Undecidability of propositional separation logic and its neighbours

James Brotherston
Dept. of Computing
Imperial College London, UK

Max Kanovich
School of Electronic Engineering and Computer Science,
Queen Mary University of London, UK

***Abstract*—Separation logic has proven an effective formalism for the analysis of memory-manipulating programs.**

We show that the purely propositional fragment of separation logic is undecidable. In fact, for *any* choice of concrete heap-like model of separation logic, validity in that model remains undecidable. Besides its intrinsic technical interest, this result also provides new insights into the nature of decidable fragments of separation logic.

In addition, we show that a number of propositional systems which approximate separation logic are undecidable as well. In particular, these include both Boolean BI and Classical BI.

All of our undecidability results are obtained by means of a single direct encoding of Minsky machines.

1. Introduction, Motivations, Summary

Separation logic has become well-established in the last decade as an effective formalism for reasoning about programs that manipulate memory (in the form of heaps, stacks, etc.) [24], [14]. Automated shape analysis tools based upon separation logic are capable of verifying properties of large industrial programs [25], [5], and have been adapted to a variety of paradigms such as object-oriented programming [21], [8], and concurrent programming [9], [13].

Separation logic is usually based on a mathematical model of *heap partitioning*. In addition to the standard connectives, which are read in the usual way, an important feature of separation logic is its 'multiplicative' *separating conjunction* $*$, which generally denotes a partial operator for composing heaps whose domains are disjoint: $A_1 * A_2$ denotes the set of heaps which can be split into two disjoint heaps satisfying respectively A_1 and A_2. The separating conjunction $*$ comes along with its *unit* I, which denotes the empty heap, and its *adjoint implication* $A_1 \mathrel{-\!\!*} A_2$, denoting those heaps whose extension with any heap satisfying A_1, satisfies A_2. As a proof system, separation logic invokes a first-order extension of the propositional *bunched logic* Boolean BI [14]. Bunched logics, originating in the "logic of bunched implications" BI [20], are substructural logics that combine a standard propositional logic with a multiplicative linear logic, and admit a Kripke-style truth interpretation in which "worlds" are understood as *resources* [14], [4].

Practical applications of separation logic are based upon *concrete* heap-like models. Our main contribution in this paper is that, whichever such model of separation logic we choose, propositional validity in that model is *undecidable*.

Along the way, we also establish the undecidability of validity in various classes of separation models, and of provability in several closely related propositional systems, including both Boolean BI [14] and Classical BI [4].

Comment 1.1. Validity in a fixed heap-like model of practical interest is a much more subtle problem than validity in general classes of models.

Traditionally, to show that a formula $\mathcal{F}$ has a property Q given that $\mathcal{F}$ is valid in a *class* of models, one constructs some model in the class such that validity of $\mathcal{F}$ in this specially designed model implies Q. (In our case, Q reads that a certain computation described by $\mathcal{F}$ terminates).

Here, however, we have to deal with a specific model given *in advance*, so we have no such freedom. Instead, we have to show Q given that $\mathcal{F}$ is valid in this concrete model. The models we consider are taken from the literature on separation logic and its applications. The most common heap-like models used in practice are listed in Example 1.1.

Since existing decidable fragments of separation logic are based on concrete models, an additional advantage of our approach is that our undecidability results for these models illuminate the restrictions on these fragments.

Example 1.1. Examples of commonly-used separation models which employ a heap memory concept (cf. [6]):

(a) *Heap models* $(H, \circ, \{e\})$, where $H = L \rightharpoonup_{\text{fin}} RV$ is the set of *heaps*, i.e. finite partial functions from an infinite L to RV. The unit e is the function with empty domain, and $h_1 \circ h_2$ is the union of h_1 and h_2 when their domains are disjoint (and undefined otherwise) [1], [14], [24].

(b) *Heap-with-permission models* $(H, \circ, \{e\})$ [3] given with an underlying *permission algebra* $(P, \bullet, \mathbb{1})$, i.e. a set P equipped with a partial commutative and associative operation $\bullet$, and a distinguished element $\mathbb{1}$ such that $\mathbb{1} \bullet \pi$ is undefined for all $\pi \in P$. Then $H = L \rightharpoonup_{\text{fin}} (RV \times P)$ is the set of *heaps-with-permissions*, and $h_1 \circ h_2$ is again the union of disjoint h_1 and h_2. However, some overlap is allowed: if $h_1(\ell) = \langle v, \pi_1 \rangle$, $h_2(\ell) = \langle v, \pi_2 \rangle$ and $\pi_1 \bullet \pi_2$ is defined then h_1 and h_2 are *compatible at* ℓ. When h_1 and h_2 are compatible at all common ℓ, then $(h_1 \circ h_2)(\ell) = \langle v, \pi_1 \bullet \pi_2 \rangle$, rather than being undefined.

(c) *Stack-and-heap models* $(S \times H, \circ, E)$ [22], where H

Brotherston supported by an EPSRC Postdoctoral Fellowship.

is a set of *heaps* or *heaps-with-permissions* as above and $S = \mathtt{Var} \rightharpoonup_{\text{fin}} \mathtt{Val}$ is the set of *stacks*, partially mapping Var to Val. Here E consists of all pairs $\langle s, e\rangle$ in which e is the empty heap, and $\langle s_1, h_1\rangle \circ \langle s_2, h_2\rangle = \langle s_1, h_1 \circ h_2\rangle$ if and only if $s_1 = s_2$ and $h_1 \circ h_2$ is defined in accordance with the previous items, and is undefined otherwise.

(d) *Petri-net* (or *finite multiset*) *models*, $(H, \circ, \{e\})$, where $H = L \rightharpoonup_{\text{fin}} \mathbb{N}$ is the set of *markings*, i.e. finite partial functions from *places* L to $\mathbb{N}$, with the $\circ$ being a *total* operation of multiset union. □

In Section 2, we give the semantics of propositional separation logic and a number of propositional systems that arise naturally in developing towards an axiomatisation of separation logic.

In Section 3, we encode two-counter Minsky machines so that, whenever machine M terminates from configuration C, the corresponding sequent $\mathcal{F}_{M,C}$ is provable in a minimal version of Boolean BI, in which negation and falsum are disallowed. (This "Minimal BBI", given by Figure 1, is extremely simple but undecidable.) By soundness, the sequent $\mathcal{F}_{M,C}$ is then valid in all separation models, including any from Example 1.1.

Then, in Section 4, we show that whenever $\mathcal{F}_{M,C}$ is valid in one of the models in Example 1.1, machine M terminates from configuration C. Thus it follows that *any property between provability of $\mathcal{F}_{M,C}$ in Minimal* BBI *and validity of $\mathcal{F}_{M,C}$ in a heap-like model is undecidable* (cf. Figure 2). We state our undecidability results in Section 5.

In Section 6, we examine the limitations on decidable fragments of separation logic imposed by our undecidability results. Oddly enough, it happens that validity under all *finite valuations* for atomic propositions in a heap-like model does not imply general validity in that model.

In Section 7, we consider "dualising" separation models and related propositional systems based on Classical BI [4]. We show that our encoding of Minsky machines also yields undecidability of these systems and of validity in associated classes of models.

2. Semantics and syntax of separation logic

Here we present the propositional language of separation logic, its interpretation in separation models and a number of related bunched logic proof systems.

We abstract from the concrete separation models found in the literature by the following definition (cf. [6], [14]):

Definition 2.1. A *separation model* is a cancellative partial commutative monoid $(H, \circ, E)$. That is, $\circ$ is a partial binary operation on H which is associative and commutative, where the equality $\alpha = \beta$ means that either α and β are both undefined, or α and β are defined and equal. Cancellativity of $\circ$ means that if $z \circ x$ is defined and $z \circ x = z \circ y$ then $x = y$. The set of *units* E is a subset of H such that $E \cdot \{h\} = \{h\}$ for all $h \in H$, where we define $X \cdot Y$ as:

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

Comment 2.1. The set of units E in a separation model forms a 'unit matrix'. That is, for any $e_i, e_j \in E$ we have:

$$e_i \circ e_j = \begin{cases} e_i, & \text{if } e_i = e_j, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

In particular, if $\circ$ is total then E is forced to be a singleton $\{e\}$. We allow a set of units E rather than a single unit e in order to cover the whole spectrum of heap-like models (cf. Example 1.1(c)).

Definition 2.2. A separation model $(H, \circ, E)$ is said to have *indivisible units* if $h_1 \circ h_2 \in E$ implies $h_1 \in E$ and $h_2 \in E$ for all $h_1, h_2 \in H$. (In fact, in this case $h_1 = h_2$.)

Notice that all of the models $(H, \circ, E)$ in Example 1.1 have indivisible units – "*the empty heap cannot be split into two non-empty heaps*".

Definition 2.3. *Formulas* are built from atomic propositions p and constants $\top$, $\bot$, and I by a unary operator $\neg$ and binary connectives $\wedge$, $\vee$, $\rightarrow$, $*$, and $-\!\!*$.

For the sake of readability, we often write a formula of the form $(A \rightarrow B)$ as the 'sequent' $A \vdash B$.

Definition 2.4. A *valuation* for a separation model $(H, \circ, E)$ is a function ρ that assigns to each atomic proposition p a set $\rho(p) \subseteq H$. Given any $h \in H$ and formula A, we define the forcing relation $h \models_\rho A$ by induction on A:

$$\begin{array}{rcl}
h \models_\rho p & \Leftrightarrow & h \in \rho(p) \\
h \models_\rho \top & \Leftrightarrow & \text{always} \\
h \models_\rho \bot & \Leftrightarrow & \text{never} \\
h \models_\rho A_1 \wedge A_2 & \Leftrightarrow & h \models_\rho A_1 \text{ and } h \models_\rho A_2 \\
h \models_\rho A_1 \vee A_2 & \Leftrightarrow & h \models_\rho A_1 \text{ or } h \models_\rho A_2 \\
h \models_\rho A_1 \rightarrow A_2 & \Leftrightarrow & \text{if } h \models_\rho A_1 \text{ then } h \models_\rho A_2 \\
h \models_\rho \neg A & \Leftrightarrow & h \not\models_\rho A \\
h \models_\rho \mathrm{I} & \Leftrightarrow & h \in E \\
h \models_\rho A_1 * A_2 & \Leftrightarrow & \exists h_1, h_2.\ h = h_1 \circ h_2 \text{ and } h_1 \models_\rho A_1 \\
 & & \text{and } h_2 \models_\rho A_2 \\
h \models_\rho A_1 -\!\!* A_2 & \Leftrightarrow & \forall h'.\ \text{if } h \circ h' \text{ defined and } h' \models_\rho A_1 \\
 & & \text{then } h \circ h' \models_\rho A_2
\end{array}$$

The intended meaning of any formula A under ρ is given by $[\![A]\!]_\rho =_{\text{def}} \{h \mid h \models_\rho A\}$. In particular, we have:

$$\begin{array}{rcl}
[\![\mathrm{I}]\!]_\rho & = & E \\
[\![A \wedge B]\!]_\rho & = & [\![A]\!]_\rho \cap [\![B]\!]_\rho \\
[\![A * B]\!]_\rho & = & [\![A]\!]_\rho \cdot [\![B]\!]_\rho \\
[\![A \rightarrow B]\!]_\rho & = & \text{largest } Z \subseteq H.\ [\![A]\!]_\rho \cap Z \subseteq [\![B]\!]_\rho \\
[\![A -\!\!* B]\!]_\rho & = & \text{largest } Z \subseteq H.\ [\![A]\!]_\rho \cdot Z \subseteq [\![B]\!]_\rho
\end{array} \quad (1)$$

Definition 2.5. A formula A is *valid* in $(H, \circ, E)$ if for any valuation ρ, we have $[\![A]\!]_\rho = H$. A sequent $A \vdash B$ is *valid* in $(H, \circ, E)$, if $[\![A]\!]_\rho \subseteq [\![B]\!]_\rho$ for any valuation ρ.

Comment 2.2. Separation models with total and non-total compositions behave differently. E.g., the sequent

$$
\begin{array}{ll}
(A * B) \vdash (B * A) & (A * \mathrm{I}) \vdash A \\
(A * (B * C)) \vdash ((A * B) * C) & A \vdash (A * \mathrm{I}) \\
(A * (A \mathrel{-\!\!*} B)) \vdash B & \\
\dfrac{A \vdash B}{(A * C) \vdash (B * C)} & \dfrac{(A * B) \vdash C}{A \vdash (B \mathrel{-\!\!*} C)}
\end{array}
$$

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

$$
\begin{array}{ll}
A \vdash (B \to A) & A \vdash (B \to (A \wedge B)) \\
(A \to (B \to C)) \vdash ((A \to B) \to (A \to C)) & (A \wedge B) \vdash A \\
((A \to B) \to A) \vdash A \quad \textit{(Peirce's law)} & (A \wedge B) \vdash B \\
\dfrac{A \qquad A \vdash B}{B} & \dfrac{(A \wedge B) \vdash C}{A \vdash (B \to C)}
\end{array}
$$

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

Figure 1. Minimal Boolean BI, which employs only $\wedge$, $\to$, $*$, $-\!\!*$ and I.

$(p \wedge (p \mathrel{-\!\!*} \bot)) \vdash \bot$ is valid in any separation model in which $\circ$ is total. But, let $(H, \circ, E)$ be a typical heap-like model in which $h \circ h$ is undefined for some $h \in H$, and ρ be a valuation with $\rho(p) = \{h\}$. Then $h \models_\rho (p \wedge (p \mathrel{-\!\!*} \bot))$ while $h \not\models_\rho \bot$, so this sequent is invalid in $(H, \circ, E)$. □

Core proof systems for propositional separation logic are provided by various *bunched logics*, a class of substructural logics pioneered by O'Hearn and Pym [20].

Definition 2.6. We consider a chain of logics as follows:

- The *logic of bunched implications*, BI (cf. [20], [23], [12]) is given by **(A)** all instances of *intuitionistically* valid propositional formulas and inference rules, and **(B)** the axioms and inference rules for $*$, $-\!\!*$ and I given in section (a) of Figure 1.
- *Boolean* BI, or BBI (see [14]) is obtained from BI by expanding **(A)** above to include all instances of *classically* valid propositional formulas and inference rules.
- Since negation $\neg$ and falsum $\bot$ tend to complicate things, we introduce a positive fragment of BBI, called *Minimal* BBI, in which the formula connectives are restricted to $\wedge$, $\to$, I, $*$ and $-\!\!*$. Minimal BBI is given by Figure 1.
- We prove (Lemma 2.2) that the *restricted $*$-contraction* $(\mathrm{I} \wedge A) \vdash (A * A)$ holds in (Minimal) BBI, whereas the analogous *restricted $*$-weakening* $(\mathrm{I} \wedge (A * B)) \vdash A$ does not. Thus we introduce the system BBI+eW by enriching BBI with $(\mathrm{I} \wedge (A * B)) \vdash A$.
- Having considered restricted $*$-weakening, it is also natural to consider BBI+W, obtained by enriching BBI with the *unrestricted $*$-weakening* $(A * B) \vdash A$.

Proposition 2.1. *If A is provable in* BBI *then A is valid in all separation models. If A is provable in* BBI+eW *then A is valid in all separation models with indivisible units.*

The connection between provability in the systems above and validity in separation models is not exact. E.g., BBI is not complete even for partial commutative monoids [11].

Corollary 2.1. *Using Proposition 2.1, we have:*

$$\mathrm{BI} \subset \mathrm{BBI} \subset \mathrm{BBI{+}eW} \subset \mathrm{BBI{+}W}$$

where $\subset$ is interpreted as strict inclusion between the sets of sequents provable in each system.

Comment 2.3. Both ends of the chain of logics in Corollary 2.1 are in fact *decidable*. BI was shown decidable in [12], and BBI+W is decidable because it collapses into ordinary classical logic (see Prop. 2.3). Since the 'Boolean component' of BBI appears much simpler than the 'intuitionistic component' of BI, it was expected for a long time that BBI might be decidable as well. As for BBI+eW, it is even closer to classical logic (i.e. BBI+W), since it enjoys both $*$-contraction and $*$-weakening in a restricted form. Therefore, technically speaking, it is relatively surprising that both BBI and BBI+eW become undecidable.

Proposition 2.2. *The following forms of the deduction theorem hold for Minimal* BBI*:*

(a) *$A \wedge B \vdash C$ is provable iff $A \vdash (B \to C)$ is provable;*
(b) *$A * B \vdash C$ is provable iff $A \vdash (B \mathrel{-\!\!*} C)$ is provable;*
(c) *$B \vdash C$ is provable iff $\mathrm{I} \vdash (B \mathrel{-\!\!*} C)$ is provable.*

Lemma 2.1. *With the help of Peirce's law, we derive the following rules in Minimal* BBI*:*

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

where $A \vee B$ is an <u>*abbreviation*</u> *for $((B \to A) \to A)$.*

One of the important features of separation logic is that the *$*$-contraction*, $A \vdash (A * A)$, is not generally valid, and hence not provable in BBI by Prop. 2.1. Surprisingly, however, BBI enjoys the *restricted $*$-contraction* which holds only at the multiplicative unit I.

Lemma 2.2. *The following is provable in Minimal* BBI*:*

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

Proof. Using weakening for $\wedge$, we can easily derive $(\mathrm{I} \wedge A) * (\mathrm{I} \wedge A) \vdash (A * A)$, whence by parts (b) and (a) of Proposition 2.2 we obtain:

$$A \vdash \mathrm{I} \to ((\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A)) \tag{2}$$

Now using weakening for $\wedge$ and the axiom $(B * \mathrm{I}) \vdash B$, we can derive each of the following:

$$
\begin{array}{l}
(\mathrm{I} \wedge (A \to (A * A))) * (\mathrm{I} \wedge A) \vdash A \\
(\mathrm{I} \wedge (A \to (A * A))) * (\mathrm{I} \wedge A) \vdash A \to (A * A)
\end{array}
$$

Thus we derive $(\mathrm{I} \wedge (A \to (A * A))) * (\mathrm{I} \wedge A) \vdash (A * A)$ by modus ponens, whence by Proposition 2.2 (b) and (a) we obtain:

$$A \to (A * A) \vdash \mathrm{I} \to ((\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A)) \qquad (3)$$

By combining (2) and (3) the second derived rule of Lemma 2.1 yields $\vdash \mathrm{I} \to ((\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A))$, which is equal to $\mathrm{I} \vdash (\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A)$. From this we obtain $(\mathrm{I} \wedge A) \vdash (A * A)$ by Proposition 2.2(c). □

In the case of BBI+eW, its restricted $*$-weakening with the restricted $*$-contraction of Lemma 2.2 induces a collapse of $\wedge$ and $*$ at the level of the unit I as follows:

Corollary 2.2. *The following hold in* BBI+eW*:*

$$(\mathrm{I} \wedge (A * B)) \equiv (\mathrm{I} \wedge A \wedge B) \equiv ((\mathrm{I} \wedge A) * (\mathrm{I} \wedge B))$$

($F \equiv G$ means that both $F \vdash G$ and $G \vdash F$ are provable.)

Proposition 2.3. BBI+W *is ordinary classical logic.*

Proof. Easily, $A \equiv (A \wedge (A * \mathrm{I})) \equiv (A \wedge \mathrm{I})$ in BBI+W. Thus by Corollary 2.2, we have $(A * B) \equiv (A \wedge B)$. □

3. From computations to Minimal BBI proofs

In this section we encode terminating computations of two-counter Minsky machines in Minimal BBI.

Definition 3.1. A non-deterministic, two-counter *Minsky machine* M [19] with non-negative counters c_1, c_2 is given by a finite set of labelled *instructions* of the form:

$$\begin{array}{ll} \text{“increment } c_k \text{ by 1”} & L_i\colon c_k{+}{+};\, \mathbf{goto}\ L_j; \\ \text{“decrement } c_k \text{ by 1”} & L_i\colon c_k{-}{-};\, \mathbf{goto}\ L_j; \\ \text{“zero-test on } c_k\text{”} & L_i\colon \mathbf{if}\ c_k{=}0\ \mathbf{goto}\ L_j; \\ \text{“goto”} & L_i\colon \mathbf{goto}\ L_j; \end{array} \qquad (4)$$

where $k \in \{1, 2\}$, $i \geq 1$ and $j \geq 0$. Each L_i may label multiple instructions. The labels L_0 and L_1 are reserved for the *final* and *initial* states of M, respectively. To cope with *zero-tests*, we also add the special labels L_{-1} and L_{-2} which come equipped with the following four instructions:

$$\begin{array}{ll} L_{-1}\colon c_2{-}{-};\, \mathbf{goto}\ L_{-1}; & L_{-1}\colon \mathbf{goto}\ L_0; \\ L_{-2}\colon c_1{-}{-};\, \mathbf{goto}\ L_{-2}; & L_{-2}\colon \mathbf{goto}\ L_0; \end{array} \qquad (5)$$

A *configuration* of M is given by $\langle L, n_1, n_2 \rangle$, where the label L is the current state of M, and n_1 and n_2 are the current values of counters c_1 and c_2, resp.. We write $\rightsquigarrow_M$ for one step of M, and write $\langle L, n_1, n_2 \rangle \rightsquigarrow_M^* \langle L', n_1', n_2' \rangle$ if M can go from $\langle L, n_1, n_2 \rangle$ to $\langle L', n_1', n_2' \rangle$ in a finite number of steps. We say that M *terminates from* $\langle L, n_1, n_2 \rangle$, written $\langle L, n_1, n_2 \rangle \Downarrow_M$, if $\langle L, n_1, n_2 \rangle \rightsquigarrow_M^* \langle L_0, 0, 0 \rangle$.

The specific role of L_{-1} and L_{-2} is explained by:

Lemma 3.1. $\langle L_{-k}, n_1, n_2 \rangle \Downarrow_M$ *if and only if* $n_k = 0$.

Proof. The only instructions applicable to the configuration $\langle L_{-k}, n_1, n_2 \rangle$ are those from the group (5). □

Definition 3.2. In our encoding we use the following abbreviation. We fix an atomic proposition b, and henceforth define a “relative negation” by: $\boldsymbol{-}A =_{\text{def}} (A \mathrel{-\!\!*} b)$.

Lemma 3.2. *The following are derivable in Minimal* BBI*:*

(a) $A \vdash \boldsymbol{-}\boldsymbol{-}A$ *and* $\boldsymbol{-}\boldsymbol{-}\boldsymbol{-}A \vdash \boldsymbol{-}A$

(b) $A * (\boldsymbol{-}B \mathrel{-\!\!*} \boldsymbol{-}A) \vdash \boldsymbol{-}\boldsymbol{-}B$

(c) $\dfrac{A * B \vdash C}{A * \boldsymbol{-}\boldsymbol{-}B \vdash \boldsymbol{-}\boldsymbol{-}C}$

(d) $\dfrac{A * B \vdash D \qquad A * C \vdash D}{A * \boldsymbol{-}\boldsymbol{-}(B \vee C) \vdash \boldsymbol{-}\boldsymbol{-}D}$

Definition 3.3 (Machine encoding)**.** We encode each instruction γ from (4) by the following formula $\kappa(\gamma)$:

$$\begin{array}{rll} \kappa(L_i\colon c_k{+}{+};\, \mathbf{goto}\ L_j;) & =_{\text{def}} & (\boldsymbol{-}(l_j * p_k) \mathrel{-\!\!*} \boldsymbol{-}l_i) \\ \kappa(L_i\colon c_k{-}{-};\, \mathbf{goto}\ L_j;) & =_{\text{def}} & (\boldsymbol{-}l_j \mathrel{-\!\!*} \boldsymbol{-}(l_i * p_k)) \\ \kappa(L_i\colon \mathbf{if}\ c_k{=}0\ \mathbf{goto}\ L_j;) & =_{\text{def}} & (\boldsymbol{-}(l_j \vee l_{-k}) \mathrel{-\!\!*} \boldsymbol{-}l_i) \\ \kappa(L_i\colon \mathbf{goto}\ L_j;) & =_{\text{def}} & (\boldsymbol{-}l_j \mathrel{-\!\!*} \boldsymbol{-}l_i) \end{array}$$

where p_1, p_2, l_{-2}, l_{-1}, l_0, l_1, l_2, $\ldots$ are distinct atomic propositions (p_1 and p_2 are used to represent the counters c_1 and c_2, respectively). For a Minsky machine M given by instructions $\gamma_1, \gamma_2, .., \gamma_t$ define its encoding formula $\kappa(M)$ by:

$$\kappa(M) =_{\text{def}} (\mathrm{I} \wedge \textstyle\bigwedge_{i=1}^{t} \kappa(\gamma_i)).$$

Lemma 3.3. *For each instruction γ of a machine M, the sequent $\kappa(M) \vdash (\kappa(M) * \kappa(\gamma))$ is derivable in Minimal* BBI.

Proof. Follows from Lemma 2.2. □

Theorem 3.1. *Suppose that $\langle L_i, n_1, n_2 \rangle \Downarrow_M$ for some M. Then the following sequent is derivable in Minimal* BBI*:*

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

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

Proof. Since $A * (\mathrm{I} \wedge \boldsymbol{-}l_0) \vdash b$ is easily derivable from $A \vdash \boldsymbol{-}\boldsymbol{-}l_0$ it suffices to prove the stronger sequent:

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

We proceed by induction on the length m of the computation of $\langle L_i, n_1, n_2 \rangle \rightsquigarrow_M^* \langle L_0, 0, 0 \rangle$. In the base case $m = 0$ we have $n_1 = n_2 = 0$, and we derive by Lemma 3.2(a):

$$\kappa(M) * l_0 * \mathrm{I} * \mathrm{I} \vdash \boldsymbol{-}\boldsymbol{-}l_0$$

Next, we assume that the result holds for all computations of length $m{-}1$, and show that it holds for any computation of length m. We then proceed by case distinction on the instruction γ which yields the first step of the computation. We show the cases for an increment instruction and for a zero-test with $k{=}1$; the other cases are treated similarly.

Case $\gamma = (L_i\colon c_1{+}{+};\, \mathbf{goto}\ L_j;)$. By the case assumption we have $\langle L_i, n_1, n_2 \rangle \rightsquigarrow_M \langle L_j, n_1 + 1, n_2 \rangle$, and we are required to show that the following is derivable:

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

This derivation is produced roughly by the following chain of backward reasoning. Since $\kappa(\gamma) = (-(l_j * p_1) \mathrel{-\!\!*} -l_i)$, we can apply Lemma 3.3 to generate the obligation:

$$\kappa(M) * (-(l_j * p_1) \mathrel{-\!\!*} -l_i) * l_i * p_1^{n_1} * p_2^{n_2} \vdash --l_0$$

We can use part (b) of Lemma 3.2 to reduce this to:

$$\kappa(M) * (--(l_j * p_1)) * p_1^{n_1} * p_2^{n_2} \vdash --l_0$$

Using part (a) of Lemma 3.2 this reduces again to:

$$\kappa(M) * (--(l_j * p_1)) * p_1^{n_1} * p_2^{n_2} \vdash ----l_0$$

which further reduces by part (c) of Lemma 3.2 to:

$$\kappa(M) * l_j * p_1^{n_1+1} * p_2^{n_2} \vdash --l_0$$

which is provable by the induction hypothesis.

Case $\gamma = (L_i\colon \textbf{if } c_1 = 0 \textbf{ goto } L_j;)$. By the case assumption we have $\langle L_i, 0, n_2\rangle \rightsquigarrow_M \langle L_j, 0, n_2\rangle$, and must derive:

$$\kappa(M) * l_i * \mathrm{I} * p_2^{n_2} \vdash --l_0$$

By the equivalence $A * \mathrm{I} \equiv A$ and using Lemma 3.3 to duplicate $\kappa(\gamma)$ as in the previous case, it suffices to show:

$$\kappa(M) * (-(l_j \vee l_{-1}) \mathrel{-\!\!*} -l_i) * l_i * p_2^{n_2} \vdash --l_0$$

By employing part (b) of Lemma 3.2 we can reduce this to:

$$\kappa(M) * --(l_j \vee l_{-1}) * p_2^{n_2} \vdash --l_0$$

which further reduces by parts (a) and (d) of the same lemma to the pair of proof obligations:

$$\kappa(M) * l_j * p_2^{n_2} \vdash --l_0 \quad \text{and} \quad \kappa(M) * l_{-1} * p_2^{n_2} \vdash --l_0$$

The first of these is immediate by induction hypothesis and $A * \mathrm{I} \equiv A$. For the second, note that by Lemma 3.1 we have $\langle L_{-1}, 0, n_2\rangle {\Downarrow_M}$. Taking into account that L_{-1} labels only decrement and goto instructions by definition, the sequent required is derived by induction on n_2, applying an argument similar to the corresponding cases in the present theorem. This completes the case, and the proof. □

4. From validity to terminating computations

In this section, our goal is to show that for each of the concrete models in Example 1.1, we have $\langle L_i, n_1, n_2\rangle {\Downarrow_M}$ whenever the following sequent is valid:

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

For the sake of perspicuity we establish this property first for the *RAM-domain model*, which can be seen as the simplest heap model from Example 1.1(a), obtained by taking $L = \mathbb{N}$ and RV a singleton set. Then, we extend our approach to the most general stack-and-heap models from Example 1.1(c), of which all the other models can be seen as special instances, obtained by choosing appropriate L, RV, set of stacks S and permission algebra P.

Definition 4.1. The *RAM-domain model* is $(\mathcal{D}, \circ, \{e_0\})$ where $\mathcal{D}$ is the class of finite subsets of $\mathbb{N}$, and $d_1 \circ d_2$ is the union of the disjoint sets d_1 and d_2 (with $d_1 \circ d_2$ undefined if d_1 and d_2 are not disjoint). The unit e_0 is $\emptyset$.

Definition 4.2. We introduce the following valuation ρ_0 for the RAM-domain model $(\mathcal{D}, \circ, \{e_0\})$:

$$\begin{aligned} \rho_0(p_1) &= \{\ \{2\}, \{4\}, \{8\}, \ldots, \{2^m\}, \ldots\ \} \\ \rho_0(p_2) &= \{\ \{3\}, \{9\}, \{27\}, \ldots, \{3^m\}, \ldots\ \} \\ \rho_0(l_i) &= \{\ \{\delta_i\}, \{\delta_i^2,\} \{\delta_i^3\}, \ldots, \{\delta_i^m\}, \ldots\ \} \end{aligned}$$

where each δ_i is taken as a fresh prime number for each of the atomic propositions l_{-2}, l_{-1}, l_0, l_1, l_2, …, and:

$$\rho_0(b) = \bigcup_{\langle L_i, n_1, n_2\rangle \Downarrow_M} [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$$

Lemma 4.1. *Definition 4.2 guarantees that for any n, a finite set d belongs to $[\![p_k^n]\!]_{\rho_0}$ if and only if d consists of exactly n distinct powers of the corresponding prime. Thus any element of $[\![p_k^n]\!]_{\rho_0}$ uniquely determines the number n.*

Proof. By induction on n. E.g., by definition, $[\![p_1 * p_1]\!]_{\rho_0}$ consists of two-element sets of the form $\{2^{m_1}, 2^{m_2}\}$. □

Comment 4.1. Our choice of $\rho_0(p_1)$ and $\rho_0(p_2)$ to have *infinitely many disjoint elements* is dictated by peculiarities of composition $\circ$ in the heap model.
Moreover, for any *finite* choice of $\rho_0(p_k)$, we must have

$$[\![p_k^n]\!]_{\rho_0} = [\![p_k^m]\!]_{\rho_0}$$

for *all* sufficiently large n and m, which obstructs us in uniquely representing the contents n of counter c_k by the formula p_k^n.
(We discuss decidability consequences in Section 6.)

Lemma 4.2. *$e_0 \models_{\rho_0} \kappa(M)$ for any machine M.*

Proof. Writing $M = \{\gamma_1, \ldots, \gamma_t\}$, we have by Defn. 2.4:

$$\begin{aligned} e_0 \models_{\rho_0} \kappa(M) \ &\Leftrightarrow\ e_0 \models_{\rho_0} \mathrm{I} \wedge \textstyle\bigwedge_{i=1}^{t} \kappa(\gamma_i) \\ &\Leftrightarrow\ e_0 \in \{e_0\} \text{ and } \forall i_{(1 \le i \le t)}.\ e_0 \models_{\rho_0} \kappa(\gamma_i) \end{aligned}$$

Thus it suffices to show that $e_0 \models_{\rho_0} \kappa(\gamma)$ for any instruction γ. We present the cases for an increment instruction and for a zero-test instruction with $k = 1$; the other cases are similar.

Case $\gamma = (L_i\colon c_1{+}{+}; \textbf{goto } L_j;)$.
We have $\kappa(\gamma) = (-(l_j * p_1) \mathrel{-\!\!*} -l_i)$. To show $e_0 \models_{\rho_0} \kappa(\gamma)$, we must show that $x \models_{\rho_0} -(l_j * p_1)$ implies $x \models_{\rho_0} -l_i$ for any $x \in \mathcal{D}$. First note that we have for all $x \in \mathcal{D}$:

$x \models_{\rho_0} -(l_j * p_1)$
$\Leftrightarrow \forall x'.\ x \circ x'$ defined and $x' \models_{\rho_0} l_j * p_1$ yields $x \circ x' \models_{\rho_0} b$
$\Leftrightarrow \forall x'.\ x \circ x'$ defined and ($x' = y \circ z$ and $y \models_{\rho_0} l_j$ and $z \models_{\rho_0} p_1$) implies $x \circ x' \models_{\rho_0} b$
$\Leftrightarrow \forall y, z.\ x \circ y \circ z$ defined and $y \in \rho_0(l_j)$ and $z \in \rho_0(p_1)$ implies $x \circ y \circ z \models_{\rho_0} b$
$\Leftrightarrow \exists n_1, n_2.\ x \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ and $\langle L_j, n_1 + 1, n_2\rangle {\Downarrow_M}$

The last equivalence follows because, since the elements of $\mathcal{D}$ are finite sets whereas $\rho_0(l_j)$ and $\rho_0(p_1)$ contain *infinitely many disjoint sets*, $x \circ y \circ z$ must be defined for some $y \in \rho_0(l_j)$, $z \in \rho_0(p_1)$, in which case $x \circ y \circ z \models_{\rho_0} b$ must hold. By the same token, we also have for all $x \in \mathcal{D}$:

$$x \models_{\rho_0} -l_i \Leftrightarrow x \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \text{ and } \langle L_i, n_1, n_2\rangle \Downarrow_M$$

Since $\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1+1, n_2\rangle$ by applying the increment instruction γ, we have that $\langle L_j, n_1+1, n_2\rangle \Downarrow_M$ implies $\langle L_i, n_1, n_2\rangle \Downarrow_M$, so that $x \models_{\rho_0} -(l_j * p_1)$ implies $x \models_{\rho_0} -l_i$ as required.

Case $\gamma = (L_i\text{: } \mathbf{if}\ c_1 = 0\ \mathbf{goto}\ L_j;)$. In this case, we have $\kappa(\gamma) = (-(l_j \vee l_{-1}) \mathrel{-\!\!*} -l_i)$. To show $e_0 \models_{\rho_0} \kappa(\gamma)$, we must show that $x \models_{\rho_0} -(l_j \vee l_{-1})$ implies $x \models_{\rho_0} -l_i$ for any $x \in \mathcal{D}$. As in the previous case, we have for all $x \in \mathcal{D}$:

$$x \models_{\rho_0} -l_i \Leftrightarrow x \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \text{ and } \langle L_i, n_1, n_2\rangle \Downarrow_M$$

We also have, for all $x \in \mathcal{D}$:

$$\begin{aligned} & x \models_{\rho_0} -(l_j \vee l_{-1}) \\ \Leftrightarrow\ & \forall x'.\ x \circ x' \text{ defined and } x' \in \rho_0(l_j) \cup \rho_0(l_{-1}) \text{ implies} \\ & x \circ x' \models_{\rho_0} b \\ \Leftrightarrow\ & x \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0},\ \langle L_j, n_1, n_2\rangle \Downarrow_M,\ \langle L_{-1}, n_1, n_2\rangle \Downarrow_M \\ \Leftrightarrow\ & x \in [\![p_2^{n_2}]\!]_{\rho_0} \text{ and } \langle L_j, 0, n_2\rangle \Downarrow_M \ \text{ (by Lemma 3.1)} \end{aligned}$$

The penultimate equivalence above requires reasoning similar to that employed in the previous case: $x \circ x'$ must be defined for some $x' \in \rho_0(l_j)$ and for some $x' \in \rho_0(l_{-1})$.

Since $\langle L_i, 0, n_2\rangle \rightsquigarrow_M \langle L_j, 0, n_2\rangle$ by applying the zero-test γ, we have: $\langle L_j, 0, n_2\rangle \Downarrow_M$ implies $\langle L_i, 0, n_2\rangle \Downarrow_M$, i.e. $x \models_{\rho_0} -(l_j \vee l_{-1})$ implies $x \models_{\rho_0} -l_i$ as required. □

Lemma 4.3. $e_0 \models_{\rho_0} (\mathrm{I} \wedge -l_0)$.

Proof. We trivially have $e_0 \models_{\rho_0} \mathrm{I}$. Since $\langle L_0, 0, 0\rangle \Downarrow_M$, we have $\rho_0(l_0) \subseteq \rho_0(b)$ by construction of ρ_0, which straightforwardly entails $e_0 \models_{\rho_0} -l_0$. □

Theorem 4.1. *If* $(\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)) \vdash b$ *is valid in* $(\mathcal{D}, \circ, \{e_0\})$ *then* $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

Proof. By the definition of validity and using the equations (1) we have:

$$\begin{aligned} & [\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)]\!]_{\rho_0} \subseteq \rho_0(b) \\ \text{i.e. } & [\![\kappa(M)]\!]_{\rho_0} \cdot [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \cdot [\![\mathrm{I} \wedge -l_0]\!]_{\rho_0} \subseteq \rho_0(b) \end{aligned}$$

Since $e_0 \in [\![\kappa(M)]\!]_{\rho_0}$ by Lemma 4.2 and $e_0 \in [\![\mathrm{I} \wedge -l_0]\!]_{\rho_0}$ by Lemma 4.3 we have in particular:

$$[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \subseteq \rho_0(b)$$

By Lemma 4.1 and (1), the set $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ uniquely determines the numbers n_1 and n_2, whence our construction of $\rho_0(b)$ yields $\langle L_i, n_1, n_2\rangle \Downarrow_M$. □

Having established our Theorem 4.1 for the basic RAM-domain model, we now extend it to the most sophisticated *stack-and-heap* models from Example 1.1(c), in which the heaps are heaps-with-permissions. All the models from Example 1.1 can be seen as special instances of such models by taking appropriate L, RV, S and P.

Definition 4.3. Let $(S \times H, \circ, E)$ be a stack-and-heap model from Example 1.1(c), where S is a set of stacks, $H = \mathbb{N} \rightharpoonup_{\text{fin}} (RV \times P)$ is a set of heaps-with-permissions and $(P, \bullet, \mathbb{1})$ is a permission algebra. (Recall that $\mathbb{1} \bullet \pi$ is undefined for all $\pi \in P$.)

Based on our valuation ρ_0 for the RAM-domain model in Definition 4.2, we introduce a valuation ρ_1 for $(S \times H, \circ, E)$ as follows. First, we fix an arbitrary stack $s_0 \in S$, and for each finite set $d \subseteq \mathbb{N}$ we define the set $[d] \subseteq S \times H$ by:

$$[d] = \{\langle s_0, h\rangle \mid \mathtt{domain}(h) = d \text{ and } \forall \ell \in d.\ h(\ell) = \langle \_, \mathbb{1}\rangle\}.$$

Then for any atomic p we define its valuation by:

$$\rho_1(p) = \bigcup_{d \in \rho_0(p)} [d].$$

Lemma 4.4. *For any atomic propositions* p *and* q*:*

$$[\![p * q]\!]_{\rho_1} = [\![p]\!]_{\rho_1} \cdot [\![q]\!]_{\rho_1} = \bigcup_{d \in [\![p*q]\!]_{\rho_0}} [d].$$

Proof. It suffices to show that $[d_1 \circ d_2] = [d_1] \cdot [d_2]$.

For disjoint d_1 and d_2 this is given by construction.

For overlapping d_1 and d_2, assume that $\ell \in d_1 \cap d_2$, and $\langle s_0, h_1\rangle \circ \langle s_0, h_2\rangle$ is defined for some $\langle s_0, h_1\rangle \in [d_1]$ and $\langle s_0, h_2\rangle \in [d_2]$. By construction of $[d_1]$ and $[d_2]$, this implies $h_1(\ell) = h_2(\ell) = \langle \_, \mathbb{1}\rangle$. But then since $\langle s_0, h_1\rangle \circ \langle s_0, h_2\rangle$ is defined, we must have $\mathbb{1} \bullet \mathbb{1}$ defined, which is a contradiction. Thus $[d_1] \cdot [d_2]$ is empty when $d_1 \circ d_2$ is undefined.

Lemma 4.5. *For any formula* A *of the form* l_i, $(l_i * p_k)$, *or* $(l_i \vee l_j)$, *we have that* $[\![A]\!]_{\rho_1} \cdot \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1}$ *holds if and only if* $\langle s_0, h\rangle \in [d]$ *with* $[\![A]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$ *where* $d = \mathtt{domain}(h)$.

Proof. Follows from Lemma 4.4. □

Lemma 4.6. $\langle s_0, e_0\rangle \models_{\rho_1} \kappa(M)$ *for any machine* M.

Proof. As in Lemma 4.2, we show that $\langle s_0, e_0\rangle \models_{\rho_1} \kappa(\gamma)$ for any γ in the group (4). Recalling that $-A = (A \mathrel{-\!\!*} b)$, each $\kappa(\gamma)$ is of the form $((A \mathrel{-\!\!*} b) \mathrel{-\!\!*} (B \mathrel{-\!\!*} b))$, so it suffices to prove $[\![A \mathrel{-\!\!*} b]\!]_{\rho_1} \subseteq [\![B \mathrel{-\!\!*} b]\!]_{\rho_1}$. Using the equations (1), this amounts to showing, for any $\langle s, h\rangle$:

$$[\![A]\!]_{\rho_1} \cdot \{\langle s, h\rangle\} \subseteq [\![b]\!]_{\rho_1} \ \Rightarrow\ [\![B]\!]_{\rho_1} \cdot \{\langle s, h\rangle\} \subseteq [\![b]\!]_{\rho_1} \quad (6)$$

If $s \neq s_0$ then $[\![A]\!]_{\rho_1} \cdot \{\langle s, h\rangle\} = [\![B]\!]_{\rho_1} \cdot \{\langle s, h\rangle\} = \emptyset$, in which case (6) holds trivially. For the case $s = s_0$, assume $[\![A]\!]_{\rho_1} \cdot \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1}$, and let $d = \mathtt{domain}(h)$. By Lemma 4.5 we have $\langle s_0, h\rangle \in [d]$, and $[\![A]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$. Lemma 4.2 shows that $e_0 \models_{\rho_0} \kappa(\gamma)$ and thus we have $[\![A \mathrel{-\!\!*} b]\!]_{\rho_0} \subseteq [\![B \mathrel{-\!\!*} b]\!]_{\rho_0}$, so that $[\![B]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$, whence Lemma 4.5 yields $[\![B]\!]_{\rho_1} \cdot \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1}$. □

Lemma 4.7. $\langle s_0, e_0\rangle \models_{\rho_1} (\mathrm{I} \wedge -l_0)$.

Proof. Similar to Lemma 4.3. □

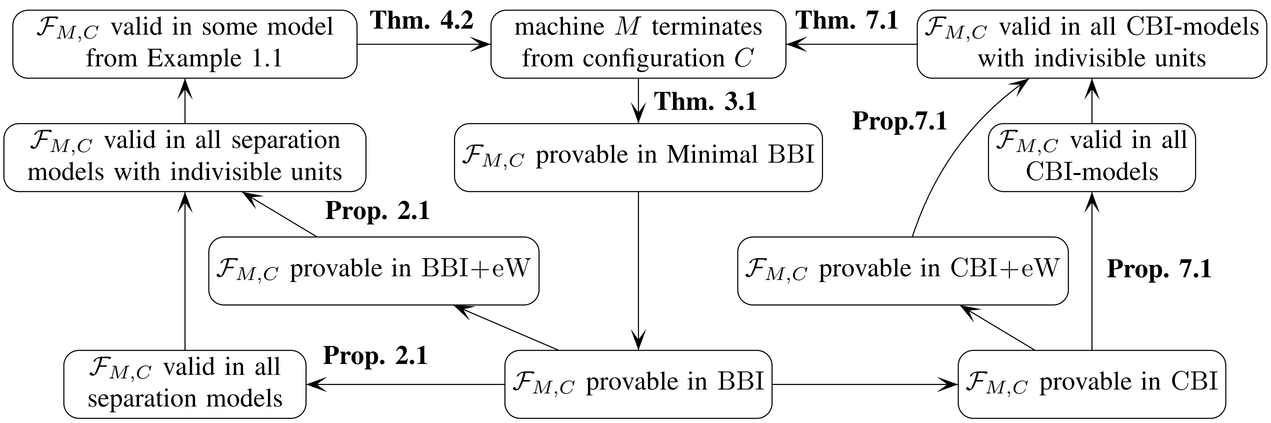

Figure 2. Diagrammatic proof of undecidability. The arrows are implications, and $\mathcal{F}_{M,C}$ is a sequent built from machine M and configuration C: For C of the form $\langle L_1, n_1, n_2\rangle$, by $\mathcal{F}_{M,C}$ we abbreviate the sequent $(\kappa(M) * l_1 * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)) \vdash b$. The problems at each node are all undecidable.

Theorem 4.2. *If a sequent $\mathcal{F}_{M,l_i,n_1,n_2}$ of the form*

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

is valid in some concrete model listed in Example 1.1 then $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

Proof. The case of Petri nets with their *total* $\circ$ can be covered by the original valuation ρ_0 from Definition 4.2.

For other models, by taking appropriate L, RV, set of stacks S and permission algebra P, we may assume that $\mathcal{F}_{M,l_i,n_1,n_2}$ is valid in a stack-and-heap model given in Definition 4.3. Thus we have by definition of validity:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -l_0)]\!]_{\rho_1} \subseteq [\![b]\!]_{\rho_1}$$

Taking into account Lemmas 4.6 and 4.7, we get:

$$[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_1} \subseteq \rho_1(b).$$

According to Lemmas 4.4 and 4.1, the set $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_1}$ uniquely determines the numbers n_1 and n_2, so that our construction of $\rho_1(b)$ yields $\langle L_i, n_1, n_2\rangle \Downarrow_M$. □

5. Undecidability of separation logic

Now, based upon Figure 2, we may state the following:

Corollary 5.1. *The following problems are undecidable:*[1]

(a) *provability in Minimal* BBI*;*
(b) *provability in* BBI*;*
(c) *provability in* BBI+eW*;*
(d) *validity in the class of all separation models;*
(e) *validity in the class of all separation models with indivisible units;*
(f) *validity in the class of all total separation models;*
(g) *validity in the class of all total separation models with indivisible units;*
(h) *validity in any of the concrete models in Example 1.1, for arbitrarily chosen locations L, values RV, stacks S, and permission algebra P (note L must be infinite).*[2]

Proof. The termination problem for Minsky machines, which is undecidable [19], reduces to each of the problems above by the diagram in Figure 2. □

Corollary 5.2. *Neither Minimal* BBI *nor* BBI *nor* BBI+eW *has the finite model property.*

Proof. A recursive enumeration of proofs and finite counter-models for any of the logics above would yield a decision procedure for provability, which is impossible. □

6. Finite approximations in infinite models

Our undecidability results for propositional separation logic seem to be at odds with the decidability of the quantifier-free fragment of a certain separation theory over an infinite heap model, due to Calcagno et al. [7]. The crucial difference is that their decidability result is restricted to *finite* valuations ρ such that $\rho(p)$ is finite for every atomic proposition p. Namely, in [7] each p represents one *cell*, i.e. a heap whose domain is a singleton. Their decidability result is highly non-trivial because their language contains $-\!\!*$ and the underlying separation model employs a non-total $\circ$, so that, e.g., whenever $[\![A]\!]_\rho$ is *finite*, $[\![A -\!\!* B]\!]_\rho$ becomes *infinite*. Below we investigate this phenomenon.

Theorem 6.1. *Let $(H, \circ, \{e_0\})$ be a heap model (cf. Example 1.1(a)). Then there is an algorithm that, for any finite valuation ρ, and any sequent $\mathcal{F}_{M,l_i,n_1,n_2}$ of the form:*

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

decides whether this sequent is valid under the valuation ρ.

Proof. In principle, this can be deduced from [7]. Our direct construction in Lemma 6.1 illustrates subtleties of the problem, caused by non-totality of the composition $\circ$.

[1] In fact, we prove undecidability for a family of formulas $\mathcal{F}_{M,C}$, which have a very simple structure of restricted depth.

[2] In the case of total Petri-net models even sufficiently large *finite* L provide undecidability.

Lemma 6.1. *There is an algorithm that, for any finite valuation ρ, decides whether $e_0 \models_\rho \kappa(M)$ holds or not.*

Proof. As in Lemmas 4.2 and 4.6 (cf. (6)), for any $\kappa(\gamma)$ of the form $((A \mathrel{-\!\!*} b) \mathrel{-\!\!*} (B \mathrel{-\!\!*} b))$, we have to check whether $e_0 \models_\rho \kappa(\gamma)$, which means checking whether the sentence:

$$\forall z((\llbracket A \rrbracket_\rho \cdot \{z\} \subseteq \llbracket b \rrbracket_\rho) \;\Rightarrow\; (\llbracket B \rrbracket_\rho \cdot \{z\} \subseteq \llbracket b \rrbracket_\rho))$$

is true or not (here $\llbracket A \rrbracket_\rho$, $\llbracket B \rrbracket_\rho$ and $\llbracket b \rrbracket_\rho$ are finite).
We consider two cases depending on the domain of z:

- Take all z with $\llbracket A \rrbracket_\rho \cdot \{z\} \neq \emptyset$. The list of all these z such that, in addition, $\llbracket A \rrbracket_\rho \cdot \{z\} \subseteq \llbracket b \rrbracket_\rho$ must be *finite* (recall our $\circ$ is a kind of disjoint union). Then it remains to check for each of these z whether $\llbracket B \rrbracket_\rho \cdot \{z\} \subseteq \llbracket b \rrbracket_\rho$ holds or not.
- If $\llbracket A \rrbracket_\rho \cdot \{z\} = \emptyset$, then trivially $\llbracket A \rrbracket_\rho \cdot \{z\} \subseteq \llbracket b \rrbracket_\rho$, so we have to check whether the following sentence is true or not:

$$\forall z((\llbracket A \rrbracket_\rho \cdot \{z\} = \emptyset) \;\Rightarrow\; (\llbracket B \rrbracket_\rho \cdot \{z\} \subseteq \llbracket b \rrbracket_\rho)) \qquad (7)$$

Let $\llbracket A \rrbracket_\rho = \{f_1, f_2, .., f_m\}$ and $\alpha_i = \mathtt{domain}(f_i)$ for all i, and $\llbracket B \rrbracket_\rho = \{g_1, .., g_t\}$ and $\beta_j = \mathtt{domain}(g_j)$ for all j.
For each choice of ℓ_1, ℓ_2,.., ℓ_m from α_1, α_2,.., α_m, respectively, we write $d_{\ell_1,\ell_2,..,\ell_m}$ for the set $\{\ell_1, \ell_2, .., \ell_m\}$.
In the following we rely upon the fact that, although $\llbracket A \rrbracket_\rho \cdot \{z\} = \emptyset$ for *infinitely many* z, the domain of each of these z must be a superset of some $d_{\ell_1,..,\ell_m}$.
Case 1. Assume $\beta_j \cap d_{\ell_1,..,\ell_m} \neq \emptyset$ for all β_j and $d_{\ell_1,..,\ell_m}$. Since, for each z in question, $\mathtt{domain}(z)$ is a superset of some $d_{\ell_1,..,\ell_m}$, we have $\llbracket B \rrbracket_\rho \cdot \{z\} = \emptyset$. So, in this case, (7) is true.
Case 2. Assume $\beta_j \cap d_{\ell_1,..,\ell_m} = \emptyset$ for some β_j and $d_{\ell_1,..,\ell_m}$. Let $\widetilde{n}$ be an element of L such that $\widetilde{n}$ does not occur in β_j, $d_{\ell_1,..,\ell_m}$, and $\llbracket b \rrbracket_\rho$, and let $\widetilde{z}$ be a heap such that $\mathtt{domain}(\widetilde{z}) = \{\widetilde{n}\} \cup d_{\ell_1,..,\ell_m}$. Then $g_j \circ \widetilde{z}$ is defined but $g_j \circ \widetilde{z} \notin \llbracket b \rrbracket_\rho$, and, in this case, (7) is false. □

We complete the proof for Theorem 6.1 as follows. Given an $\mathcal{F}_{M,l_i,n_1,n_2}$ we first use Lemma 6.1 to compute $\llbracket \kappa(M) \rrbracket_\rho$ and $\llbracket \mathrm{I} \wedge - l_0 \rrbracket_\rho$ (both are subsets of $\{e_0\}$). If either of these sets is empty then trivially $\mathcal{F}_{M,l_i,n_1,n_2}$ is valid under the ρ. Otherwise, $\llbracket \kappa(M) \rrbracket_\rho = \llbracket \mathrm{I} \wedge - l_0 \rrbracket_\rho = \llbracket \mathrm{I} \rrbracket_\rho$, and it only remains to check if the *finite* set $\llbracket l_i * p_1^{n_1} * p_2^{n_2} \rrbracket_\rho$ is a subset of the *finite* set $\llbracket b \rrbracket_\rho$, which is straightforward. □

Corollary 6.1. *There is a sequent $\mathcal{F}_{M,l_1,n_0,0}$ of the form*

$$(\kappa(M) * l_1 * p_1^{n_0} * (\mathrm{I} \wedge - l_0)) \vdash b$$

such that, for each separation model from Example 1.1, $\mathcal{F}_{M,l_1,n_0,0}$ is not valid in this model (and hence the corresponding computation does not terminate), but $\mathcal{F}_{M,l_1,n_0,0}$ is valid in this model under all finite valuations ρ.

Proof. Take M such that $K_M = \{n \mid \langle L_1, n, 0\rangle \Downarrow_M\}$ is undecidable. Let W_M be the set of all n such that $\mathcal{F}_{M,l_1,n,0}$ is not valid in some model from Example 1.1 under some finite valuation ρ. By Theorem 6.1, W_M is recursively enumerable. According to Theorem 3.1 and Proposition 2.1, K_M and W_M are disjoint. Moreover, since K_M is recursively enumerable but undecidable, W_M is not the whole complement of K_M. Therefore, we can find a number n_0 such that $n_0 \notin K_M \cup W_M$. Since $n_0 \notin K_M$, Theorem 4.2 implies that $\mathcal{F}_{M,l_1,n_0,0}$ is not valid in any model from Example 1.1. However, $n_0 \notin W_M$ implies that, in every such model, $\mathcal{F}_{M,l_1,n_0,0}$ is valid under all finite valuations ρ. □

7. Extension to Classical BI

In this section, we extend our undecidability results to the class of "dualising separation models", whose proof-theoretical basis is given by Classical BI, or CBI [4].

Definition 7.1. A CBI-*model* is given by $(H, \circ, e, \cdot^{-1})$, where $\langle H, \circ, \{e\}\rangle$ is a separation model (with a single unit e) and $\cdot^{-1} : H \to H$ satisfies $h \circ h^{-1} = e \circ e^{-1} = e^{-1}$ for all $h \in H$. (As a corollary, $(h^{-1})^{-1} = h$.)

The CBI-models we consider here form a subclass of the more general *relational* CBI-models given in [4].

Example 7.1. Examples of CBI-models (cf. [4]):

(a) $([0, 1], \circ, 0, \cdot^{-1})$, where $x_1 \circ x_2$ is $x_1 + x_2$ but undefined when $x_1 + x_2 > 1$. The inverse x^{-1} is $1 - x$.

(b) $(\Sigma, \circ, \varepsilon, \overline{\cdot})$ where Σ is any class of *languages* containing the empty language and closed under union $\cup$ and complement $\overline{\cdot}$. Here $d_1 \circ d_2$ is the union of disjoint languages d_1 and d_2 (in the overlapping case, $d_1 \circ d_2$ is undefined). E.g., Σ may be the class of regular languages, or the class of finite and co-finite sets.

(c) *Effect algebras* [10], which arise in the foundations of quantum mechanics, can be considered as CBI-models with indivisible units.

(d) *Permission algebras* $(P, \bullet, \mathbb{1})$ [3] enriched with a 'formal unit' e and 'formal equalities' $e \bullet h = h \bullet e = e$ can be shown to be exactly non-degenerate CBI-models with indivisible units.

Definition 7.2. Following Definition 2.6, we introduce a second chain of logics as follows:

- CBI [4] is obtained from BBI by extending its language with a constant $\widetilde{\mathrm{I}}$, and adding the axiom $\sim\sim A \vdash A$, where $\sim A$ is an abbreviation for $(A \mathrel{-\!\!*} \widetilde{\mathrm{I}})$.
- CBI+eW is obtained by extending CBI with the restricted $*$-weakening $(\mathrm{I} \wedge (A * B)) \vdash A$;
- CBI+W is obtained by extending CBI with the unrestricted $*$-weakening $(A * B) \vdash A$.

Validity of CBI-formulas with respect to CBI-models $(H, \circ, e, \cdot^{-1})$ is defined as in Definition 2.5 once we extend the forcing relation $h \models_\rho A$ given in Definition 2.4 with the clause: $h \models_\rho \widetilde{\mathrm{I}} \Leftrightarrow h \neq e^{-1}$.

Proposition 7.1. *If A is provable in* CBI *then A is valid in all* CBI-*models, and if A is provable in* CBI+eW *then A is valid in all* CBI-*models with indivisible units.*

Corollary 7.1. *Using Proposition 7.1 we have:*

$$\mathrm{BBI} \subset \mathrm{CBI} \subset \mathrm{CBI{+}eW} \subset \mathrm{CBI{+}W}$$

where the inclusions hold even for the original language without $\widetilde{\mathrm{I}}$ *(the inclusion* BBI ⊂ CBI *was established in* [4]*).*

Proposition 7.2. CBI+W *collapses into classical logic.*

Proof. As in Proposition 2.3, $A * B \equiv A \wedge B$ and $\mathrm{I} \equiv \top$. Then $\widetilde{\mathrm{I}} \equiv {\sim}\mathrm{I} \equiv {\sim}\top \equiv \bot$, which forces ${\sim}A \equiv \neg A$. □

Since Minimal BBI-provability implies CBI-provability, to establish undecidability for CBI it suffices (see Figure 2) to prove the analogue of Theorem 4.1 for a CBI-model.

Definition 7.3. We introduce the model $(\mathcal{D}^+, \circ, e_0, \cdot^{-1})$, where $\mathcal{D}^+$ is the class of finite and co-finite subsets of $\mathbb{N}$, $\circ$ is disjoint union, the unit e_0 is $\emptyset$ and $\cdot^{-1}$ is set complement.

Definition 7.4. By extending the valuation ρ_0 in Definition 4.2, we define a valuation ρ_C for $(\mathcal{D}^+, \circ, e_0, \cdot^{-1})$ as follows: ρ_C coincides with ρ_0 on all atomic propositions except b, and $\rho_C(b) = \rho_0(b) \cup \{d \in \mathcal{D}^+ \mid d \text{ is cofinite}\}$.

Lemma 7.1. $e_0 \models_{\rho_C} \kappa(M)$ *for any machine* M.

Proof. As in Lemma 4.2, we must show $e_0 \models_{\rho_C} \kappa(\gamma)$ for any instruction γ. Here we only examine the case of an increment instruction $\gamma = (L_i\colon c_k{+}{+};\, \mathbf{goto}\ L_j;)$ for $k{=}1$. As in the corresponding case of Lemma 4.2, we must show that $[\![-\!\!\!*\, (l_j * p_1)]\!]_{\rho_C} \subseteq [\![-\!\!\!*\, l_i]\!]_{\rho_C}$. Assuming that $x \models_{\rho_C} -\!\!\!*\, (l_j * p_1)$, we have:

$\forall y, z.\ ((x \circ y \circ z$ defined and $y \in \rho_C(l_j)$ and $z \in \rho_C(p_1))$ implies $x \circ y \circ z \in \rho_C(b))$

We need to show $x \models_{\rho_C} -\!\!\!*\, l_i$, for which we have two cases. First, in the case that x is *finite* then the reasoning from the analogous case of Lemma 4.2 applies since ρ_C coincides with ρ_0 on all variables except b. That is, for some $y \in \rho_C(l_j)$ and $z \in \rho_C(p_1)$, $x \circ y \circ z$ is defined and thus $x \circ y \circ z \in \rho_C(b)$, whence $x \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_C}$ and $\langle L_j, n_1 + 1, n_2\rangle \Downarrow_M$. By applying the instruction γ, we have $\langle L_i, n_1, n_2\rangle \Downarrow_M$, which implies that $x \circ x' \in \rho_C(b)$ whenever $x \circ x'$ is defined and $x' \in \rho_C(l_i)$, i.e. $x \models_{\rho_C} -\!\!\!*\, l_i$ as required. In the case that x is *cofinite* then, by the same token, $x \circ x'$ is either undefined or cofinite for any $x' \in \rho_C(l_i)$, in which case $x \models_{\rho_C} -\!\!\!*\, l_i$ as required because $\rho_C(b)$ contains *all* cofinite sets. □

Lemma 7.2. *Similar to Lemma 4.3,* $e_0 \models_{\rho_C} (\mathrm{I} \wedge -\!\!\!*\, l_0)$.

Theorem 7.1. *If* $\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -\!\!\!*\, l_0) \vdash b$ *is valid in the model* $(\mathcal{D}^+, \circ, e_0, \cdot^{-1})$, *then* $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

Proof. By the definition of validity we have:

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

By Lemmas 7.1 and 7.2 we have in particular:

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

Since ρ_C coincides with ρ_0 on all atomic propositions except b, the set $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_C}$ is finite, and uniquely determines n_1 and n_2 by Lemma 4.1 and equations (1). Our construction of $\rho_C(b)$ then yields $\langle L_i, n_1, n_2\rangle \Downarrow_M$. □

Again, based on Figure 2, we can assert the following:

Corollary 7.2. *The following problems are undecidable:*
(a) *provability in* CBI*;*
(b) *provability in* CBI+eW*;*
(c) *validity in the class of all* CBI*-models;*
(d) *validity in the class of all* CBI*-models with indivisible units;*
(e) *validity in the concrete* CBI*-model* $(\mathcal{D}^+, \circ, e_0, \cdot^{-1})$.

Proof. Similar to Corollary 5.1. □

8. Concluding remarks

Our main contribution is that separation logic is undecidable even at the *purely propositional* level. Specifically, we have established undecidability of validity in any particular heap-like model drawn from the literature on separation logic and its applications. Corollary 5.1(h) provides an infinite number of concrete undecidable models of practical and theoretical importance. That is, however we choose L, RV, S and P in Example 1.1 (with L infinite and RV, S, P possibly degenerate), we *always* get an undecidable model.

We also have established the undecidability of validity in various classes of separation models, and of provability in BBI and CBI and their siblings. In fact, to obtain a new exhibit for the 'Undecidability Zoo', we need add only *classical* conjunction and implication to the multiplicatives, without invoking negation and falsum (see Figure 1).

One of the above problems, namely the decidability of BBI, was widely considered to be open for quite a long time. Undecidability of BBI is a corollary of our main results on separation logic. However, immediately prior to publication of this paper, we discovered that undecidability of BBI can in fact be deduced from the undecidability of equational theories over certain algebras stated in [17]. An alternative proof of BBI undecidability has also been given independently in [11]. However, our paper overlaps with [17] and [11] *only* with respect to undecidability of BBI (which is obtained by very different techniques in all cases). In this paper, we give a direct proof not only of undecidability of BBI, but also of Minimal BBI, BBI+eW, CBI, CBI+eW, and of validity in various classes of separation models, as well as validity in any *particular* heap-like model of practical interest. There is no ad hoc connection between the latter problems and BBI.

Our undecidability results also shed new light on the correlations between separation logic and linear logic.

From the point of view of logical principles, there are clear differences between separation logic and linear logic. E.g., distributivity of additive conjunction over disjunction:

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$

holds even in BI but fails in linear logic. More specific to Boolean BI, the restricted $*$-contraction:

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

holds even in Minimal BBI as shown by our Lemma 2.2, but this too fails in linear logic. Finally, while adding the unrestricted $*$-weakening $(A * B) \vdash A$ to linear logic gives us *affine logic*, adding it to BBI forces a collapse into classical logic (Proposition 2.3).

From a semantic perspective, the precise expression of properties of memory in separation logic is based on the fact that we have:

$$[\![A * B]\!]_\rho = [\![A]\!]_\rho \cdot [\![B]\!]_\rho$$

i.e. the interpretation of $A * B$ is *exactly* the product of the interpretations of A and B. (This fact is also of crucial importance to its undecidability.) Linear logic interpretations deal only with sets that are *closed* with respect to a certain closure operator Cl, which, in particular, violates the above exact equality. Indeed, the same is true of BI interpretations [20]. Not only is this less precise, it admits no possibility of finite valuations in these logics since, e.g., in linear logic even $Cl(\emptyset)$ is always infinite.

From a technical point of view, we kill *all* our 'undecidable birds' with one stone – a direct encoding of Minsky machines, which single encoding suffices to cover all cases of interest (see Figure 2). A direct adaptation of the encoding of Minsky machines developed for full linear logic in [15] does not work properly for separation logic due to the differences between the two mentioned above, so we have had to develop a new encoding. Roughly speaking, in the linear logic encoding, each step in the derivation corresponds to a single forward step in the computation. In contrast, in our encoding, each step in the derivation corresponds to a 'backward move' from a class of terminating computations to a class of shorter terminating computations. An additional twist is that we require a much more complicated interpretation in heap-like models because of the way *partial* composition is defined in these models.

Finally, our undecidability results for *concrete* heap-like models give new insights into the nature of decidable fragments of separation logic such as those given in [2], [7], as well as imposing boundaries on decidability. E.g., we can deduce that to obtain decidability in a heap-like model, one should either give up infinite valuations (as in [7]) or restrict the formula language (as in [2]).

Acknowledgments. We are greatly indebted to Peter O'Hearn and Hongseok Yang for their fruitful discussions and suggestions. Thanks to Sam Staton for his observation on effect algebras [10], and to Tadeusz Litak who has attracted our attention to [17]. We owe special thanks to the referees for their scrupulous and insightful comments, which have allowed us to improve the paper.

References

[1] A. Ahmed, L. Jia, and D. Walker. Reasoning about hierarchical storage. In *Proc. LICS-18*, pp.33-44, 2003.

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

[3] R. Bornat, C. Calcagno, P. O'Hearn, and M. Parkinson. Permission accounting in separation logic. In *Proc. POPL-32*, pp.59-70, 2005.

[4] J. Brotherston and C. Calcagno. Classical BI (A logic for reasoning about dualising resource). In *Proc. POPL-36*, pp.328-339, 2009.

[5] C. Calcagno, D. Distefano, P. O'Hearn, and H. Yang. Compositional shape analysis by means of bi-abduction. In *Proc. POPL-36*, pp.289-300, 2009.

[6] C. Calcagno, P. O'Hearn, and H. Yang. Local action and abstract separation logic. In *Proc. LICS-22*, pp.366-378, 2007.

[7] C. Calcagno, H. Yang, and P. O'Hearn. Computability and complexity results for a spatial assertion language for data structures. In *Proc. FSTTCS*, pp.108-119, 2001.

[8] D. Distefano and M. Parkinson. jStar: Towards practical verification for Java. In *Proc. OOPSLA*, pp.213-226, 2008.

[9] M. Dodds, X. Feng, M. Parkinson, and V. Vafeiadis. Deny-guarantee reasoning. In *Proc. ESOP*, pp.363-377, 2009.

[10] D. Foulis and M. Bennett. Effect algebras and unsharp quantum logics. *Foundations of Physics*, 24:1331-1352, 1994.

[11] D. Galmiche and D. Larchey-Wendling. Undecidability of Boolean BI through phase semantics. LICS-2010, to appear.

[12] D. Galmiche, D. Mery, and D. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, 15:1033-1088, 2005.

[13] A. Gotsman, B. Cook, M. Parkinson, and V. Vafeiadis. Proving that non-blocking algorithms don't block. In *Proc. POPL-36*, pp.16-28, 2009.

[14] S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *Proc. POPL-28*, 2001.

[15] M. Kanovich. The direct simulation of Minsky machines in linear logic. In *Advances in Linear Logic*, pp.123-145, 1995.

[16] M. Kanovich. Petri nets, Horn programs, linear logic, and vector games, *Annals of Pure and Applied Logic*, 75:107-135, 1995.

[17] A. Kurucz, I. Nemeti, I. Sain and A. Simon. Decidable and undecidable modal logics with a binary modality, J. Logic, Language and Information, 4 (1995), 191-206.

[18] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56(1-3):239-311, 1992.

[19] M. Minsky. *Computation: finite and infinite machines*, 1967.

[20] P. O'Hearn and D. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215-244, June 1999.

[21] M. Parkinson and G. Bierman. Separation logic, abstraction and inheritance. In *Proc. POPL-35*, 2008.

[22] M. Parkinson, R. Bornat, and C. Calcagno. Variables as resource in Hoare logics. In *Proc. LICS*, pp.137-146, 2006.

[23] D. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*. Applied Logic Series, Kluwer, 2002.

[24] J. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proc. 17th LICS*, 2002.

[25] H. Yang, O. Lee, J. Berdine, C. Calcagno, B. Cook, D. Distefano, and P. O'Hearn. Scalable shape analysis for systems code. In *Proc. CAV*, 2008.