circ, E \rangle$, where $H = L \rightharpoonup_{\text{fin}} (RV \times P)$ is a set of *heaps with permissions*. $h_1 \circ h_2$ is defined as before, except that for heaps with the same value at overlapping locations, we add the permissions.

- Stack-and-heap models $\langle S \times H, \circ, E \rangle$, where H is a set of *heaps* or *heaps-with-permissions*, $S = \texttt{Var} \rightharpoonup_{\text{fin}} \texttt{Val}$ is a set of *stacks*, and $\langle s_1, h_1 \rangle \circ \langle s_2, h_2 \rangle$ is defined when $s_1 = s_2$ and $h_1 \circ h_2$ is defined (as above).

Semantics (I)

Formulas extend standard propositional connectives with the "multiplicatives" I, $*$ and $-\!*$.

Semantics (I)

Formulas extend standard propositional connectives with the "multiplicatives" I, $*$ and $-\!*$.

A valuation for a separation model $\langle H, \circ, E \rangle$ is a function ρ from propositional variables to $\mathcal{P}(H)$.

Semantics (I)

Formulas extend standard propositional connectives with the "multiplicatives" I, $*$ and $-\!*$.

A valuation for a separation model $\langle H, \circ, E \rangle$ is a function ρ from propositional variables to $\mathcal{P}(H)$.

Given $h \in H$ and formula A we define the relation $h \models_\rho A$ by induction on A:

Semantics (I)

Formulas extend standard propositional connectives with the "multiplicatives" I, $*$ and $-\!*$.

A valuation for a separation model $\langle H, \circ, E\rangle$ is a function ρ from propositional variables to $\mathcal{P}(H)$.
Given $h \in H$ and formula A we define the relation $h \models_\rho A$ by induction on A:

$$
\begin{array}{rcl}
h \models_\rho P & \Leftrightarrow & h \in \rho(P) \\
h \models_\rho F_1 \wedge F_2 & \Leftrightarrow & h \models_\rho F_1 \text{ and } r \models_\rho F_2 \\
 & \vdots & \\
h \models_\rho \mathrm{I} & \Leftrightarrow & h = e \\
h \models_\rho F_1 * F_2 & \Leftrightarrow & h = h_1 \circ h_2 \text{ and } h_1 \models_\rho F_1 \text{ and } h_2 \models_\rho F_2 \\
h \models_\rho F_1 -\!* F_2 & \Leftrightarrow & \forall h'.\ h \circ h' \text{ defined and } h' \models_\rho F_1 \text{ implies } h \circ h' \models_\rho F_2
\end{array}
$$

Semantics (I)

Formulas extend standard propositional connectives with the "multiplicatives" I, $*$ and $\mathrel{-\!\!*}$.

A valuation for a separation model $\langle H, \circ, E \rangle$ is a function ρ from propositional variables to $\mathcal{P}(H)$.

Given $h \in H$ and formula A we define the relation $h \models_\rho A$ by induction on A:

$$
\begin{array}{rcl}
h \models_\rho P & \Leftrightarrow & h \in \rho(P) \\
h \models_\rho F_1 \wedge F_2 & \Leftrightarrow & h \models_\rho F_1 \text{ and } r \models_\rho F_2 \\
& \vdots & \\
h \models_\rho \mathrm{I} & \Leftrightarrow & h = e \\
h \models_\rho F_1 * F_2 & \Leftrightarrow & h = h_1 \circ h_2 \text{ and } h_1 \models_\rho F_1 \text{ and } h_2 \models_\rho F_2 \\
h \models_\rho F_1 \mathrel{-\!\!*} F_2 & \Leftrightarrow & \forall h'.\ h \circ h' \text{ defined and } h' \models_\rho F_1 \text{ implies } h \circ h' \models_\rho F_2
\end{array}
$$

We define $[\![A]\!]_\rho =_{\text{def}} \{h \mid h \models_\rho A\}$.

Semantics (I)

Formulas extend standard propositional connectives with the "multiplicatives" I, $*$ and $-\!\!*$.

A valuation for a separation model $\langle H, \circ, E \rangle$ is a function ρ from propositional variables to $\mathcal{P}(H)$.
Given $h \in H$ and formula A we define the relation $h \models_\rho A$ by induction on A:

$$
\begin{array}{rcl}
h \models_\rho P & \Leftrightarrow & h \in \rho(P) \\
h \models_\rho F_1 \wedge F_2 & \Leftrightarrow & h \models_\rho F_1 \text{ and } r \models_\rho F_2 \\
 & \vdots & \\
h \models_\rho \mathrm{I} & \Leftrightarrow & h = e \\
h \models_\rho F_1 * F_2 & \Leftrightarrow & h = h_1 \circ h_2 \text{ and } h_1 \models_\rho F_1 \text{ and } h_2 \models_\rho F_2 \\
h \models_\rho F_1 -\!\!* F_2 & \Leftrightarrow & \forall h'.\ h \circ h' \text{ defined and } h' \models_\rho F_1 \text{ implies } h \circ h' \models_\rho F_2
\end{array}
$$

We define $[\![A]\!]_\rho =_{\mathrm{def}} \{h \mid h \models_\rho A\}$.

A "sequent" $A \vdash B$ is valid in $\langle H, \circ, E \rangle$ if $[\![A]\!]_\rho \subseteq [\![B]\!]_\rho$ for all ρ.

Semantics (II)

In any separation model $\langle H, \circ, E \rangle$ we have:

$$
\begin{array}{rcl}
[\![\mathrm{I}]\!]_\rho & = & E \\
[\![A * B]\!]_\rho & = & [\![A]\!]_\rho \cdot [\![B]\!]_\rho \\
[\![A \mathrel{-\!*} B]\!]_\rho & = & \text{largest } Z \subseteq H.\ Z \cdot [\![A]\!]_\rho \subseteq [\![B]\!]_\rho
\end{array}
$$

Semantics (II)

In any separation model $\langle H, \circ, E \rangle$ we have:

$$\begin{array}{rcl} [\![\mathrm{I}]\!]_\rho & = & E \\ [\![A * B]\!]_\rho & = & [\![A]\!]_\rho \cdot [\![B]\!]_\rho \\ [\![A \mathrel{-\!\!*} B]\!]_\rho & = & \text{largest } Z \subseteq H.\ Z \cdot [\![A]\!]_\rho \subseteq [\![B]\!]_\rho \end{array}$$

In particular this implies restricted $*$-contraction:

$$[\![I \wedge A]\!]_\rho = [\![\mathrm{I} \wedge A]\!]_\rho \cdot [\![\mathrm{I} \wedge A]\!]_\rho = [\![(\mathrm{I} \wedge A) * (\mathrm{I} \wedge A)]\!]_\rho$$

Semantics (II)

In any separation model $\langle H, \circ, E\rangle$ we have:

$$\begin{array}{rcl} [\![\mathrm{I}]\!]_\rho & = & E \\ [\![A * B]\!]_\rho & = & [\![A]\!]_\rho \cdot [\![B]\!]_\rho \\ [\![A \mathrel{-\!\!*} B]\!]_\rho & = & \text{largest } Z \subseteq H.\ Z \cdot [\![A]\!]_\rho \subseteq [\![B]\!]_\rho \end{array}$$

In particular this implies restricted $*$-contraction:

$$[\![I \wedge A]\!]_\rho = [\![\mathrm{I} \wedge A]\!]_\rho \cdot [\![\mathrm{I} \wedge A]\!]_\rho = [\![(\mathrm{I} \wedge A) * (\mathrm{I} \wedge A)]\!]_\rho$$

which doesn't hold in linear logic because, e.g.:

$$[\![A * B]\!]_\rho = \mathrm{Cl}([\![A]\!]_\rho \cdot [\![B]\!]_\rho)$$

where Cl is a *closure* operator. This is less precise, and rules out finite valuations since, e.g., $\mathrm{Cl}(\emptyset)$ is infinite.

Possible axiomatisations of separation logic

- BI, obtained by extending intuitionistic logic with the standard MILL axioms and rules for I, $*$ and $-\!\!*$;

Possible axiomatisations of separation logic

- BI, obtained by extending intuitionistic logic with the standard MILL axioms and rules for I, $*$ and $\mathrel{-\!\!*}$;
- BBI, obtained by extending classical logic with the standard MILL axioms and rules for I, $*$ and $\mathrel{-\!\!*}$;

Possible axiomatisations of separation logic

- BI, obtained by extending intuitionistic logic with the standard MILL axioms and rules for I, $*$ and $-\!*$;
- BBI, obtained by extending classical logic with the standard MILL axioms and rules for I, $*$ and $-\!*$;
- a *minimal* BBI with additives restricted to $\wedge$ and $\rightarrow$, i.e. no negation and no falsum (see next slide);

Possible axiomatisations of separation logic

- BI, obtained by extending intuitionistic logic with the standard MILL axioms and rules for I, $*$ and $-\!\!*$;
- BBI, obtained by extending classical logic with the standard MILL axioms and rules for I, $*$ and $-\!\!*$;
- a *minimal* BBI with additives restricted to $\wedge$ and $\rightarrow$, i.e. no negation and no falsum (see next slide);
- BBI+eW where eW is the restricted $*$-weakening: $\mathrm{I} \wedge (A * B) \vdash \mathrm{I} \wedge A$, which holds in all models with indivisible units. Because of restricted $*$-contraction we have $\mathrm{I} \wedge (A * B) \equiv \mathrm{I} \wedge A \wedge B$;

Possible axiomatisations of separation logic

- BI, obtained by extending intuitionistic logic with the standard MILL axioms and rules for I, $*$ and $-\!*$;
- BBI, obtained by extending classical logic with the standard MILL axioms and rules for I, $*$ and $-\!*$;
- a *minimal* BBI with additives restricted to $\wedge$ and $\rightarrow$, i.e. no negation and no falsum (see next slide);
- BBI+eW where eW is the restricted $*$-weakening: $\mathrm{I} \wedge (A * B) \vdash \mathrm{I} \wedge A$, which holds in all models with indivisible units. Because of restricted $*$-contraction we have $\mathrm{I} \wedge (A * B) \equiv \mathrm{I} \wedge A \wedge B$;
- BBI+W where W is the full $*$-weakening: $A * B \vdash A$. This system collapses into classical logic!

Minimal BBI

$$(A * B) \vdash (B * A) \qquad (A * \mathrm{I}) \vdash A$$
$$(A * (B * C)) \vdash ((A * B) * C) \qquad A \vdash (A * \mathrm{I})$$
$$(A * (A \mathrel{-\!\!*} B)) \vdash B$$

$$\frac{A \vdash B}{(A * C) \vdash (B * C)} \qquad \frac{(A * B) \vdash C}{A \vdash (B \mathrel{-\!\!*} C)}$$

(a) Axioms and rules for $*$, $\mathrel{-\!\!*}$ and I.

Minimal BBI

$$(A * B) \vdash (B * A) \qquad (A * \mathrm{I}) \vdash A$$
$$(A * (B * C)) \vdash ((A * B) * C) \qquad A \vdash (A * \mathrm{I})$$
$$(A * (A \mathrel{-\!\!*} B)) \vdash B$$

$$\frac{A \vdash B}{(A * C) \vdash (B * C)} \qquad \frac{(A * B) \vdash C}{A \vdash (B \mathrel{-\!\!*} C)}$$

(a) Axioms and rules for $*$, $\mathrel{-\!\!*}$ and I.

$$A \vdash (B \to A) \qquad A \vdash (B \to (A \wedge B))$$
$$(A \to (B \to C)) \vdash ((A \to B) \to (A \to C)) \qquad (A \wedge B) \vdash A$$
$$((A \to B) \to A) \vdash A \quad (\textit{Peirce's law}) \qquad (A \wedge B) \vdash B$$

$$\frac{A \qquad A \vdash B}{B} \qquad \frac{(A \wedge B) \vdash C}{A \vdash (B \to C)}$$

(b) Axioms and rules for $\to$ and $\wedge$.

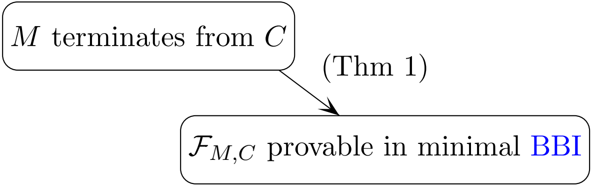
Part II

Undecidability

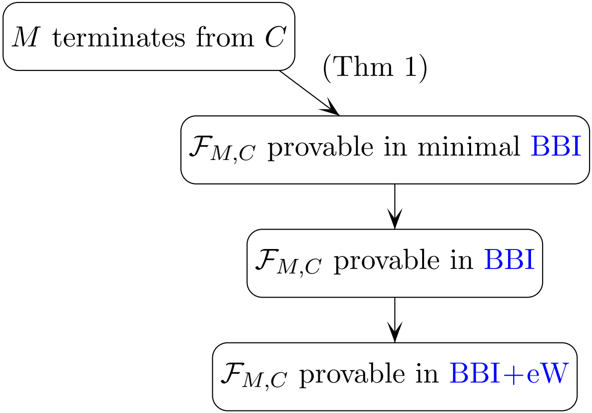
Outline proof of undecidability

M terminates from C

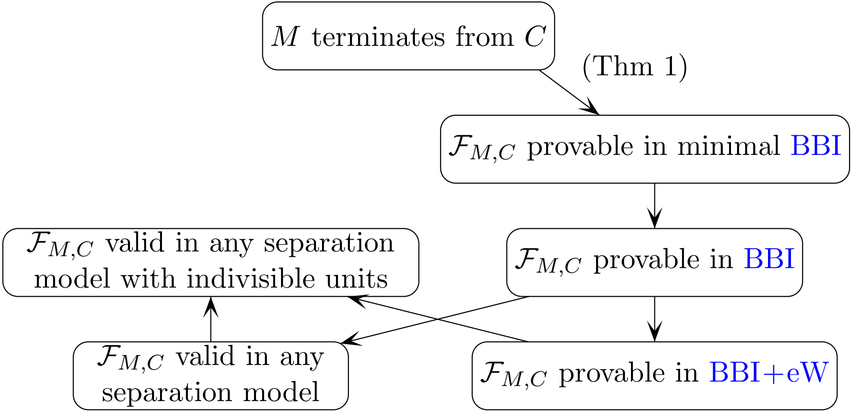
Outline proof of undecidability



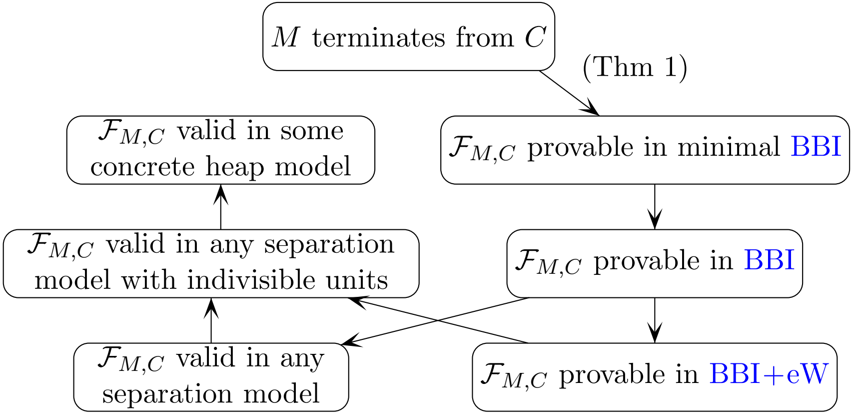
Outline proof of undecidability



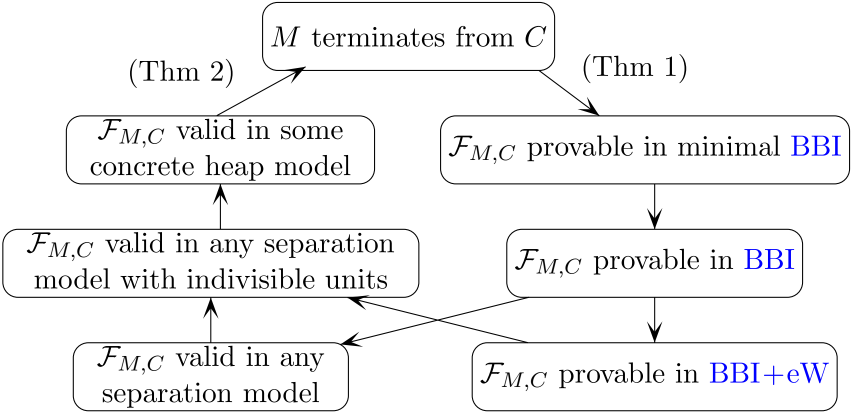
Outline proof of undecidability



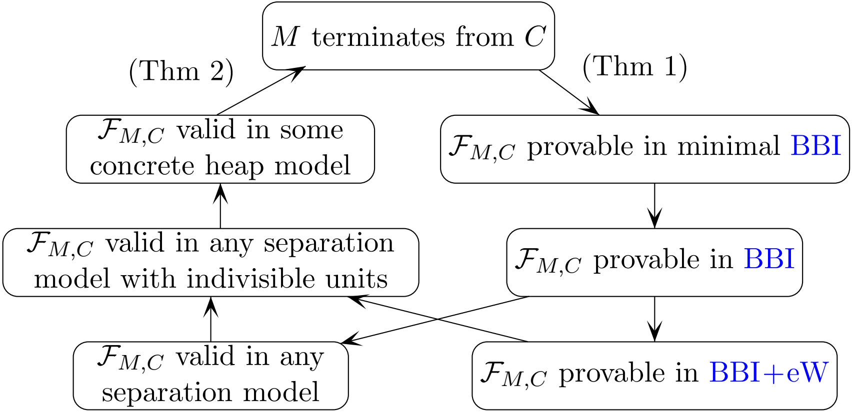
Outline proof of undecidability



Outline proof of undecidability



Outline proof of undecidability



All problems above are undecidable. Undecidability of BBI also established by Larchey-Wendling and Galmiche 2010.

Minsky machines

A Minsky machine M with counters c_1, c_2 is given by a finite set of labelled instructions of the following types, where $k \in \{1, 2\}$:

L_i: c_k++; **goto** L_j;	"increment c_k (and jump)"
L_i: c_k−−; **goto** L_j;	"decrement c_k (and jump)"
L_i: **if** $c_k = 0$ **goto** L_j;	"zero-test c_k (and jump)"
L_i: **goto** L_j;	"jump"

Minsky machines

A Minsky machine M with counters c_1, c_2 is given by a finite set of labelled instructions of the following types, where $k \in \{1, 2\}$:

L_i: c_k++; **goto** L_j;	"increment c_k (and jump)"
L_i: c_k−−; **goto** L_j;	"decrement c_k (and jump)"
L_i: **if** $c_k = 0$ **goto** L_j;	"zero-test c_k (and jump)"
L_i: **goto** L_j;	"jump"

Configurations of M have the form $\langle L_i, n_1, n_2 \rangle$. We write $\langle L_i, n_1, n_2 \rangle \Downarrow_M$ if $\langle L_i, n_1, n_2 \rangle \rightsquigarrow_M^* \langle L_0, 0, 0 \rangle$.

Minsky machines

A Minsky machine M with counters c_1, c_2 is given by a finite set of labelled instructions of the following types, where $k \in \{1, 2\}$:

L_i: c_k++; **goto** L_j;	"increment c_k (and jump)"
L_i: c_k−−; **goto** L_j;	"decrement c_k (and jump)"
L_i: **if** $c_k = 0$ **goto** L_j;	"zero-test c_k (and jump)"
L_i: **goto** L_j;	"jump"

Configurations of M have the form $\langle L_i, n_1, n_2 \rangle$. We write $\langle L_i, n_1, n_2 \rangle \Downarrow_M$ if $\langle L_i, n_1, n_2 \rangle \rightsquigarrow_M^* \langle L_0, 0, 0 \rangle$.
We introduce special labels L_{-1}, L_{-2} with instructions:

L_{-1}: c_2−−; **goto** L_{-1};	L_{-1}: **goto** L_0;
L_{-2}: c_1−−; **goto** L_{-2};	L_{-2}: **goto** L_0;

whence $\langle L_{-k}, n_1, n_2 \rangle \Downarrow_M$ iff $n_k = 0$.

Encoding configurations in minimal BBI

For each label L_i we have a propositional variable l_i. We also pick two propositional variables p_1, p_2 to represent counters c_1, c_2.

Encoding configurations in minimal BBI

For each label L_i we have a propositional variable l_i. We also pick two propositional variables p_1, p_2 to represent counters c_1, c_2. A configuration $\langle L_i, n_1, n_2 \rangle$ will be represented as:

$$l_i * p_1^{n_1} * p_2^{n_2}$$

where p_k^n denotes the formula $\underbrace{p_k * p_k * \cdots * p_k}^{n \text{ times}}$, with $p_k^0 = \mathrm{I}$.

Encoding configurations in minimal BBI

For each label L_i we have a propositional variable l_i. We also pick two propositional variables p_1, p_2 to represent counters c_1, c_2. A configuration $\langle L_i, n_1, n_2 \rangle$ will be represented as:

$$l_i * p_1^{n_1} * p_2^{n_2}$$

where p_k^n denotes the formula $\underbrace{p_k * p_k * \cdots * p_k}_{n \text{ times}}$, with $p_k^0 = \mathrm{I}$.
Also pick propositional variable b and write

$$-A =_{\mathrm{def}} A \mathrel{-\!\!*} b$$

b will be interpreted as "all terminating configurations".
$\mathrel{-\!\!*}$ corresponds to replacement of parts of configurations.

Encoding machines in minimal BBI

We code each instruction γ of a machine M as a formula $\kappa(\gamma)$ of minimal BBI:

$$\begin{array}{lcl}
L_i\colon c_k{+}{+};\ \textbf{goto}\ L_j; & \Rightarrow & (-(l_j * p_k) \mathrel{-\!\!*} -l_i) \\
L_i\colon c_k{-}{-};\ \textbf{goto}\ L_j; & \Rightarrow & (-l_j \mathrel{-\!\!*} -(l_i * p_k)) \\
L_i\colon \textbf{if}\ c_k{=}0\ \textbf{goto}\ L_j; & \Rightarrow & (-(l_j \vee l_{-k}) \mathrel{-\!\!*} -l_i) \\
L_i\colon \textbf{goto}\ L_j; & \Rightarrow & (-l_j \mathrel{-\!\!*} -l_i)
\end{array}$$

Encoding machines in minimal BBI

We code each instruction γ of a machine M as a formula $\kappa(\gamma)$ of minimal BBI:

$$\begin{array}{lcl}
L_i\colon c_k{+}{+};\,\textbf{goto}\ L_j; & \Rightarrow & (\boldsymbol{-}(l_j * p_k) \mathrel{-\!\!*} \boldsymbol{-} l_i) \\
L_i\colon c_k{-}{-};\,\textbf{goto}\ L_j; & \Rightarrow & (\boldsymbol{-} l_j \mathrel{-\!\!*} \boldsymbol{-}(l_i * p_k)) \\
L_i\colon \textbf{if}\ c_k{=}0\ \textbf{goto}\ L_j; & \Rightarrow & (\boldsymbol{-}(l_j \vee l_{-k}) \mathrel{-\!\!*} \boldsymbol{-} l_i) \\
L_i\colon \textbf{goto}\ L_j; & \Rightarrow & (\boldsymbol{-} l_j \mathrel{-\!\!*} \boldsymbol{-} l_i)
\end{array}$$

We code a whole machine $M = \{\gamma_1, \ldots, \gamma_t\}$ as:

$$\kappa(M) = \mathrm{I} \wedge \bigwedge_{i=1}^{t} \kappa(\gamma_i)$$

We'll use restricted $*$-contraction to duplicate instructions as needed!

First main theorem

Theorem

Suppose $\langle L_i, n_1, n_2 \rangle \Downarrow_M$. *Then the following sequent is derivable in minimal* BBI*:*

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge -l_0) \vdash b$$

First main theorem

Theorem

Suppose $\langle L_i, n_1, n_2 \rangle \Downarrow_M$. Then the following sequent is derivable in minimal BBI*:*

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge - l_0) \vdash b$$

Proof relies heavily on "quasi-negation" properties of $-$ (e.g. $- A \equiv - - - A$) and the restricted $*$-contraction:

$$\mathrm{I} \wedge A \vdash (\mathrm{I} \wedge A) * (\mathrm{I} \wedge A)$$

which is derivable in minimal BBI.

Second main theorem

Theorem

$\langle L_i, n_1, n_2\rangle \Downarrow_M$ *whenever the following sequent is valid in some* *concrete* *heap-like model used in practice (recall examples):*

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge -\!\!* \, l_0) \vdash b$$

Second main theorem

Theorem
$\langle L_i, n_1, n_2 \rangle \Downarrow_M$ *whenever the following sequent is valid in some concrete heap-like model used in practice (recall examples):*

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge -\!\!\!\!- l_0) \vdash b$$

Proof outline. Consider for simplicity the RAM-domain model $\langle \mathcal{D}, \circ, \{e_0\} \rangle$ based on subsets of $\mathbb{N}$. We have for any ρ:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge -\!\!\!\!- l_0)]\!]_\rho \subseteq [\![b]\!]_\rho$$

Second main theorem

Theorem
$\langle L_i, n_1, n_2 \rangle \Downarrow_M$ *whenever the following sequent is valid in some concrete heap-like model used in practice (recall examples):*

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge \neg l_0) \vdash b$$

Proof outline. Consider for simplicity the RAM-domain model $\langle \mathcal{D}, \circ, \{e_0\} \rangle$ based on subsets of $\mathbb{N}$. We have for any ρ:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (I \wedge \neg l_0)]\!]_\rho \subseteq [\![b]\!]_\rho$$

We want to pick ρ with $e_0 \in [\![\kappa(M)]\!]_\rho$ and $e_0 \in [\![I \wedge \neg l_0]\!]_\rho$ to get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \subseteq [\![b]\!]_\rho$$

and infer $\langle L_i, n_1, n_2 \rangle \Downarrow_M$.

$e_0 \in [\![\kappa(M)]\!]_\rho$: *The edge of disaster*

To check $e_0 \in [\![\kappa(M)]\!]_\rho$ we check $e_0 \in [\![\kappa(\gamma)]\!]_\rho$ for each instruction γ.

$e_0 \in [\![\kappa(M)]\!]_\rho$: *The edge of disaster*

To check $e_0 \in [\![\kappa(M)]\!]_\rho$ we check $e_0 \in [\![\kappa(\gamma)]\!]_\rho$ for each instruction γ.
Why do we encode, e.g., $L_i\colon c_k{+}{+};\,\mathbf{goto}\ L_j;$ as

$$(\neg(l_j * p_k) \mathrel{-\!\!*} \neg l_i) \;\text{ and not }\; l_i \mathrel{-\!\!*} (l_j * p_k) \;\; ?$$

$e_0 \in [\![\kappa(M)]\!]_\rho$: *The edge of disaster*

To check $e_0 \in [\![\kappa(M)]\!]_\rho$ we check $e_0 \in [\![\kappa(\gamma)]\!]_\rho$ for each instruction γ.
Why do we encode, e.g., $L_i\colon c_k{+}{+};\,\mathbf{goto}\ L_j;$ as

$$(\neg(l_j * p_k) \mathrel{-\!\!*} \neg l_i) \ \text{ and not } \ l_i \mathrel{-\!\!*} (l_j * p_k) \ \ ?$$

Let's try to check: $e_0 \in [\![l_i \mathrel{-\!\!*} (l_j * p_k)]\!]_\rho$, i.e. $[\![l_i]\!]_\rho \subseteq [\![l_j * p_k]\!]_\rho$.

$e_0 \in [\![\kappa(M)]\!]_\rho$: *The edge of disaster*

To check $e_0 \in [\![\kappa(M)]\!]_\rho$ we check $e_0 \in [\![\kappa(\gamma)]\!]_\rho$ for each instruction γ.
Why do we encode, e.g., $L_i\colon c_k{+}{+}; \textbf{goto } L_j;$ as

$$(\neg(l_j * p_k) \mathrel{-\!\!*} \neg l_i) \text{ and not } l_i \mathrel{-\!\!*} (l_j * p_k) \ ?$$

Let's try to check: $e_0 \in [\![l_i \mathrel{-\!\!*} (l_j * p_k)]\!]_\rho$, i.e. $[\![l_i]\!]_\rho \subseteq [\![l_j * p_k]\!]_\rho$.

But suppose $L_i = L_j$. In separation models this means:

$$[\![l_i]\!]_\rho \subseteq [\![l_i]\!]_\rho \cdot [\![p_k]\!]_\rho \subseteq [\![l_i]\!]_\rho \cdot [\![p_k]\!]_\rho \cdot [\![p_k]\!]_\rho \subseteq \ldots$$

i.e., any heap can be split into arbitrarily many pieces!
(Not a problem in linear logic.)

$[\![p_k^n]\!]_\rho$: *The (second) edge of disaster*

We intend that $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ should encode configuration $\langle L_i, n_1, n_2 \rangle$. Thus $[\![p_k^{n_k}]\!]_\rho$ should determine the number n_k.

$[\![p_k^n]\!]_\rho$: *The (second) edge of disaster*

We intend that $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ should encode configuration $\langle L_i, n_1, n_2 \rangle$. Thus $[\![p_k^{n_k}]\!]_\rho$ should determine the number n_k.

But composition of heaps is disjoint so that, e.g., if we take $\rho(p_k) = \{h\}$ for a nonempty heap h, then $\rho(p_k^2) = \rho(p_k * p_k)$ is empty!

$[\![p_k^n]\!]_\rho$: *The (second) edge of disaster*

We intend that $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ should encode configuration $\langle L_i, n_1, n_2 \rangle$. Thus $[\![p_k^{n_k}]\!]_\rho$ should determine the number n_k.

But composition of heaps is disjoint so that, e.g., if we take $\rho(p_k) = \{h\}$ for a nonempty heap h, then $\rho(p_k^2) = \rho(p_k * p_k)$ is empty!

In general, whenever $\rho(p_k)$ is finite we must have:

$$[\![p_k^n]\!]_\rho = [\![p_k^m]\!]_\rho$$

for sufficiently large n and m, which obstructs us in uniquely representing the number n_k by the formula p_k^n.
(We discuss decidability consequences shortly.)

Choosing a valuation

We choose a valuation ρ for $\langle \mathcal{D}, \circ, \{e_0\}\rangle$ as follows:

$$\begin{array}{rcl}\rho(p_1) & = & \{\{2^m\} \mid m \in \mathbb{N}\} \\ \rho(p_2) & = & \{\{3^m\} \mid m \in \mathbb{N}\} \\ \rho(l_i) & = & \{\{\delta_i^m\} \mid m \in \mathbb{N}\}\end{array}$$

where δ_i is a fresh prime number for each propositional variable $l_{-2}, l_{-1}, l_0, l_1, \ldots$

Choosing a valuation

We choose a valuation ρ for $\langle \mathcal{D}, \circ, \{e_0\}\rangle$ as follows:

$$\begin{array}{rcl} \rho(p_1) & = & \{\{2^m\} \mid m \in \mathbb{N}\} \\ \rho(p_2) & = & \{\{3^m\} \mid m \in \mathbb{N}\} \\ \rho(l_i) & = & \{\{\delta_i^m\} \mid m \in \mathbb{N}\} \end{array}$$

where δ_i is a fresh prime number for each propositional variable $l_{-2}, l_{-1}, l_0, l_1, \ldots$
Finally, we define:

$$\rho(b) = \bigcup_{\langle L_i, n_1, n_2\rangle \Downarrow_M} [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$$

so $\rho(b)$ is the set of interpretations of all terminating configurations.

Proof of Theorem 2

If $\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0) \vdash b$ is valid in $\langle \mathcal{D}, \circ, \{e_0\} \rangle$ then:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)]\!]_\rho \subseteq [\![b]\!]_\rho$$

Proof of Theorem 2

If $\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0) \vdash b$ is valid in $\langle \mathcal{D}, \circ, \{e_0\} \rangle$ then:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)]\!]_\rho \subseteq [\![b]\!]_\rho$$

Since $e_0 \in [\![\kappa(M)]\!]_\rho$ we get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)]\!]_\rho \subseteq [\![--l_0]\!]_\rho$$

Proof of Theorem 2

If $\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \boldsymbol{-} l_0) \vdash b$ is valid in $\langle \mathcal{D}, \circ, \{e_0\} \rangle$ then:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \boldsymbol{-} l_0)]\!]_\rho \subseteq [\![b]\!]_\rho$$

Since $e_0 \in [\![\kappa(M)]\!]_\rho$ we get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \boldsymbol{-} l_0)]\!]_\rho \subseteq [\![\boldsymbol{-}\boldsymbol{-} l_0]\!]_\rho$$

Since $e_0 \in [\![\mathrm{I} \wedge \boldsymbol{-} l_0]\!]_\rho$ (because $\langle L_0, 0, 0 \rangle \Downarrow_M$), we get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \subseteq [\![b]\!]_\rho$$

Proof of Theorem 2

If $\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \boldsymbol{-} l_0) \vdash b$ is valid in $\langle \mathcal{D}, \circ, \{e_0\} \rangle$ then:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \boldsymbol{-} l_0)]\!]_\rho \subseteq [\![b]\!]_\rho$$

Since $e_0 \in [\![\kappa(M)]\!]_\rho$ we get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \boldsymbol{-} l_0)]\!]_\rho \subseteq [\![\boldsymbol{-}\boldsymbol{-} l_0]\!]_\rho$$

Since $e_0 \in [\![\mathrm{I} \wedge \boldsymbol{-} l_0]\!]_\rho$ (because $\langle L_0, 0, 0 \rangle \Downarrow_M$), we get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \subseteq [\![b]\!]_\rho$$

Since $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ uniquely determines n_1 and n_2 we conclude $\langle L_i, n_1, n_2 \rangle \Downarrow_M$ from definition of $\rho(b)$.

Part III

Decidability: finite vs. infinite valuations

Finite valuations

The quantifier-free fragment of a certain separation theory over an infinite heap model is decidable (Calcagno et al., 2001).
WTF?

Finite valuations

The quantifier-free fragment of a certain separation theory over an infinite heap model is decidable (Calcagno et al., 2001). WTF?

There, valuations are constrained to be finite, whereas our valuation ρ is necessarily infinite.

Finite valuations

The quantifier-free fragment of a certain separation theory over an infinite heap model is decidable (Calcagno et al., 2001). WTF?

There, valuations are constrained to be finite, whereas our valuation ρ is necessarily infinite.

Theorem
There is a sequent of the form $\kappa(M) * l_i * p_1^{n_1} * (\mathrm{I} \wedge -l_0) \vdash b$ *such that, for any choice of heap-like model* $\langle H, \circ, E \rangle$*, the sequent is invalid in the model, but valid under all finite valuations* ρ*.*

Finite valuations

The quantifier-free fragment of a certain separation theory over an infinite heap model is decidable (Calcagno et al., 2001). WTF?

There, valuations are constrained to be finite, whereas our valuation ρ is necessarily infinite.

Theorem

There is a sequent of the form $\kappa(M) * l_i * p_1^{n_1} * (\mathrm{I} \wedge - l_0) \vdash b$ *such that, for any choice of heap-like model* $\langle H, \circ, E \rangle$*, the sequent is invalid in the model, but valid under all finite valuations* ρ*.*

So to obtain decidable fragments of separation logic, one should either give up infinite valuations (Calcagno et al., 2001), or restrict the formula language (Berdine et al., 2004).

Part IV

Additional results

Classical BI *(Brotherston and Calcagno, 2009)*

A CBI-model is a separation model $\langle H, \circ, E\rangle$ enriched with a total involution $\cdot^{-1}$ such that for all $h \in H.\ h \circ h^{-1} = e^{-1}$. (Cf. effect algebras in quantum mechanics.)

Classical BI *(Brotherston and Calcagno, 2009)*

A CBI-model is a separation model $\langle H, \circ, E \rangle$ enriched with a total involution $\cdot^{-1}$ such that for all $h \in H.\ h \circ h^{-1} = e^{-1}$. (Cf. effect algebras in quantum mechanics.)

E.g., can take $\langle \mathcal{D}, \circ, \{e_0\}, \cdot^{-1} \rangle$ where $\mathcal{D}$ is now the class of finite and cofinite subsets of $\mathbb{N}$, $\circ$ is union of disjoint sets, $e_0 = \emptyset$ and $\cdot^{-1}$ is set complement.

Classical BI *(Brotherston and Calcagno, 2009)*

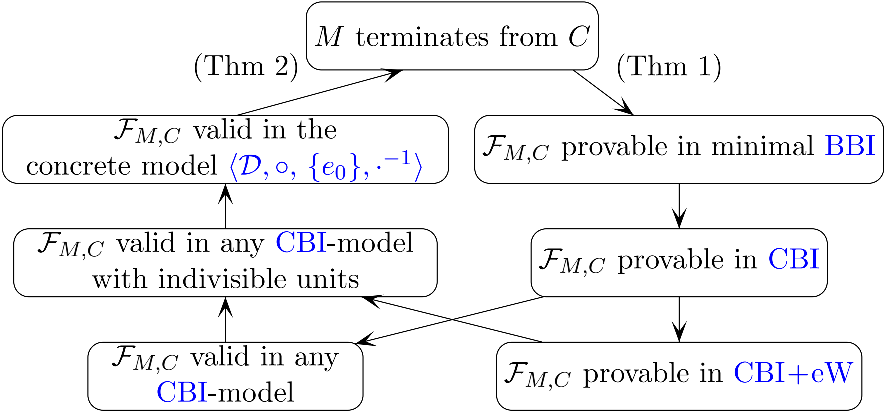
A CBI-model is a separation model $\langle H, \circ, E \rangle$ enriched with a total involution $\cdot^{-1}$ such that for all $h \in H$. $h \circ h^{-1} = e^{-1}$. (Cf. effect algebras in quantum mechanics.)

E.g., can take $\langle \mathcal{D}, \circ, \{e_0\}, \cdot^{-1} \rangle$ where $\mathcal{D}$ is now the class of finite and cofinite subsets of $\mathbb{N}$, $\circ$ is union of disjoint sets, $e_0 = \emptyset$ and $\cdot^{-1}$ is set complement.

CBI extends BBI with a multiplicative negation $\sim$ defined by:

$$h \models_\rho {\sim}A \ \Leftrightarrow\ h^{-1} \not\models_\rho A$$

Undecidability of CBI *and related problems*



Proof of Thm 2 now uses a slightly modified valuation ρ. All problems above are again undecidable.

Some references

J. Berdine, C. Calcagno and P. O'Hearn.
A decidable fragment of separation logic.
In *Proceedings of FSTTCS*, 2004.

J. Brotherston and C. Calcagno.
Classical BI (a logic for reasoning about dualising resources).
In *Proceedings of POPL*, 2009.

C. Calcagno, P. O'Hearn and H. Yang.
Computability and complexity results for a spatial assertion language for data structures.
In *Proceedings of FSTTCS*, 2001.

D. Larchey-Wendling and D. Galmiche.
Undecidability of Boolean BI through phase semantics.
In *Proceedings of LICS*, 2010.

J.C. Reynolds.
Separation logic: a logic for shared mutable data structures.
In *Proceedings of LICS*, 2002.