A

Undecidability of propositional separation logic and its neighbours

JAMES BROTHERSTON, University College London
and MAX KANOVICH, Queen Mary, University of London

In this paper, we investigate the logical structure of memory models of theoretical and practical interest. Separation logic provides us with an effective language for reasoning about such memory models. Our main result is that for *any* concrete choice of heap-like memory model, validity in that model is *undecidable* even for purely propositional formulas in this language.

The main novelty of our approach to the problem is that we focus on validity in specific, concrete memory models, as opposed to validity in general classes of models.

Besides its intrinsic technical interest, this result also provides new insights into the nature of their decidable fragments. In particular, we show that in order to obtain such decidable fragments, either the formula language must be severely restricted or the valuations of propositional variables must be constrained.

In addition, we show that a number of propositional systems that approximate separation logic are undecidable as well. In particular, this resolves the open problems of decidability for Boolean BI and Classical BI.

Moreover, we provide one of the simplest undecidable propositional systems currently known in the literature, called "Minimal Boolean BI", by combining the purely positive implication-conjunction fragment of Boolean logic with the laws of multiplicative $*$-conjunction, its unit and its adjoint implication, originally provided by intuitionistic multiplicative linear logic. Each of these two components is individually decidable: the implication-conjunction fragment of Boolean logic is co-NP-complete, and intuitionistic multiplicative linear logic is NP-complete.

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

Categories and Subject Descriptors: F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*Logics of programs*; *Assertions*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Model theory*; *Proof theory*

General Terms: Theory, Verification

Additional Key Words and Phrases: separation logic, undecidability, memory models, bunched logic

ACM Reference Format:
Brotherston, J., and Kanovich, M. 2012. Undecidability of propositional separation logic and its neighbours. J. ACM V, N, Article A (January YYYY), 41 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

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. [Reynolds 2002; Ishtiaq and O'Hearn 2001; Calcagno et al. 2007]. Automated shape analysis tools based upon separation logic are capable of verifying properties of large industrial programs [Yang et al. 2008; Calcagno et al. 2011], and have been adapted to a variety of paradigms such as object-oriented programming [Parkinson and Bier-

Brotherston and Kanovich are supported by EPSRC.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

man 2008; Distefano and Parkinson 2008; Gardner et al. 2012] and concurrent programming [Dodds et al. 2009; Gotsman et al. 2009].

Separation logic is usually based on a mathematical model of *heap partitioning*. In addition to the standard 'additive' Boolean connectives, which are read in the usual way, separation logic features certain 'multiplicative' connectives which are interpreted as operations in this model. The most important of these is the so-called *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. Depending on the application, first-order quantifiers and inductively defined predicates may also be included in the logic.

Our main contribution in this paper is that, separation logic is *undecidable even at the purely propositional level and irrespective of the choice of underlying memory model*.

The novelty of our approach to the problem is that we focus on the logic for *concrete* memory models of practical interest, and prove that, *whichever concrete heap-like memory model we choose, the validity of propositional separation logic formulas in that model is undecidable*. The models we consider - which are the most common memory models used in practice - are listed in Section 2 below.

As an immediate corollary of this strict result, we get undecidability of validity in various classes of such models, and undecidability of provability in several closely related propositional systems, including Boolean BI [Ishtiaq and O'Hearn 2001] and Classical BI [Brotherston and Calcagno 2009].

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

As usual, to demonstrate that a validity problem in a classes of models is undecidable, we aim to exhibit a reduction from the halting problem for a suitable class of machines. In our case, these are non-deterministic, 2-counter Minsky machines. Thus, given a formal system, we can construct a provable formula $\mathcal{F}_{M,C}$ in its language which is intended to represent the termination of the Minsky machine M from the configuration C. By soundness of the system, provability of $\mathcal{F}_{M,C}$ implies its validity in some class of models.

In order to prove the faithfulness of our encoding, we have to show the converse direction that if $\mathcal{F}_{M,C}$ is valid in this class of models then M terminates from C. Now, traditionally, to show that a formula $\mathcal{F}$ has the desired property Q given that $\mathcal{F}$ is valid in a *class* of models, one constructs *some* "canonical" model $\mathcal{M}$ in this class such that validity of $\mathcal{F}$ in this specially designed model $\mathcal{M}$ implies Q.

However, when we consider the problem of validity in a concrete model $\mathcal{M}_0$ *specified in advance*, we have no such freedom. Instead, we have to show Q given only that $\mathcal{F}$ is valid in this fixed concrete model $\mathcal{M}_0$. The difficulty of this task is that there is no *ad hoc* connection between the given model $\mathcal{M}_0$ (of practical interest) and the artificially designed "canonical" model $\mathcal{M}$ one would normally seek to create. □

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. In particular, it transpires that undecidability is closely connected to the trade-off between finite and infinite valuations of the atomic propositions (cf. Section 10).

Seen as a proof system, separation logic invokes a first-order extension of the propositional *bunched logic* Boolean BI [Ishtiaq and O'Hearn 2001]. Bunched logics, originating in the "logic of bunched implications" BI [O'Hearn and Pym 1999], can be conceived of as substructural logics that combine a standard propositional logic with various "multiplicative" connectives originally provided by linear logic (cf. [Pym 2002; Brotherston 2012]). Their

practical significance for computer science stems from their Kripke-style truth interpretation in which the "worlds" of the Kripke models are understood as *resources* [Ishtiaq and O'Hearn 2001; Pym et al. 2004; Brotherston and Calcagno 2010].

In proving our main undecidability results on separation logic, we also establish the undecidability of provability in several propositional systems based on bunched logic. In particular, these include the bunched logics Boolean BI [Ishtiaq and O'Hearn 2001] and Classical BI [Brotherston and Calcagno 2010]. The decidability of Boolean BI was widely considered an open problem for quite a long time (see, e.g., [Galmiche and Larchey-Wendling 2006]).

The remainder of this paper is structured as follows.

First, in Section 2, we list a series of common memory models of theoretical and/or practical interest in various subfields of computer science and mathematics. Then, in Section 3, we give the formula language and semantics of propositional separation logic, which is suitable for expressing properties of these memory models, and, in Section 4, we present a number of propositional logical systems that arise naturally in developing towards an axiomatisation of separation logic.

An overview of our undecidability results and the proof methodology used to establish them is then given in Section 5. The proofs themselves occupy Sections 6–9.

In Section 6, we give a minimal version of Boolean BI, called Minimal BBI, in which negation and falsum are disallowed; this system looks "as simple as possible", but is nonetheless proved to be undecidable.

In Section 7, we encode two-counter Minsky machines as sequents of Minimal BBI so that whenever the machine M terminates from configuration C, the corresponding sequent $\mathcal{F}_{M,C}$ is provable in Minimal BBI. The proof of this fact is given in Section 8. By soundness, the sequent $\mathcal{F}_{M,C}$ is then valid in any memory model chosen from Section 2. Then, in Section 9, we show that whenever $\mathcal{F}_{M,C}$ is valid in one of these concrete models, the machine M must terminate from configuration C. Thus it follows that "*any property between provability in Minimal* BBI *and validity in a heap-like model is undecidable*" (see Figure 1, Section 5).

In Section 10, we examine the limitations on decidable fragments of memory models imposed by our undecidability results. Of course, one option for obtaining such decidable fragments is to constrain the formula language so that sequents of the form $\mathcal{F}_{M,C}$ cannot be expressed, which can be done e.g. by excluding the multiplicative implication $-\!\!*$, as happens in [Berdine et al. 2004] and in most of the verification tools employing separation logic (cf. [Yang et al. 2008; Distefano and Parkinson 2008; Calcagno et al. 2011]).

Oddly enough, however, it also happens that validity under all *finite valuations* for atomic propositions in a heap-like model does not imply general validity in that model. Consequently, at least in some models, decidable fragments of memory models can be obtained by restricting valuations to be finite (cf. [Calcagno et al. 2001]).

In Section 11, we consider "dualising" separation models and related propositional systems based on Classical BI [Brotherston and Calcagno 2010]. Models of Classical BI differ from the models in Section 2 in that every element is required to have a unique dual; thus, for example, every Abelian group is a model of Classical BI (in which the dual of element is its group inverse), as are e.g. bit arrays (where the dual of an array is given by inverting each of its bits). Classical BI is a nonconservative extension of Boolean BI, and thus validity behaves quite differently. We show that our encoding of Minsky machines also yields undecidability of systems based on Classical BI, and undecidability of validity in associated classes of models.

This is an extended and revised version of a conference paper which appeared at LICS 2010 [Brotherston and Kanovich 2010]. Sections 6–9 contain the details of our undecidability proofs, and are mostly technical.

2. MEMORY MODELS OF THEORETICAL AND PRACTICAL INTEREST

In this section, we list a series of models employing a memory-like concept that are encountered in various areas of computer science and mathematics.

2.1. The RAM-domain model

The basic memory model we consider in this paper, that is "as simple as possible but no simpler", is the *RAM-domain model* of [Aho et al. 1974], given by

$$(\mathcal{D}, \circ, \{e_0\})$$

where $\mathcal{D}$ is the class of finite subsets of $\mathbb{N}$, to be understood as finite sets of memory locations. The operation $\circ$ denotes a partial operation for combining disjoint sets of locations:

$$d_1 \circ d_2 = \begin{cases} d_1 \cup d_2, & \text{if } d_1 \cap d_2 = \emptyset \\ \text{undefined}, & \text{otherwise} \end{cases}$$

Here, e_0 is defined to be the neutral element or *unit* of $\circ$, i.e., the empty set.

(We write the unit as a singleton set for consistency with later models employing a *set* of units rather than a single unit.)

2.2. Heap models

A *heap model* [Ahmed et al. 2003; Ishtiaq and O'Hearn 2001; Reynolds 2002] is given by

$$(H, \circ, \{e\})$$

where H is the set of *heaps* - that is, finite partial functions from an infinite set of *locations* L into a set of *record values* RV. The *domain* of a heap h is the finite set of locations on which h is defined.

The operation $\circ$ is the union of heaps when their domains are disjoint, and undefined otherwise:

$$h_1 \circ h_2 = \begin{cases} h_1 \cup h_2 & \text{if } h_1, h_2 \text{ have disjoint domains} \\ \text{undefined} & \text{otherwise} \end{cases}$$

The unit e of $\circ$ is the *empty heap* - i.e., the function with the empty domain.

2.3. Stack-and-heap models

Stack-and-heap models, as used to model Java memory [Parkinson et al. 2006], are given by

$$(S \times H,\ \circ,\ E)$$

where H is a set of *heaps* as defined in the previous subsection, and S is a set of *stacks*, which are finite partial functions from variables `Var` to stack values `Val`.

The combination operation $\circ$ for stack-heap pairs is defined as follows:

$$\langle s_1, h_1 \rangle \circ \langle s_2, h_2 \rangle = \begin{cases} \langle s_1, h_1 \circ h_2 \rangle & \text{if } s_1 = s_2 \text{ and } h_1 \circ h_2 \text{ is defined (by Section 2.2)} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Here, the set of units E consists of all pairs $\langle s, e \rangle$ where $s \in S$ and e is the empty heap.

2.4. Heap-with-permission models

In the models above, enforcing the non-overlapping of domains allows us to avoid possible "read-write" memory conflicts when combining heaps. The *heap-with-permission models* [Bornat et al. 2005] relax this strict condition; they allow overlapping heaps to be combined, providing that their values agree and "permission" for combination is permitted at all overlapping locations. Permissions are encapsulated by an underlying *permission algebra*

$$(P, \bullet, \mathbb{1})$$

which is a set P equipped with a partial commutative and associative operation $\bullet$, and a distinguished element $\mathbb{1}$ (the "write permission") such that $\mathbb{1} \bullet \pi$ is undefined for all $\pi \in P$.

Given such a permission algebra, a heap-with-permission model is then given by

$$(H, \circ, \{e\})$$

where H is the set of *heaps-with-permissions*, which are finite partial functions from an infinite set of *locations* L to a set of pairs from $RV \times P$.

We say that two such heaps h_1 and h_2 are *compatible at* ℓ 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. If h_1 and h_2 are compatible at all ℓ from the intersection of their domains, then $h_1 \circ h_2$ is defined as follows:

$$(h_1 \circ h_2)(\ell) = \begin{cases} \langle v, \pi_1 \bullet \pi_2 \rangle & \text{if } h_1(\ell) = \langle v, \pi_1 \rangle \text{ and } h_2(\ell) = \langle v, \pi_2 \rangle \\ h_1(\ell) & \text{if } h_1(\ell) \text{ defined and } h_2(\ell) \text{ undefined} \\ h_2(\ell) & \text{if } h_1(\ell) \text{ undefined and } h_2(\ell) \text{ defined} \\ \text{undefined} & \text{otherwise} \end{cases}$$

If h_1 and h_2 are not compatible at some common ℓ, then $h_1 \circ h_2$ is undefined. As in the heap models above, the unit e is the heap with the empty domain.

2.5. Stack-and-heap-with-permission models

Combining the stack-and-heap models of Section 2.3 with the heap-with-permission models of Section 2.4, we obtain the general class of *stack-and-heap-with-permission models* (cf. [Parkinson et al. 2006]), given by

$$(S \times H,\ \circ,\ E)$$

where S is a set of stacks as defined in Section 2.3, and H is a set of heaps-with-permissions (with underlying permission algebra) as defined in Section 2.4.

The combination of stack and heap-with-permission pairs is then given by the combination operations in Sections 2.3 and 2.4 in the obvious way:

$$\langle s_1, h_1 \rangle \circ \langle s_2, h_2 \rangle = \begin{cases} \langle s_1, h_1 \circ h_2 \rangle & \text{if } s_1 = s_2 \text{ and } h_1 \circ h_2 \text{ is defined (by Section 2.4)} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Again, the set of units E consists of all pairs $\langle s, e \rangle$ where $s \in S$ and e is the empty heap.

Remark 2.1. Any of the models above can be seen as a special case of a stack-and-heap-with-permission model, obtained by an appropriate choice of locations L, values RV, stacks S and permission algebra $(P, \bullet, \mathbb{1})$.

2.6. Petri net models

Models of *Petri-net markings* [Peterson 1981; Murata 1989] are given by

$$(H, \circ, \{e_0\})$$

where for some $K \subseteq \mathbb{N}$ and some set L of *locations*, the set H is the set of *markings*, defined as finite partial functions from L into K.

By taking $K = \mathbb{N}$ we model states in traditional Petri nets, with $\circ$ being the *total* operation of multiset union, i.e., $(h_1 \circ h_2)(\ell) = h_1(\ell) + h_2(\ell)$.

By taking $K = \{0, 1, .., k\}$ we model states in Petri nets with capacity k, where $\circ$ is multiset union as above, but with $(h_1 \circ h_2)(\ell)$ undefined when $h_1(\ell) + h_2(\ell) > k$.

The unit e_0 is the empty marking, i.e. the marking with empty domain.

2.7. Integer partition models

Models related to *integer partitions* [Andrews 1976] are given by

$$(H, \circ, \{e_0\})$$

where H is a set of *finite integer multisets*, with $\circ$ being the *total* operation of multiset union as in the previous subsection, and the unit e_0 being the empty multiset.

These models can be also thought of as a certain kind of memory model, where every location ℓ has an associated size or "number of blocks" $h(\ell)$.

3. PROPOSITIONAL SEPARATION LOGIC: A FORMALISM FOR REASONING ABOUT MEMORY MODELS

Here we present the language of propositional separation logic and its interpretation in the usual class of 'separation models' from [Calcagno et al. 2007], which abstract from the concrete memory models employed in the literature (cf. Section 2).

Definition 3.1. A *separation model* is a cancellative partial commutative monoid

$$(H, \circ, E)$$

where $\circ$ is a partial binary operation on H which is associative and commutative. The equality of expressions $\alpha = \beta$ means that either α and β are both undefined, or α and β are defined and equal. Cancellativity of $\circ$ means that if $h \circ h'$ is defined and $h \circ h' = h \circ h''$ then $h' = h''$.

We extend $\circ$ to a multiplication $\cdot$ on subsets of H by

$$X \cdot Y =_{\text{def}} \{h \circ h' \mid h \in X,\ h' \in Y, \text{ and } h \circ h' \text{ is defined}\} \tag{1}$$

The set of *units* E is then a subset of H such that, for all $h \in H$,

$$E \cdot \{h\} = \{h\}. \tag{2}$$

All the models from Section 2 are separation models. We allow a set of units E rather than a single unit e in Definition 3.1 in order to cover the whole spectrum of heap-like models (e.g., the stack-and-heap models in Section 2.3).

PROPOSITION 3.2. *For any $h \in H$, there is a unique $e \in E$ such that $h \circ e = h$.*

PROOF. Assume $h \circ e = h \circ e' = h$. Then by cancellativity we get $e = e'$. □

PROPOSITION 3.3. *The set of units E in a separation model $(H, \circ, E)$ 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 $\{e\}$.

PROOF. Let $e_i, e_j \in E$. Using (2) and commutativity of $\circ$, we have $\{e_i\} \cdot E = \{e_i\}$ whence $e_i \circ e_j$ is either e_i or undefined. But, by the same token, $E \cdot \{e_j\} = \{e_j\}$ whence $e_i \circ e_j$ is either e_j or undefined. Thus if $e_i \neq e_j$ then $e_i \circ e_j$ must be undefined. To see that $e_i \circ e_i = e_i$, observe that we must have $e_i \circ e = e_i$ for some $e \in E$ because $e_i \in \{e_i\} \cdot E$. □

Definition 3.4. 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 $h_1 = h_2$ because of the observation in Proposition 3.3 that $h_1 \circ h_2$ must be undefined when $h_1 \neq h_2$.)

Remark 3.5. The memory models employed in the literature (see Section 2) all have indivisible units in the sense of Definition 3.4. This natural property can be summarised by the following slogan on 'conservation of matter':

"*The empty memory cannot be split into non-empty pieces*". □

However, in contrast to the memory models, we can easily construct a separation model whose units are divisible; e.g., the model $(\mathbb{Z}, +, \{0\})$, where $+$ is addition on the integers $\mathbb{Z}$, is one such.

Definition 3.6. *Formulas* are built from an infinite set of atomic propositions p and constants $\top$, $\bot$, and I, by a unary operator $\neg$ and binary connectives $\wedge$, $\vee$, $\to$, $*$, and $-\!\!*$. (We call $*$ the *multiplicative conjunction*, with $-\!\!*$ its associated *adjoint implication* and I its *unit*.)

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

By convention, negation $\neg$ has the greatest precedence, followed by $\wedge$, $\vee$, and $*$, with the implications $\to$ and $-\!\!*$ having lowest precedence.

Definition 3.7. 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:

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

The intended meaning of any formula A under ρ is given by

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

Lemma 3.8. *Given a separation model $(H, \circ, E)$ and a valuation ρ for the model, we have the following identities:*

$$[\![p]\!]_\rho = \rho(p) \tag{3}$$
$$[\![\bot]\!]_\rho = \emptyset \tag{4}$$
$$[\![\mathrm{I}]\!]_\rho = E \tag{5}$$
$$[\![A \wedge B]\!]_\rho = [\![A]\!]_\rho \cap [\![B]\!]_\rho \tag{6}$$
$$[\![A * B]\!]_\rho = [\![A]\!]_\rho \cdot [\![B]\!]_\rho \tag{7}$$
$$[\![A \to B]\!]_\rho = \textit{largest } Z \subseteq H.\ [\![A]\!]_\rho \cap Z \subseteq [\![B]\!]_\rho \tag{8}$$
$$[\![A -\!\!* B]\!]_\rho = \textit{largest } Z \subseteq H.\ [\![A]\!]_\rho \cdot Z \subseteq [\![B]\!]_\rho \tag{9}$$

Proof. We just show the identities (7) and (9). In the case of (7), we have:

$$\begin{array}{rcl}
[\![A * B]\!]_\rho & = & \{h \mid \exists h_1, h_2.\ h = h_1 \circ h_2 \text{ and } h_1 \models_\rho A \text{ and } h_2 \models_\rho B\} \\
& = & \{h_1 \circ h_2 \mid h_1 \models_\rho A \text{ and } h_2 \models_\rho B \text{ and } h_1 \circ h_2 \text{ defined}\} \\
& = & \{h_1 \circ h_2 \mid h_1 \in [\![A]\!]_\rho \text{ and } h_2 \in [\![B]\!]_\rho \text{ and } h_1 \circ h_2 \text{ defined}\} \\
& = & [\![A]\!]_\rho \cdot [\![B]\!]_\rho
\end{array}$$

To show (9), we first show that $[\![A \multimap\!\!* B]\!]_\rho$ satisfies the required inclusion, that is

$$[\![A]\!]_\rho \cdot [\![A \multimap\!\!* B]\!]_\rho \subseteq [\![B]\!]_\rho$$

To see this, let $h \in [\![A]\!]_\rho \cdot [\![A \multimap\!\!* B]\!]_\rho$. Then there are $h_1, h_2 \in H$ with $h = h_1 \circ h_2$ and $h_1 \models_\rho A$ and $h_2 \models_\rho A \multimap\!\!* B$. In particular, this means that for any h' such that $h' \circ h_2$ is defined and $h' \models_\rho A$, we have $h' \circ h_2 \models_\rho B$. Since $h_1 \circ h_2$ is defined, we have $h_1 \circ h_2 \models_\rho B$, i.e. $h \models_\rho B$ as required.

To see that $[\![A \multimap\!\!* B]\!]_\rho$ is the *largest* set satisfying the required inclusion, we let $Z \subseteq H$ satisfy $[\![A]\!]_\rho \cdot Z \subseteq [\![B]\!]_\rho$, and show that $Z \subseteq [\![A \multimap\!\!* B]\!]_\rho$. This means showing that if $h \in Z$ then $h \models_\rho A \multimap\!\!* B$. Let $h' \in H$ be such that $h \circ h'$ is defined and $h' \models_\rho A$, i.e. $h' \in [\![A]\!]_\rho$. Since $[\![A]\!]_\rho \cdot Z \subseteq [\![B]\!]_\rho$, we have $h \circ h' \in [\![B]\!]_\rho$, i.e. $h \circ h' \models_\rho B$ as required. □

Definition 3.9. A formula A is *valid* in a separation model $(H, \circ, E)$ if for any valuation ρ, we have $[\![A]\!]_\rho = H$.

In particular, a sequent $A \vdash B$ is valid in $(H, \circ, E)$ if $[\![A]\!]_\rho \subseteq [\![B]\!]_\rho$ for any valuation ρ.

In the next theorem we establish the exact correlation between seemingly different concepts: the *indivisibility of units* in the memory models, and the *restricted $*$-weakening* in formal systems, given by axioms of the form

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

THEOREM 3.10.

(a) *For any formulas A and B, the restricted $*$-weakening principle*

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

is valid in all separation models $(H, \circ, E)$ with indivisible units.

(b) *In addition, the above correlation is exact. Namely, the basic instance of the restricted $*$-weakening principle (here p and q are atomic propositions),*

$$\mathrm{I} \wedge (p * q) \vdash p,$$

is valid in a separation model $(H, \circ, E)$ if and only if $(H, \circ, E)$ has indivisible units. □

PROOF.

(a) Let $(H, \circ, E)$ be a separation model with indivisible units, let $h \in H$ and suppose that $h \models_\rho \mathrm{I} \wedge (A * B)$. Thus $h \in E$ and there exist $h_1, h_2 \in H$ with $h = h_1 \circ h_2$ and $h_1 \models_\rho A$ and $h_2 \models_\rho B$. As $(H, \circ, E)$ has indivisible units, we have $h_1 \in E$ and $h_2 \in E$, whence by Proposition 3.3 we have $h_1 = h_2 = h$. Thus $h \models_\rho A$ and, as h was arbitrarily chosen, $\mathrm{I} \wedge (A * B) \vdash A$ is valid in this model as required.

(b) Let $(H', \circ', E)$ be a separation model whose set of units E is divisible, i.e. such that $h' \circ' h'' = e \in E$ for some $h' \notin E$.
Let ρ be a valuation with $\rho(p) = \{h'\}$ and $\rho(q) = \{h''\}$. Then we easily have that $e \models_\rho I \wedge (p * q)$ but $e \not\models_\rho p$, so $\mathrm{I} \wedge (p * q) \vdash p$ is not valid in this model as required. □

The following proposition shows that separation models with total and non-total compositions $\circ$ behave quite differently with respect to validity.

PROPOSITION 3.11. *A sequent of the form*

$$\mathrm{I} \wedge (p * q \multimap\!\!* \bot) \vdash (p \multimap\!\!* \bot) \vee (q \multimap\!\!* \bot) \tag{10}$$

is valid in a separation model $(H, \circ, E)$ if and only if the operation $\circ$ is total.

PROOF. In fact, the sequent (10) represents the following 'natural' law:

$$[\![p]\!]_\rho \cdot [\![q]\!]_\rho = \emptyset \text{ implies } ([\![p]\!]_\rho = \emptyset \text{ or } [\![q]\!]_\rho = \emptyset)$$

($\Leftarrow$) Let $(H, \circ, E)$ be a separation model with $\circ$ total, and suppose $h \models_\rho \mathrm{I} \wedge (p * q \mathrel{-\!\!*} \bot)$. Then $h \in E$ and $h \models_\rho p * q \mathrel{-\!\!*} \bot$. Using Lemma 3.8, we have $[\![p * q]\!]_\rho \subseteq [\![\bot]\!]_\rho = \emptyset$, and thus $[\![p]\!]_\rho \cdot [\![q]\!]_\rho = [\![p * q]\!]_\rho = \emptyset$.

Since $\circ$ is total, $[\![p]\!]_\rho \cdot [\![q]\!]_\rho = \emptyset$ implies that either $[\![p]\!]_\rho = \emptyset$ or $[\![q]\!]_\rho = \emptyset$. Consequently, either $h \models_\rho p \mathrel{-\!\!*} \bot$ or $h \models_\rho q \mathrel{-\!\!*} \bot$, i.e. $h \models_\rho (p \mathrel{-\!\!*} \bot) \vee (q \mathrel{-\!\!*} \bot)$ as required.

($\Rightarrow$) Let $(H, \circ, E)$ be a separation model in which $\circ$ is non-total, that is for some h' and h'', their product $h' \circ h''$ is undefined. Let ρ be a valuation with $\rho(p) = \{h'\}$ and $\rho(q) = \{h''\}$. Then $[\![p]\!]_\rho \cdot [\![q]\!]_\rho = \emptyset$ by construction. Thus, using Lemma 3.8, we have

$$\begin{array}{rcl} E \cdot [\![p * q]\!]_\rho & = & [\![p * q]\!]_\rho \\ & = & [\![p]\!]_\rho \cdot [\![q]\!]_\rho \\ & = & \emptyset \\ & = & [\![\bot]\!]_\rho \end{array}$$

Thus in particular $E \cdot [\![p * q]\!]_\rho \subseteq [\![\bot]\!]_\rho$ and, by Lemma 3.8, we have $E \subseteq [\![p * q \mathrel{-\!\!*} \bot]\!]_\rho$, which means that $[\![\mathrm{I} \wedge (p * q \mathrel{-\!\!*} \bot)]\!]_\rho \neq \emptyset$. On the other hand, by construction both $[\![p \mathrel{-\!\!*} \bot]\!]_\rho = \emptyset$ and $[\![q \mathrel{-\!\!*} \bot]\!]_\rho = \emptyset$, i.e. $[\![(p \mathrel{-\!\!*} \bot) \vee (q \mathrel{-\!\!*} \bot)]\!]_\rho = \emptyset$. We conclude that our sequent is not valid in this model as required. □

4. FORMAL SYSTEMS FOR SEPARATION LOGIC

In this section, we explore various notions of *provability* in the language of propositional separation logic, and examine their connection to validity in various classes of models. First, in Section 4.1, we introduce the core axiomatisation of the multiplicative connectives $*$, $\mathrel{-\!\!*}$ and I. Then, in Section 4.2, we examine a chain of logical systems extending this axiomatisation, including the well known *bunched logics* BI and Boolean BI.

4.1. Axiomatisation of the multiplicatives

The mathematical behaviour of separation models $(H, \circ, E)$ (see Definition 3.1), and the interpretation of the multiplicative connectives $*$, $\mathrel{-\!\!*}$ and I (Definition 3.7) naturally gives rise to a number of axioms and rules for these connectives, as follows.

(a) First, the combination operation $\circ$ is associative and commutative by definition, which means that the multiplicative conjunction $*$ should also be associative and commutative:

$$A * B \vdash B * A \qquad \text{and} \qquad A * (B * C) \vdash (A * B) * C$$

(b) Next, the set E is a unit for $\circ$, i.e., $E \circ \{h\} = \{h\}$, which means that I (representing E) should also be a unit for $*$:

$$A * \mathrm{I} \vdash A \qquad \text{and} \qquad A \vdash A * \mathrm{I}$$

(c) The "magic wand" connective $\mathrel{-\!\!*}$ is defined as the implication adjoint to the conjunction $*$, which leads to the following residuation principles for $*$ and $\mathrel{-\!\!*}$:

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

(d) Finally, we have the following monotonicity or "frame" principle for building entailments between $*$-conjunctions:

$$\frac{A \vdash B}{A * C \vdash B * C}$$

□

In fact, this collection of axioms and inference rules for the multiplicatives is originally provided by intuitionistic multiplicative linear logic, IMLL [Girard and Lafont 1987; Benton et al. 1993].

4.2. Bunched logic proof systems for separation logic

Core proof systems for propositional separation logic are provided by *bunched logic*, a class of substructural logics pioneered by O'Hearn and Pym. Specifically, the propositional proof theory of separation logic is usually considered to be given by the bunched logic Boolean BI, although other choices are also possible. Every separation model can be seen as a model of Boolean BI [Ishtiaq and O'Hearn 2001; Galmiche and Larchey-Wendling 2006].

Definition 4.1. We consider the following chain of logical systems

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

each defined as follows:

- BI, a.k.a. the *logic of bunched implications* (cf. [O'Hearn and Pym 1999; Pym 2002; Galmiche et al. 2005]) is given by:

  (A) all instances of intuitionistically valid formulas and inference rules, and

  (B) the axioms and inference rules for $*$, $-\!\!*$ and I given in Section 4.1 above.

  □

- *Boolean* BI, a.k.a. BBI (cf. [Ishtiaq and O'Hearn 2001; Galmiche and Larchey-Wendling 2006]) is obtained from BI by expanding **(A)** above to include all instances of *classically* valid propositional formulas and inference rules.

- As we shall see (Lemma 6.3), the *restricted $*$-contraction*

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

  holds in BBI, whereas the analogous *restricted $*$-weakening*

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

  does not. However, the restricted $*$-weakening exactly expresses the indivisibility of units in a separation model (see Theorem 3.10). Thus we introduce the system BBI+eW by enriching BBI with all new axioms of the form $\mathrm{I} \wedge (A * B) \vdash A$.

- Having considered restricted $*$-weakening, it is also natural to consider BBI+W, obtained by enriching BBI with the *unrestricted $*$-weakening* - that is, axioms of the form

  $$A * B \vdash A$$

  (or, equivalently, $B \vdash \mathrm{I}$). □

Proposition 4.2 (Soundness of BBI). *If A is provable in* BBI *then A is valid in all separation models.*

Proof. We just need to show that validity in any separation model is preserved by the axioms and rules of classical propositional logic and by the axioms and rules given in Section 4.1, which is an easy exercise. □

In many cases the connection between provability in the systems in Definition 4.1 and validity in the classes of corresponding separation models is not exact. E.g., BBI is not complete even for validity in the class of all partial commutative monoids [Larchey-Wendling and Galmiche 2010]. No complete natural axiomatisation is currently known for validity with respect to the class of all separation models (or an interesting subclass thereof).

COROLLARY 4.3 (SOUNDNESS OF BBI+eW). *If A is provable in* BBI+eW *then A is valid in all separation models with indivisible units.*

PROOF. Immediate from Proposition 4.2 and Theorem 3.10. □

COROLLARY 4.4. *Interpreting $\subset$ as strict inclusion between the sets of sequents provable in each system, we have:*

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

PROOF. First, note that the non-strict inclusions $\mathrm{BI} \subseteq \mathrm{BBI} \subseteq \mathrm{BBI{+}eW} \subseteq \mathrm{BBI{+}W}$ hold easily by the construction of the systems in Definition 4.1. Thus we just need to show $\mathrm{BBI{+}W} \not\subseteq \mathrm{BBI{+}eW} \not\subseteq \mathrm{BBI} \not\subseteq \mathrm{BI}$.

First, $\mathrm{BBI} \not\subseteq \mathrm{BI}$ holds because BI is conservative over intuitionistic logic [Pym 2002] and thus, e.g., the classical tautology $(p \to \bot) \to \bot \vdash p$ is invalid in BI, whereas it is valid in BBI.

$\mathrm{BBI{+}eW} \not\subseteq \mathrm{BBI}$ holds because, by Theorem 3.10, the instance $\mathrm{I} \wedge (p * q) \vdash p$ of BBI+eW's restricted $*$-weakening axiom is not valid in all separation models, and hence not provable in BBI by Proposition 4.2.

Finally, $\mathrm{BBI{+}W} \not\subseteq \mathrm{BBI{+}eW}$ holds because the instance $p \vdash \mathrm{I}$ of BBI+W's unrestricted $*$-weakening axiom is not valid in all separation models with indivisible units, and hence not provable in BBI+eW by Corollary 4.3. To see this, let $(H, \circ, E)$ be a separation model with indivisible units, and let $\rho(p) = \{h\}$ for some $h \notin E$. Then, easily, $h \models_\rho p$ but $h \not\models_\rho \mathrm{I}$. □

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 Proposition 4.2. Surprisingly, however, BBI does enjoy the *restricted* $*$-*contraction* (see Lemma 6.3 in Section 6)

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

which holds only at the level of the multiplicative unit I.

Remark 4.5. We note that the following basic instance of restricted $*$-contraction

$$\mathrm{I} \wedge p \vdash p * p$$

is *not* valid in ordinary intuitionistic BI. This can most easily be seen from the elementary Kripke semantics of BI in [Galmiche et al. 2005], whereby models are ordered partial commutative monoids $(M, \circ, e, \sqsubseteq)$, and we have the following clauses for satisfaction of formulas by $m \in M$ and valuation ρ:

$$\begin{array}{rcl} m \models_\rho \mathrm{I} & \Leftrightarrow & e \sqsubseteq m \\ m \models_\rho A \wedge B & \Leftrightarrow & m \models_\rho A \text{ and } m \models_\rho B \\ m \models_\rho A * B & \Leftrightarrow & \exists n, n' \in M.\ n \circ n' \leq m \text{ and } n \models_\rho A \text{ and } n \models_\rho B \end{array}$$

Now consider the 2-element BI-model

$$(\{e, b\}, \circ, e, \sqsubseteq)$$

where $\circ$ is given by $e \circ e = b \circ b = e$ and $e \circ b = b \circ e = b$, and $\sqsubseteq$ is given by $e \sqsubset b$. Let ρ be a valuation with $\rho(p) = \{b\}$. Then, easily, $b \models_\rho \mathrm{I} \wedge p$ but $b \not\models_\rho p * p$, i.e. $\mathrm{I} \wedge p \vdash p * p$ is not valid in this model.[1] □

In the case of BBI+eW, its restricted $*$-weakening with the restricted $*$-contraction of Lemma 6.3 induces a collapse of $\wedge$ and $*$ at the level of the unit I, as follows:

COROLLARY 4.6. *The following equivalences hold in* BBI+eW*:*

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

where the equivalence $F \equiv G$ means that both $F \vdash G$ and $G \vdash F$ are provable.

PROOF. First, $I \wedge (A * B) \vdash I \wedge A \wedge B$ is derivable using the restricted $*$-weakening. Second, using the restricted $*$-contraction and the usual contraction for $\wedge$ we can derive:

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

whence $\mathrm{I} \wedge A \wedge B \vdash (\mathrm{I} \wedge A) * (\mathrm{I} \wedge B)$ follows from weakening for $\wedge$. Finally, we can derive $(\mathrm{I} \wedge A) * (\mathrm{I} \wedge B) \vdash \mathrm{I} \wedge (A * B)$ using the equivalence $\mathrm{I} * C \equiv C$ and weakening for $\wedge$. The required equivalences then follow by transitivity of $\vdash$. □

PROPOSITION 4.7. BBI+W *collapses into ordinary classical logic.*

PROOF. Trivially, $\mathrm{I} \wedge A \vdash A$ is provable in BBI+W. Using the equivalence $\mathrm{I} * A \equiv A$ and contraction for $\wedge$ we also have $A \vdash (\mathrm{I} * A) \wedge A$ provable. Using the unrestricted $*$-weakening of BBI+W we obtain $A \vdash \mathrm{I} \wedge A$, and thus $\mathrm{I} \wedge A \equiv A$ holds in BBI+W. Using this equivalence together with Corollary 4.6, we have:

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

which also guarantees that $A \mathrel{-\!\!*} B \equiv A \to B$. Finally, using the equivalence $\mathrm{I} \wedge A \equiv A$ we have $\mathrm{I} \equiv \mathrm{I} \wedge \top \equiv \top$. Thus all the multiplicative connectives of BBI+W collapse into their classical additive equivalents. □

Remark 4.8. Both ends of the chain of logics BI $\subset$ BBI $\subset$ BBI+eW $\subset$ BBI+W given by Corollary 4.4 are in fact *decidable*. BI was shown decidable in [Galmiche et al. 2005], and BBI+W is decidable because, by Proposition 4.7, it collapses into ordinary classical logic. Thus, technically speaking, it is relatively surprising that — as we shall see in the next section — the intermediate systems BBI and BBI+eW are both in fact *undecidable*. □

5. UNDECIDABILITY OF PROPOSITIONAL SEPARATION LOGIC

In this section, we give a high-level outline of our undecidability results and their proof, the details of which occupy most of the remainder of the paper.

Figure 1 shows the overall development of our proof. As is typical, we show that a property Q is undecidable by showing that a problem already known to be undecidable reduces to the problem of deciding Q. In our case, the undecidable problem to be encoded is the termination of a two-counter, non-deterministic Minsky machine M from an arbitrary configuration C, shown as the top-centre node in Figure 1.

The proof strategy then goes as follows.

We encode M and C as a formula $\mathcal{F}_{M,C}$ (defined in Section 7) such that termination of M from C implies that $\mathcal{F}_{M,C}$ is provable in BBI (Theorem 8.3). In fact, we can obtain this result even for a *minimal* version of BBI whose Boolean connectives are restricted to $\wedge$ and $\to$: in particular, negation $\neg$ and falsum $\bot$ are disallowed. We present this "Minimal BBI" in Section 6.

[1] The key difference compared to validity in separation models is that, in the satisfaction relation for BBI in Definition 3.7, $b \models_\rho \mathrm{I} \wedge p$ forces $e = b$, while in the satisfaction relation for BI above it only forces $e \sqsubseteq b$.

By construction of the proof systems in Definition 4.1, provability of $\mathcal{F}_{M,C}$ in Minimal BBI implies its provability in BBI and BBI+eW. By our soundness results (Proposition 4.2 and Corollary 4.3) this implies validity of $\mathcal{F}_{M,C}$ in the class of all separation models, and in the class of all separation models with indivisible units, respectively. Specifically, $\mathcal{F}_{M,C}$ must be valid in *any particular* model chosen from those in Section 2.

Finally, we can complete the circle of implications — and thus the reduction from the halting problem for Minsky machines — by establishing that validity of $\mathcal{F}_{M,C}$ in *any particular* such model implies that the machine M terminates from configuration C. This is established by Theorem 9.11.

As a consequence, *every property of formulas between provability in Minimal* BBI *and validity in a model from Section 2 is undecidable*.

The right-hand side of Figure 1 essentially repeats this proof structure, but for *Classical* BI (CBI) and its models, which obey stronger properties than BBI and its separation models. Classical BI, its models and the corresponding undecidability results are presented in Section 11.

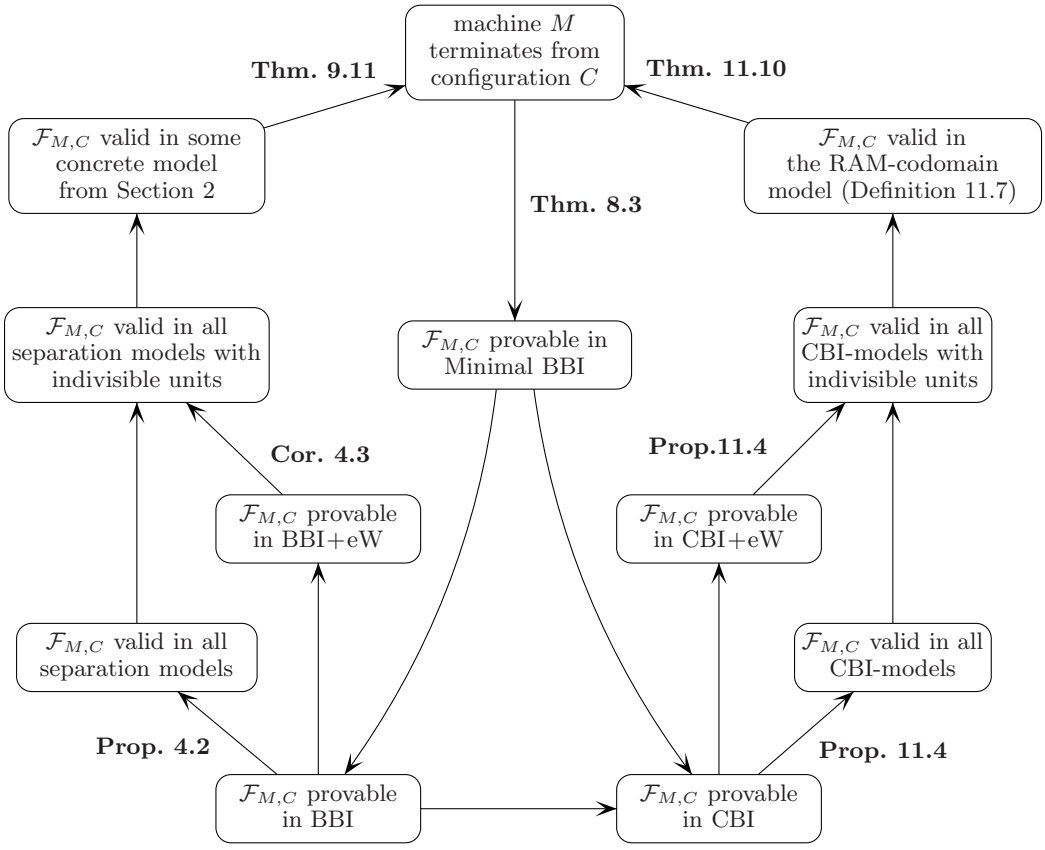

Fig. 1. Diagrammatic proof of undecidability. The arrows are implications, and $\mathcal{F}_{M,C}$ is a sequent built from machine M and configuration C. The problems at each node are all undecidable.

COROLLARY 5.1. *The following properties of formulas are undecidable, even when restricted to the language* $(\wedge, \to, \mathrm{I}, *, -\!\!*)$ *of Minimal* BBI*:*

(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 one of the concrete models in Section 2, for arbitrarily chosen locations L*, values* RV*, stacks* S*, and permission algebra* $(P, \bullet, \mathbb{1})$ *(note* L *must be infinite).*

PROOF. The termination problem for Minsky machines, which is undecidable [Minsky 1967], reduces to each of the problems above by the diagram in Figure 1. □

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

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

6. MINIMAL BOOLEAN BI - A VERY SIMPLE UNDECIDABLE PROPOSITIONAL SYSTEM

One might be tempted to think that the undecidability of BBI, in particular, is mainly an artifact of Boolean negation, which is its most visible point of difference to standard (intuitionistic) BI. In fact, this is not the case. In this section we introduce a minimal positive fragment of BBI, called *Minimal* BBI, in which the formula connectives are restricted to the minimal set of $\wedge$, $\to$, I, $*$ and $-\!\!*$. Minimal BBI is given by the axioms and rules in Figure 2, which extend the axioms and rules for $*$, $-\!\!*$ and I in Section 4 (here labelled (α_i)) with minimal rules and axioms for $\to$ and $\wedge$ (labelled (β_i)).

Remark 6.1. As foreshadowed in the previous section, Minimal BBI, even though it looks extremely simple, is still *undecidable*, notwithstanding the fact that both its components are decidable: the implication-conjunction fragment of Boolean logic is co-NP-complete, and intuitionistic multiplicative linear logic is NP-complete [Kanovich 1992].

Oddly enough, removing *Peirce's law* from Figure 2 results in (a restricted version of) standard intuitionistic BI, which is decidable. This is quite counterintuitive, since by removing Peirce's law we replace the simple Boolean logic component in Figure 2 with the implication-conjunction fragment of intuitionistic logic, the complexity of which is higher (it is PSPACE-complete [Statman 1979]). □

Note that in Figure 2 we must include the axioms and rules for $\wedge$, since $\wedge$ itself is not definable exclusively in terms of $\to$. Also, neither $\neg$ nor $\bot$ is expressible by $\wedge$ and $\to$ alone. As a consequence, it is necessary to establish that various expected principles of classical logic, present in full BBI by construction, still hold in Minimal BBI.

(α_1) $A * B \vdash B * A$

(α_2) $A * (B * C) \vdash (A * B) * C$

(α_3) $A * (A \mathrel{-\!\!*} B) \vdash B$

(α_4) $A * \mathrm{I} \vdash A$

(α_5) $A \vdash A * \mathrm{I}$

$$\frac{A \vdash B}{A * C \vdash B * C}\,(\alpha_6) \qquad \frac{A * B \vdash C}{A \vdash (B \mathrel{-\!\!*} C)}\,(\alpha_7)$$

(β_1) $A \vdash (B \to A)$

(β_2) $(A \to (B \to C)) \vdash ((A \to B) \to (A \to C))$

(β_3) $((A \to B) \to A) \vdash A$ (*Peirce's law*)

(β_4) $A \vdash (B \to (A \wedge B))$

(β_5) $A \wedge B \vdash A$

(β_6) $A \wedge B \vdash B$

$$\frac{A \qquad A \vdash B}{B}\,(\mathrm{MP})$$

Fig. 2. A minimal set of axioms and rules for Minimal Boolean BI. The axioms and rules labelled (α_i) axiomatize the behaviour of I, $*$ and $\mathrel{-\!\!*}$, while the axioms and rules labelled (β_i) axiomatize $\wedge$ and $\to$.

Proposition 6.2. *The following proof rules are derivable in Minimal* BBI*, where we write* $A \vee B$ *as an* abbreviation *for* $((B \to A) \to A)$*:*

$$\frac{}{A \vdash A}\,(\mathrm{Id}) \qquad \frac{A \vdash B \quad B \vdash C}{A \vdash C}\,(\mathrm{Tr}) \qquad \frac{A \vdash B \quad A \vdash C}{A \vdash B \wedge C}\,(\wedge)$$

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

$$\frac{A \vdash B}{B \to C \vdash A \to C}\,(\mathrm{MT}) \qquad \frac{A \vdash C \quad B \vdash C}{A \vee B \vdash C}\,(\vee) \qquad \frac{A \vdash C \quad (A \to B) \vdash C}{C}\,(\mathrm{EM})$$

Moreover, the rules $(\to)$*,* $(\mathrel{-\!\!*})$ *and* (I) *are all* reversible*, i.e. their premise and conclusion are interchangeable.*

Proof. We show how to derive each proof rule directly by the standard technique. In the following, recall that the sequent $A \vdash B$ is merely another notation for the formula $A \to B$.

(Id). By instantiating the axioms (β_1) and (β_2) we obtain, respectively,

$$A \;\vdash\; ((A \to A) \to A)$$

$$(A \to ((A \to A) \to A)) \;\vdash\; ((A \to (A \to A)) \to (A \to A))$$

Thus by (MP) we obtain $(A \to (A \to A)) \vdash (A \to A)$. Thus, since $A \vdash (A \to A)$ is an instance of (β_1), we obtain $A \vdash A$ by (MP).

(Tr). We have $B \vdash C$ provable by assumption and $(B \to C) \vdash (A \to (B \to C))$ an instance of (β_1). Thus by (MP) we obtain $A \to (B \to C)$. Now by instantiating the axiom (β_2)

we obtain:

$$(A \to (B \to C)) \vdash ((A \to B) \to (A \to C))$$

whence by (MP) we obtain $(A \to B) \vdash (A \to C)$. Then, since $A \vdash B$ is provable by assumption we obtain $A \vdash C$ by (MP) again as required.

($\wedge$). We have $A \vdash B$ by assumption and $B \vdash (C \to B \wedge C)$ an instance of (β_4). Thus by (Tr) we obtain $A \vdash (C \to B \wedge C)$. Now by instantiating (β_2) we have:

$$(A \to (C \to B \wedge C)) \vdash ((A \to C) \to (A \to B \wedge C))$$

Hence by (MP) we obtain $(A \to C) \vdash (A \to B \wedge C)$. Since we have $A \vdash C$ by assumption we obtain $A \vdash B \wedge C$ by (MP) as required.

($\to$). For the top-to-bottom direction, note that we have $A \wedge B \vdash A$ an instance of (β_5) and $A \vdash (B \to C)$ provable by assumption. Thus by (Tr) we obtain $A \wedge B \vdash (B \to C)$. Now by instantiating (β_2) we have:

$$(A \wedge B \to (B \to C)) \vdash ((A \wedge B \to B) \to (A \wedge B \to C))$$

Hence by (MP) we obtain $(A \wedge B \to B) \vdash (A \wedge B \to C)$. Since $A \wedge B \vdash B$ is an instance of (β_6), we can apply (MP) again to obtain $A \wedge B \vdash C$ as required.

For the bottom-to-top direction, we have $A \wedge B \vdash C$ by assumption. Thus, using (β_1) and (Tr), we obtain $B \vdash (A \wedge B \to C)$. By instantiating (β_2) we have

$$B \to ((A \wedge B) \to C) \vdash ((B \to A \wedge B) \to (B \to C))$$

Hence by (MP) we obtain $(B \to A \wedge B) \vdash (B \to C)$. Applying (β_1) and (Tr) again, this gives us $A \vdash ((B \to A \wedge B) \to (B \to C))$. Now we have as a further instantiation of (β_2)

$$A \to ((B \to A \wedge B) \to (B \to C)) \vdash ((A \to (B \to A \wedge B)) \to (A \to (B \to C)))$$

Hence by (MP) we obtain

$$(A \to (B \to A \wedge B)) \vdash (A \to (B \to C))$$

As $A \vdash B \to A \wedge B$ is an instance of (β_4) we obtain $A \vdash B \to C$ by (MP) as required.

($-\!\!*$). The bottom-to-top direction is immediate by (α_7). For the top-to-bottom direction, since $A \vdash (B -\!\!* C)$ is provable by assumption we have $A * B \vdash (B -\!\!* C) * B$ by the rule (α_6). Since $(B -\!\!* C) * B \vdash B * (B -\!\!* C)$ is an instance of (α_1) we then obtain $A * B \vdash B * (B -\!\!* C)$ by (Tr). Then since $B * (B -\!\!* C) \vdash C$ is an instance of (α_3) we can apply (Tr) again to obtain $A * B \vdash C$ as required.

(I). For the top-to-bottom direction, we have $B * I \vdash B$ an instance of (α_4) and $I * B \vdash B * I$ an instance of (α_1) whence $I * B \vdash B$ is provable by (Tr). Since $B \vdash C$ is provable by assumption, we obtain $I * B \vdash C$ by (Tr), whence $I \vdash (B -\!\!* C)$ is provable using the rule ($-\!\!*$) above. The reverse direction is similar.

(MT). (This rule is named for being "modus tollens-like".) First, $(B \to C) \vdash (B \to C)$ is derivable using (Id). Thus by the rule ($\to$) together with (Tr) and commutativity of $\wedge$ (easily derivable from the rule ($\wedge$)), we obtain $B \vdash ((B \to C) \to C)$. Since we have $A \vdash B$ by assumption, we obtain $A \vdash ((B \to C) \to C)$ by (Tr). Thus by applying the deduction theorem again, together with commutativity of $\wedge$ and (Tr), we have $(B \to C) \vdash (A \to C)$ as required.

($\vee$). Since we have $B \vdash C$ by assumption, we can obtain by applying the rule (MT) twice

$$((B \to A) \to A) \vdash ((C \to A) \to A)$$

By applying the rule ($\to$) we then obtain $(B \to A) \to A) \wedge (C \to A) \vdash A$. Since $A \vdash C$ is provable by assumption, we obtain $(B \to A) \to A) \wedge (C \to A) \vdash C$ by applying (Tr). By applying ($\to$) again, we have

$$((B \to A) \to A) \vdash ((C \to A) \to C)$$

Now we have $((C \to A) \to C) \vdash C$ an instance of Peirce's law (β_3). Thus we obtain $((B \to A) \to A) \vdash C$ by (Tr), which is abbreviated by $A \vee B \vdash C$.

(EM). (This rule is named for being "excluded middle-like".) We have by assumption $A \vdash C$ and $(A \to B) \vdash C$, so we obtain $A \vee (A \to B) \vdash C$ by applying the rule ($\vee$). By definition, $A \vee (A \to B)$ is equal to $((A \to B) \to A) \vdash A$, which is an instance of Peirce's law (α_3). Thus we obtain C by (MP).

□

We remark that the rules ($\wedge$) and ($\vee$) given by Proposition 6.2 immediately imply the associativity and commutativity of $\wedge$ and $\vee$, whereas associativity and commutativity of $*$ is directly expressed by axioms (α_1) and (α_2) of Minimal BBI. From now on, when we work in Minimal BBI, we implicitly treat $\wedge$, $\vee$ and $*$ as being commutative and associative.

As mentioned in Section 4, the *restricted $*$-contraction* given by $I \wedge A \vdash A * A$ holds in BBI, and in fact it holds even in Minimal BBI, which is of utmost importance for our undecidability results. This principle is obviously semantically valid, since for any $e \in E$, we have $e = e \circ e$ and thus if $e \models_\rho A$ then $e \models_\rho A * A$.

However, it is far from obvious that it is provable in BBI, and still less so in Minimal BBI. The broad idea of the BBI proof is as follows. First, from $\mathrm{I} \wedge A$ we can deduce $((\mathrm{I} \wedge A) * \mathrm{I})$ and thus $(\mathrm{I} \wedge A) * (\mathrm{I} \wedge (A \vee \neg A))$ using the usual law of excluded middle in classical logic. Since $\vee$ distributes over both $\wedge$ and $*$, the latter sequent yields $((\mathrm{I} \wedge A) * (\mathrm{I} \wedge A)) \vee ((\mathrm{I} \wedge A) * (\mathrm{I} \wedge \neg A))$ but $(\mathrm{I} \wedge A) * (\mathrm{I} \wedge \neg A)$ implies $A \wedge \neg A$, which is inconsistent.

In Minimal BBI, the situation is complicated further by the absence of $\neg$ in the language, forcing us to use Peirce's law, in the shape of the rule (EM). The following lemma spells out the full details.

LEMMA 6.3. *The restricted $*$-contraction, given by a sequent of the form*

$$\mathrm{I} \wedge A \vdash A * A \tag{11}$$

is provable in Minimal BBI.

PROOF. We make free use of the derived rules of Minimal BBI given by Proposition 6.2. According to the reversible version of the rule (I), the sequent (11) follows from

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

which is notation for the formula

$$\mathrm{I} \to ((\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A)) \tag{12}$$

Using the "excluded middle" rule (EM) in Proposition 6.2, we can derive the formula (12) from the following two sequents:

$$A \vdash \mathrm{I} \to ((\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A)) \tag{13}$$

and

$$A \to (A * A) \vdash \mathrm{I} \to ((\mathrm{I} \wedge A) \mathrel{-\!\!*} (A * A)) \tag{14}$$

We proceed by demonstrating how to derive each of (13) and (14).

- We derive the sequent (13) as follows. First, using the fact that $\mathrm{I} \wedge A \vdash A$ (axiom (β_6)) and the rule (α_6) for $*$ we have:

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

  By a similar sequence of reasoning, but additionally using commutativity of $*$ (α_1), we have $A * (\mathrm{I} \wedge A) \vdash A * A$, whereby we have by (Tr)

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

  Using the derived rule $(-\!\!*)$, we obtain:

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

  and, by means of the derived rule $(\rightarrow)$, we conclude with the desired sequent (13).

- We derive the sequent (14) in the following way. First, using $\wedge$-weakening (β_5), (β_6) and the axiom (α_4) for I, we can derive each of the following:

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

  and

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

  whence by transitivity (Tr) we obtain:

$$(\mathrm{I} \wedge (A \rightarrow (A * A))) * (\mathrm{I} \wedge A) \vdash A \tag{15}$$

  Next, using the same principles, we can derive

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

  and

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

  resulting by transitivity (Tr) in

$$(\mathrm{I} \wedge (A \rightarrow (A * A))) * (\mathrm{I} \wedge A) \vdash (A \rightarrow (A * A)) \tag{16}$$

  Using the axiom (β_2) from Figure 2 and the derived rules $(\wedge)$, $(\rightarrow)$ and (Tr) it is straightforward to derive from (15) and (16):

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

  Hence by applying the derived rules $(-\!\!*)$ we obtain

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

  and, by means of the derived rules $(\rightarrow)$, we conclude with the desired sequent (14). □

The proof principles of Proposition 6.2 and Lemma 6.3 are the main ones that are needed for correctness of our encoding of Minsky machines in the next section.

7. ENCODING MINSKY MACHINE COMPUTATIONS IN MINIMAL BBI

In this section we give our encoding of terminating computations of (non-deterministic) two-counter Minsky machines as provable sequents of Minimal BBI. That is to say, given a Minsky machine M and initial configuration C our aim is to give a sequent $\mathcal{F}_{M,C}$ such that M terminates from C just in case $\mathcal{F}_{M,C}$ is provable in Minimal BBI (as per the strategy in Section 5). First, we explain how $\mathcal{F}_{M,C}$ is constructed from M and C, and then in Section 7.1 we develop some intuition for our choice of encoding.

$$\frac{(L_i\colon c_1{+}{+}; \mathbf{goto}\ L_j;) \in M}{\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1+1, n_2\rangle} \qquad \frac{(L_i\colon c_2{+}{+}; \mathbf{goto}\ L_j;) \in M}{\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1, n_2+1\rangle}$$

$$\frac{(L_i\colon c_1{-}{-}; \mathbf{goto}\ L_j;) \in M}{\langle L_i, n_1+1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1, n_2\rangle} \qquad \frac{(L_i\colon c_2{-}{-}; \mathbf{goto}\ L_j;) \in M}{\langle L_i, n_1, n_2+1\rangle \rightsquigarrow_M \langle L_j, n_1, n_2\rangle}$$

$$\frac{(L_i\colon \mathbf{if}\ c_1{=}0\ \mathbf{goto}\ L_j;) \in M}{\langle L_i, 0, n_2\rangle \rightsquigarrow_M \langle L_j, 0, n_2\rangle} \qquad \frac{(L_i\colon \mathbf{if}\ c_2{=}0\ \mathbf{goto}\ L_j;) \in M}{\langle L_i, n_1, 0\rangle \rightsquigarrow_M \langle L_j, n_1, 0\rangle}$$

$$\frac{(L_i\colon \mathbf{goto}\ L_j;) \in M}{\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1, n_2\rangle}$$

Fig. 3. One-step semantics of two-counter Minsky machines.

Definition 7.1. A non-deterministic, two-counter *Minsky machine* M [Minsky 1967] with non-negative counters c_1, c_2 is given by a finite set of labelled *instructions* of the form:

$$\begin{array}{ll} \text{“increment } c_k \text{ by 1”} & L_i\colon c_k{+}{+}; \mathbf{goto}\ L_j; \\ \text{“decrement } c_k \text{ by 1”} & L_i\colon c_k{-}{-}; \mathbf{goto}\ L_j; \\ \text{“zero-test on } c_k\text{”} & L_i\colon \mathbf{if}\ c_k{=}0\ \mathbf{goto}\ L_j; \\ \text{“goto”} & L_i\colon \mathbf{goto}\ L_j; \end{array} \tag{17}$$

where $k \in \{1,2\}$, $i \geq 1$ and $j \geq 0$.

Non-determinism of M means that each “state label” L_i may have multiple associated instructions. The labels L_0 and L_1 are reserved for the *final* and *initial* states of M, respectively. In particular, L_0 may not label any instructions.

To cope more easily with zero-test instructions, we also add the special labels L_{-1} and L_{-2} which come equipped with the following four instructions (and which may not label any other instructions):

$$\begin{array}{ll} L_{-1}\colon c_2{-}{-}; \mathbf{goto}\ L_{-1}; & L_{-1}\colon \mathbf{goto}\ L_0; \\ L_{-2}\colon c_1{-}{-}; \mathbf{goto}\ L_{-2}; & L_{-2}\colon \mathbf{goto}\ L_0; \end{array} \tag{18}$$

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, respectively.

We write $\rightsquigarrow_M$ for the one-step relation between configurations of M; the one-step semantics of Minsky machines is given by Figure 3. We let $\rightsquigarrow_M^*$ be the reflexive-transitive closure of $\rightsquigarrow_M$, so that $\langle L, n_1, n_2\rangle \rightsquigarrow_M^* \langle L', n_1', n_2'\rangle$ 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. If $\langle L, n_1, n_2\rangle \rightsquigarrow_M^* \langle L_0, 0, 0\rangle$, we say that M *terminates from* $\langle L, n_1, n_2\rangle$, written $\langle L, n_1, n_2\rangle \Downarrow_M$.

The specific role of the special labels L_{-1} and L_{-2} is explained by the following lemma:

Lemma 7.2. *$\langle L_{-k}, n_1, n_2\rangle \Downarrow_M$ if and only if $n_k = 0$, where $k \in \{1,2\}$.*

Proof. Formally, the proof proceeds by induction on n_k; we just provide a sketch here since the details are straightforward. For the “if” direction, we simply apply the relevant instructions from the group (18). For the “only if” direction, we simply notice that, by construction, no other instructions except those in the group (18) are applicable to any configuration of the form $\langle L_{-k}, n_1, n_2\rangle$. □

Definition 7.3. In our encoding we use the following abbreviation. We fix an atomic proposition b, and henceforth define a "relative negation" by: $\mathbf{-}A =_{\text{def}} (A \mathrel{-\!\!*} b)$.

Our relative negation, defined as multiplicative implication into a constant, can be seen as a natural multiplicative analogue of standard intuitionistic negation, which is defined by $\neg A = A \to \bot$. We derive some useful properties of relative negation in Lemma 8.1.

Definition 7.4 (*Machine encoding*). For any instruction γ from Definition 7.1, we define the formula $\kappa(\gamma)$ as follows:

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

where p_1, p_2, l_{-2}, l_{-1}, l_0, l_1, l_2, ... are distinct atomic propositions (p_1 and p_2 are used to represent the counters c_1 and c_2, respectively).

Then for a Minsky machine M given by instructions $\{\gamma_1, \gamma_2, .., \gamma_t\}$ we define its encoding formula $\kappa(M)$ by

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

Definition 7.5 (*Encoding of terminating configurations*). We will encode nonnegative integers n, the contents of a Minsky machine counter, as a formula of the form

$$p^n =_{\text{def}} \underbrace{p * p * \cdots * p}_{n \text{ times}},$$

with $p^0 =_{\text{def}} \mathrm{I}$.

Accordingly, we encode a machine configuration of the form

$$C = \langle L_i, n_1, n_2 \rangle$$

as a $*$-conjunction of the propositional variable l_i with n_1 copies of p_1 and n_2 copies of p_2:

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

The fact that $\langle L_0, 0, 0 \rangle$ is a terminating configuration is encoded by the following formula:

$$\mathrm{I} \wedge \mathbf{-}l_0$$

Putting everything together, termination of M from $\langle L_i, n_1, n_2 \rangle$ will be encoded by the following sequent $\mathcal{F}_{M,C}$:

$$\mathcal{F}_{M,C} \ =_{\text{def}} \quad \kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \mathbf{-}l_0) \vdash b$$

where $\kappa(M)$ is the machine encoding given by Definition 7.4 and b is the fixed proposition letter used in our "relative negation" (Definition 7.3).

7.1. Intuition for our encoding

In accordance with our proof strategy in Section 5, our encoding is intended to achieve two complementary goals.

(1) On one hand, we have to show provability (in Minimal BBI) of $\mathcal{F}_{M,C}$ whenever M terminates from C. We construct the proof of $\mathcal{F}_{M,C}$ by induction on the length m of the terminating computation, using as sub-proofs the proofs given by induction for

terminating computations of shorter length. We give the full details of this construction in Section 8.

(2) On the other hand, for a fixed memory model from Section 2, we have to find a valuation ρ such that M terminates from C of the form $\langle L_i, n_1, n_2\rangle$ whenever $\mathcal{F}_{M,C}$ is valid under this valuation ρ, i.e. whenever

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

To obtain the required termination of M from C from this inclusion, it stands to reason to define $\rho(b)$ so as to represent *all* terminating configurations, and then to show that

$$e \in [\![\kappa(M)]\!]_\rho \quad \text{and} \quad e \in [\![\mathrm{I} \wedge \mathbin{-\!\!*} l_0]\!]_\rho,$$

where e is a unit of the memory model, resulting in the inclusion:

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

Providing that each element of $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ uniquely determines the configuration $\langle L_i, n_1, n_2\rangle$ (cf. Lemma 9.2), we can then deduce that $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

However, an additional twist to the problem is that we need a rather complicated valuation ρ in the memory models of Section 2, because of the way the *partial* composition is defined in these models. We look at this point in more detail in Sections 9.2 and 10. □

We note that a direct adaptation of the encoding of Minsky machines developed for full linear logic in [Kanovich 1995] does not work properly for separation logic due to the differences between linear logic and separation logic (discussed further in Section 12).

Roughly speaking, in the encoding of [Kanovich 1995], each step in the derivation of the "encoding sequent" corresponds to a single *forward step* from C to C' within the computation of machine M (which is not necessarily terminating).

In contrast, in our encoding, each step in the derivation corresponds to a *'backward move'* from the class of all terminating computations starting from C' to the class of *longer* terminating computations starting from C.

We illustrate this point in more detail.

Given a memory model, suppose that a valuation ρ is chosen such that each element of $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ uniquely determines the configuration $\langle L_i, n_1, n_2\rangle$ (cf. Lemma 9.2).

As discussed above, to guarantee the faithfulness of our encoding, we need to check that for such a valuation ρ, the encoding guarantees that $e \in [\![\kappa(M)]\!]_\rho$, that is, for each of the machine instruction γ we can provide $e \in [\![\kappa(\gamma)]\!]_\rho$.

For any instruction γ, the formula $\kappa(\gamma)$ is of the form $(\mathbin{-\!\!*} A \mathbin{-\!\!*} \mathbin{-\!\!*} B)$ where A is of the form l_j, $(l_j \vee l_{-k})$ or $(l_j * p_k)$, and B is of the form l_i or $(l_i * p_k)$. Thus, recalling that $\mathbin{-\!\!*} A =_{\mathrm{def}} (A \mathbin{-\!\!*} b)$, and $\mathbin{-\!\!*} B =_{\mathrm{def}} (B \mathbin{-\!\!*} b)$, the statement $e \in [\![\kappa(\gamma)]\!]_\rho$ is equivalent to the inclusion

$$[\![A \mathbin{-\!\!*} b]\!]_\rho \;\subseteq\; [\![B \mathbin{-\!\!*} b]\!]_\rho$$

Since, by Lemma 3.8, the statement $d \in [\![A \mathbin{-\!\!*} b]\!]_\rho$ means that $[\![A]\!]_\rho \cdot \{d\} \subseteq \rho(b)$, this inclusion can be rewritten as the rule that for all d,

$$\frac{[\![A]\!]_\rho \cdot \{d\} \;\subseteq\; \rho(b)}{[\![B]\!]_\rho \cdot \{d\} \;\subseteq\; \rho(b)} \tag{19}$$

For instance, in the case of the increment instruction

$$L_i\colon c_1{+}{+}; \mathbf{goto}\ L_j;, \tag{20}$$

we must show that the following rule is correct for any d:

$$\frac{[\![l_j * p_1]\!]_\rho \cdot \{d\} \subseteq \rho(b)}{[\![l_i]\!]_\rho \cdot \{d\} \subseteq \rho(b)}$$

Our choice of valuation (cf. Lemma 9.2) ensures that if $[\![l_j * p_1]\!]_\rho \cdot \{d\} \subseteq \rho(b)$, then d must belong to a set of the form $[\![p_1^{n_1} * p_2^{n_2}]\!]_\rho$, so that this rule becomes

$$\frac{[\![l_j * p_1^{n_1+1} * p_2^{n_2}]\!]_\rho \subseteq \rho(b)}{[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \subseteq \rho(b)}$$

or, paraphrasing,

$$\frac{\text{“}M \text{ terminates from the configuration } \langle L_j, n_1+1, n_2\rangle\text{”}}{\text{“}M \text{ terminates from the configuration } \langle L_i, n_1, n_2\rangle\text{”}}$$

which is a correct statement according to the increment instruction (20) at hand. Other instructions are treated similarly.

8. FROM COMPUTATIONS TO MINIMAL BBI PROOFS

Here, we show in detail that the sequent $\mathcal{F}_{M,C}$, defined in the previous section, is provable in Minimal BBI whenever M terminates on C. The converse direction will be established in Section 9.

First, the following lemma derives useful properties of $-\!\!*A$ which illustrate its behaviour as a multiplicative analogue of intuitionistic negation, and will be essential to the proof of our main result.

LEMMA 8.1. *The following sequents and proof rules are derivable in Minimal* BBI*:*

(a) $A \vdash (A -\!\!* b) -\!\!* b$

(b) $(((A -\!\!* b) -\!\!* b) -\!\!* b) \vdash A -\!\!* b$

(c) $A * (B -\!\!* b) \vdash (A -\!\!* b) -\!\!* b$

(d) $\dfrac{A * B \vdash C}{A * ((B -\!\!* b) -\!\!* b) \vdash (C -\!\!* b) -\!\!* b}$

(e) $\dfrac{A * B \vdash D \qquad A * C \vdash D}{A * ((B \vee C) -\!\!* b) \vdash D -\!\!* b}$ □

PROOF. We derive each sequent or proof rule directly in Minimal BBI, making use of the derived rules given by Proposition 6.2.

(a) According to the Minimal BBI axiom (α_3), we have $A * (A -\!\!* b) \vdash b$ provable. Thus by applying the derived rule $(-\!\!*)$ we obtain $A \vdash ((A -\!\!* b) -\!\!* b)$, i.e. $A \vdash (A -\!\!* b) -\!\!* b$ as required.

(b) According to the Minimal BBI axiom (α_3), we have $(((A -\!\!* b) -\!\!* b) -\!\!* b) * ((A -\!\!* b) -\!\!* b) \vdash b$ provable. Also, $A \vdash (A -\!\!* b) -\!\!* b$ is provable by part (a) above. Thus by the derived rule (Tr) we obtain $A * ((A -\!\!* b) -\!\!* b) \vdash b$. By applying $(-\!\!*)$ again we obtain $((A -\!\!* b) -\!\!* b) \vdash A -\!\!* b$, i.e. $(((A -\!\!* b) -\!\!* b) -\!\!* b) \vdash A -\!\!* b$ as required.

(c) We have $(B -\!\!* b) * ((B -\!\!* b) -\!\!* b) \vdash A -\!\!* b$ an instance of the Minimal BBI axiom (α_3), which can then be rewritten as $(B -\!\!* b) * ((B -\!\!* b) -\!\!* b) \vdash A -\!\!* b$. By applying $(-\!\!*)$ we obtain

$A * (\mathbf{-}B \mathrel{-\!\!*} \mathbf{-}A) \vdash \mathbf{-}B \mathrel{-\!\!*} b$, i.e. $A * (\mathbf{-}B \mathrel{-\!\!*} \mathbf{-}A) \vdash \mathbf{--}B$ as required.

(d) We have $A * B \vdash C$ by assumption and $C \vdash \mathbf{--}C$ by part (a) above, whence by (Tr) we obtain $A * B \vdash \mathbf{--}C$. By applying $(\mathrel{-\!\!*})$, we obtain $A * \mathbf{-}C \vdash \mathbf{-}B$. Since $\mathbf{-}B \vdash \mathbf{---}B$ is derivable by part (a), we obtain $A * \mathbf{-}C \vdash \mathbf{---}B$ by (Tr). By applying $(\mathrel{-\!\!*})$ again, this can be transformed to the required $A * \mathbf{--}B \vdash \mathbf{--}C$.

(e) By applying $(\mathrel{-\!\!*})$ to the premises $A * B \vdash D$ and $A * C \vdash D$ we have $B \vdash (A \mathrel{-\!\!*} D)$ and $C \vdash (A \mathrel{-\!\!*} D)$. Thus by applying the derived rule $(\vee)$ we obtain $B \vee C \vdash (A \mathrel{-\!\!*} D)$. Applying $(\mathrel{-\!\!*})$, we get $A * (B \vee C) \vdash D$. Finally, we apply the derived rule given by part (d) above to obtain $A * \mathbf{--}(B \vee C) \vdash \mathbf{--}D$ as required.

□

In the process of computation each instruction of a Minsky machine can be reused an unlimited number of times. In order for us to simulate Minsky computations within Minimal BBI proofs, it is essential that the formulas $\kappa(\gamma)$ representing reusable instructions can be duplicated as needed, which follows from the restricted $*$-contraction given by Lemma 6.3.

LEMMA 8.2. *For each instruction γ of a machine M, the sequent*

$$\kappa(M) \vdash \kappa(M) * \kappa(\gamma)$$

is derivable in Minimal BBI.

PROOF. Writing $M = \{\gamma_1, \gamma_2, .., \gamma_t\}$, we have $\kappa(M) = (\mathrm{I} \wedge \bigwedge_{i=1}^{t} \kappa(\gamma_i))$ and $\gamma = \gamma_j$ for some $1 \leq j \leq t$. First, using the derived rules (Id) and $(\wedge)$, and the axiom (β_5) for $\wedge$, we can derive

$$\kappa(M) \vdash \mathrm{I} \wedge \kappa(M)$$

Next, we have as an instance of restricted $*$-contraction (Lemma 6.3)

$$\mathrm{I} \wedge \kappa(M) \vdash \kappa(M) * \kappa(M)$$

We have $\kappa(M) \vdash \kappa(\gamma_j)$ derivable in Minimal BBI by using weakening for $\wedge$ (given by the axioms (β_5) and (β_6)) and the derived transitivity rule (Tr) of Proposition 6.2. Thus by applying the rule (α_6) we obtain

$$\kappa(M) * \kappa(M) \vdash \kappa(M) * \kappa(\gamma_j)$$

Thus using the transitivity rule (Tr) we obtain $\kappa(M) \vdash \kappa(M) * \kappa(\gamma_j)$ as required. □

THEOREM 8.3. *Let M be a Minsky machine, and for a configuration C of the form $\langle L_i, n_1, n_2 \rangle$, suppose that M terminates from C. Then the following sequent $\mathcal{F}_{M,C}$ is derivable in Minimal* BBI*:*

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge \mathbf{-}l_0) \vdash b \tag{21}$$

PROOF. We make free use of the derived rules of Minimal BBI given by Proposition 6.2. First we show that it suffices to derive the sequent:

$$\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} \vdash \mathbf{--}l_0 \tag{22}$$

To see this, suppose we have $A \vdash \mathbf{--}l_0$, whence by applying the derived rule $(\mathrel{-\!\!*})$ we obtain $\mathbf{-}l_0 \vdash (A \mathrel{-\!\!*} b)$. Since $\mathrm{I} \wedge \mathbf{-}l_0 \vdash \mathbf{-}l_0$ is an instance of the axiom (β_6) we obtain $\mathrm{I} \wedge \mathbf{-}l_0 \vdash (A \mathrel{-\!\!*} b)$ by applying the derived rule (Tr). By applying $(\mathrel{-\!\!*})$ again we obtain $A * (\mathrm{I} \wedge \mathbf{-}l_0) \vdash b$ as required.

We show that (22) is derivable in Minimal BBI 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 must derive:

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

This is easily derivable using the equivalence $I * A \equiv A$, weakening for $\wedge$ and part (a) of Lemma 8.1.

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 a goto instruction, and for increment, decrement and zero-test instructions with the counter variable $k = 1$; the cases for $k = 2$ are similar.

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

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

By the induction hypothesis, we can derive

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

By applying part (d) of Lemma 8.1, we obtain

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

Since $\mathbf{----}l_0 \vdash \mathbf{--}l_0$ is derivable by part (b) of Lemma 8.1, we obtain by (Tr)

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

Now since $(\mathbf{-}l_j \mathrel{-\!\!*} \mathbf{-}l_i) * l_i \vdash \mathbf{--}l_j$ is provable by part (c) of Lemma 8.1, we obtain by (Tr) and the derived rule $(\mathrel{-\!\!*})$

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

Since $\gamma \in M$ and $\kappa(\gamma) = (\mathbf{-}l_j \mathrel{-\!\!*} \mathbf{-}l_i)$, we have $\kappa(M) \vdash \kappa(M) * (\mathbf{-}l_j \mathrel{-\!\!*} \mathbf{-}l_i)$ provable by Lemma 8.2. Thus we derive by (Tr) and $(\mathrel{-\!\!*})$:

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

which completes the case.

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 sequent is derivable:

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

By the induction hypothesis, we can derive

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

By applying part (d) of Lemma 8.1, we obtain

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

Since $\mathbf{----}l_0 \vdash \mathbf{--}l_0$ is derivable by part (b) of Lemma 8.1, we obtain by (Tr)

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

Now since $(\mathbf{-}(l_j * p_1) \mathrel{-\!\!*} \mathbf{-}l_i) * l_i \vdash \mathbf{--}(l_j * p_1)$ is provable by part (c) of Lemma 8.1, we obtain by (Tr) and $(\mathrel{-\!\!*})$

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

As $\gamma \in M$ and $\kappa(\gamma) = (\mathsf{-}(l_j{*}p_1) \mathrel{-\!\!*} \mathsf{-}l_i)$, we have $\kappa(M) \vdash \kappa(M) * (\mathsf{-}(l_j * p_1) \mathrel{-\!\!*} \mathsf{-}l_i)$ provable by Lemma 8.2. Thus we derive by (Tr) and ($-\!\!*$)

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

which completes the case.

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

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

By the induction hypothesis, we can derive

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

Using parts (b), (c) and (d) of Lemma 8.1 together with (Tr) and ($-\!\!*$) in a similar way to previous cases, we obtain

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

Since $\gamma \in M$ and $\kappa(\gamma) = (\mathsf{-}l_j \mathrel{-\!\!*} \mathsf{-}(l_i * p_1))$, we have $\kappa(M) \vdash \kappa(M) * (\mathsf{-}l_j \mathrel{-\!\!*} \mathsf{-}(l_i * p_1))$ provable by Lemma 8.2. Thus we derive by (Tr) and ($-\!\!*$), as required:

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

Case $\gamma =$ $(L_i\colon \mathbf{if}\ c_1{=}0\ \mathbf{goto}\ L_j;)$**.**
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 the following sequent:

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

First, by the induction hypothesis we can derive

$$\kappa(M) * l_j * \mathrm{I} * p_2^{n_2} \vdash \mathsf{--}l_0 \tag{23}$$

Next, note that by Lemma 7.2 we have $\langle L_{-1}, 0, n_2\rangle{\Downarrow_M}$. Since the computation of $\langle L_{-1}, 0, n_2\rangle \rightsquigarrow_M^* \langle L_0, 0, 0\rangle$ involves only decrement and goto instructions (see group (18) in Definition 7.1) by construction, the cases already considered above are sufficient to establish the present theorem for the configuration $\langle L_{-1}, 0, n_2\rangle$. Thus we can also derive

$$\kappa(M) * l_{-1} * \mathrm{I} * p_2^{n_2} \vdash \mathsf{--}l_0 \tag{24}$$

Thus by applying part (e) of Lemma 8.1 we obtain from (23) and (24)

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

Since $\mathsf{----}l_0 \vdash \mathsf{--}l_0$ is derivable by part (b) of Lemma 8.1, we obtain by (Tr)

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

Then as $(\mathsf{-}(l_j \vee l_{-1}) \mathrel{-\!\!*} \mathsf{-}l_i) * l_i \vdash \mathsf{--}(l_j \vee l_{-1})$ is provable by part (c) of Lemma 8.1, we obtain by (Tr) and ($-\!\!*$)

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

Since $\gamma \in M$ and $\kappa(\gamma) = (\mathsf{-}(l_j \vee l_{-1}) \mathrel{-\!\!*} \mathsf{-}l_i)$, we have $\kappa(M) \vdash \kappa(M) * (\mathsf{-}(l_j \vee l_{-1}) \mathrel{-\!\!*} \mathsf{-}l_i)$ provable by Lemma 8.2. Thus we derive by (Tr) and ($-\!\!*$)

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

which completes the case, and the proof. □

9. FROM VALIDITY TO TERMINATING COMPUTATIONS

In this section, our goal is to show that for each of the concrete memory models given in Section 2, we have $\langle L_i, n_1, n_2\rangle\Downarrow_M$ whenever the following sequent $\mathcal{F}_{M,\langle L_i,n_1,n_2\rangle}$ is valid:

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

For the sake of perspicuity, we establish this property first for the simplest such memory model, the *RAM-domain model* from Section 2.1. Then, we show how to extend our approach to any of the memory models from Section 2. In particular, we extend our approach to the most general stack-and-heap models from Section 2.5, of which the other models in Section 2.1-Section 2.4 can be seen as special instances (see Remark 2.1). (The RAM-domain model case also suffices to cover the Petri net models of Section 2.6 and the integer partition models of Section 2.7.)

9.1. Valuation for the RAM-domain model

Definition 9.1. Given a machine M, we introduce the following valuation ρ_0 for the RAM-domain model $(\mathcal{D}, \circ, \{e_0\})$ of Section 2.1:

$$\begin{array}{rcccl}
\rho_0(p_1) & = & \{\ \{6\}, \{10\}, \{14\}, \{18\}\ \ldots, \} & = & \{\ \{2(2m+1)\} \mid m \geq 1\ \} \\
\rho_0(p_2) & = & \{\ \{12\}, \{20\}, \{28\}, \{36\}\ \ldots\} & = & \{\ \{4(2m+1)\} \mid m \geq 1\ \} \\
\rho_0(l_i) & = & \{\ \{2^{i+5}\cdot 3\}, \{2^{i+5}\cdot 5\}, \{2^{i+5}\cdot 7\}, \ldots\} & = & \{\ \{2^{i+5}\cdot(2m+1)\} \mid m \geq 1\ \}
\end{array}$$

where $i \geq -2$, and

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

where b is the distinguished propositional variable introduced in Definition 7.3. (The reason behind our choice for $\rho_0(b)$ is given in Section 7.1.)

LEMMA 9.2. *For any $d \in \mathcal{D}$, $n \in \mathbb{N}$ and $k \in \{1, 2\}$, Definition 9.1 guarantees that $d \in [\![p_k^n]\!]_{\rho_0}$ if and only if d consists of exactly n distinct numbers of the form $2^k \cdot (2m+1)$ (where $m \geq 1$).*

Thus $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ is not empty, and each of the elements of $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ uniquely determines the configuration $\langle L_i, n_1, n_2\rangle$.

PROOF. The first part is by induction on n. In the case $n = 0$ we have $[\![p_k^0]\!]_{\rho_0} = [\![\mathrm{I}]\!]_{\rho_0} = \{e_0\}$, where e_0 is $\emptyset$. Thus $d \in [\![p_k^0]\!]_{\rho_0}$ iff $d = \emptyset$ iff d consists of zero numbers of the form $2^k \cdot (2m+1)$.

In the case $n > 0$ we have, using Lemma 3.8:

$$\begin{array}{rcl}
[\![p_k^n]\!]_{\rho_0} & = & [\![p_k * p_k^{n-1}]\!]_{\rho_0} \\
& = & [\![p_k]\!]_{\rho_0} \cdot [\![p_k^{n-1}]\!]_{\rho_0} \\
& = & \{d' \circ d'' \mid d' \in \rho_0(p_k), d'' \in [\![p_k^{n-1}]\!]_{\rho_0}\}
\end{array}$$

Thus $d \in [\![p_k^n]\!]_{\rho_0}$ iff it is the union of disjoint sets d' and d'' with $d' \in \rho_0(p_k)$ and $d'' \in [\![p_k^{n-1}]\!]_{\rho_0}$. By induction hypothesis, $d'' \in [\![p_k^{n-1}]\!]_{\rho_0}$ iff d'' consists of exactly $n-1$ distinct numbers of the form $2^k \cdot (2m+1)$, and by construction it holds that $d' \in \rho_0(p_k)$ iff $d' = \{2^k \cdot (2m'+1)\}$ for some m'. Since d' and d'' must be disjoint, we have $d \in [\![p_k^n]\!]_{\rho_0}$ iff d consists of exactly n numbers of the form $2^k \cdot (2m+1)$, as required.

For the second part of the lemma, we have, using Lemma 3.8:

$$\begin{array}{rcl}
[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} & = & [\![l_i]\!]_{\rho_0} \cdot [\![p_1^{n_1}]\!]_{\rho_0} \cdot [\![p_2^{n_2}]\!]_{\rho_0} \\
& = & \{d_1 \circ d_2 \circ d_3 \mid d_1 \in \rho_0(l_i), d_2 \in [\![p_1^{n_1}]\!]_{\rho_0}, d_3 \in [\![p_2^{n_2}]\!]_{\rho_0}\}
\end{array}$$

Thus $d \in [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ iff it is the union of disjoint sets d_1, d_2 and d_3 with $d_1 \in \rho_0(l_i)$, $d_2 \in [\![p_1^{n_1}]\!]_{\rho_0}$ and $d_3 \in [\![p_2^{n_2}]\!]_{\rho_0}$. By the first part of the lemma, d_2 and d_3 consist of n_1 distinct numbers of form $2(2m+1)$ and n_2 distinct numbers of the form $4(2m+1)$, respectively, whereas by construction $d_1 = \{2^{i+5} \cdot (2m'+1)\}$ for some $m' \geq 1$ and $i \geq -2$.

To see that the decomposition of d into $d_1 \circ d_2 \circ d_3$ exists and is unique, we must show that d_1, d_2, d_3 are non-overlapping. To see this, suppose that $2^i \cdot (2m+1) = 2^j \cdot (2m'+1)$ where $i \neq j$, and assume without loss of generality that $j > i$. Then by simple manipulation we have $2m = 2^{j-i+1}m' + 2^{j-i} - 1$, but then $2m$ is even while $2^{j-i+1}m' + 2^{j-i} - 1$ is odd, contradiction. Thus $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ is not empty, and each of its elements uniquely determines L_i, n_1 and n_2, and hence the configuration $\langle L_i, n_1, n_2 \rangle$. □

9.2. Do we need infinite valuations ?

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 can show that for all sufficiently large n,

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

which obstructs us in uniquely representing the contents n of the Minsky machine counter c_k by the formula p_k^n. We discuss the impact upon decidability of a restriction to finite valuations in Section 10.

9.3. Formal details

Now we prove the basic lemma.

LEMMA 9.3. *$e_0 \models_{\rho_0} \kappa(M)$ for any machine M.*

PROOF. Writing $M = \{\gamma_1, \ldots, \gamma_t\}$, we have by Definition 3.7:

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

Thus it suffices to show that $e_0 \models_{\rho_0} \kappa(\gamma)$ for any instruction γ. Recalling that $\mathbf{-}A =_{\mathrm{def}} (A \mathrel{-\!\!*} b)$, we shall make use of the fact that, for any $d \in \mathcal{D}$ and any formula A,

$$d \models_{\rho_0} \mathbf{-}A \;\Leftrightarrow\; \forall d'.\ (d, d' \text{ disjoint and } d' \models_{\rho_0} A) \text{ implies } d \circ d' \in \rho_0(b) \tag{25}$$

We present the cases for a goto instruction, and for increment, decrement and zero-test instructions with counter variable $k = 1$; the cases where $k = 2$ are similar.

Case $\gamma = (L_i$: goto $L_j;)$. We have $\kappa(\gamma) = (\mathbf{-}l_j \mathrel{-\!\!*} \mathbf{-}l_i)$, so must show $e_0 \models_{\rho_0} \mathbf{-}l_j \mathrel{-\!\!*} \mathbf{-}l_i$. Using the fact that $e_0 \circ d = d$, this amounts to showing, for any $d \in \mathcal{D}$,

$$d \models_{\rho_0} \mathbf{-}l_j \text{ implies } d \models_{\rho_0} \mathbf{-}l_i$$

To show this implication, suppose $d \models_{\rho_0} \mathbf{-}l_j$. Since $d \in \mathcal{D}$ is a finite set whereas $\rho_0(l_j)$ contains infinitely many disjoint sets by construction, there must be some (in fact infinitely many) $d_j \in \rho_0(l_j)$ such that d and d' are disjoint. Thus, using the equivalence (25) and Lemma 3.8, we have:

$$\begin{aligned} d \circ d_j \;&\in\; \rho_0(b) \\ &= \textstyle\bigcup_{\langle L_i, n_1, n_2 \rangle \Downarrow_M} [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \\ &= \textstyle\bigcup_{\langle L_i, n_1, n_2 \rangle \Downarrow_M} \rho_0(l_i) \cdot [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \end{aligned}$$

By construction of ρ_0 and using the fact that $d_j \in \rho_0(l_j)$, we must have $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ for some n_1, n_2. Moreover, by Lemma 9.2, $d \circ d_j$ uniquely determines $\langle L_j, n_1, n_2 \rangle$, so we must

have $\langle L_j, n_1, n_2\rangle \Downarrow_M$. Since $\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1, n_2\rangle$ by applying the goto instruction γ, we then have $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

Now to see that $d \models_{\rho_0} \mathbin{-\!\!*} l_i$, let $d' \in \mathcal{D}$ be disjoint from d with $d' \in \rho_0(l_i)$. By (25) it suffices to show that $d \circ d' \in \rho_0(b)$. Since $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$, we have $d \circ d' \in [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ where $\langle L_i, n_1, n_2\rangle \Downarrow_M$, and thus $d \circ d' \in \rho_0(b)$ as required.

Case $\gamma = (L_i\colon c_1{+}{+}; \textbf{goto } L_j;)$**.** We have $\kappa(\gamma) = (\mathbin{-\!\!*}(l_j * p_1) \mathbin{-\!\!*} \mathbin{-\!\!*} l_i)$. As in the previous case, to show $e_0 \models_{\rho_0} \kappa(\gamma)$, we must show that, for any $d \in \mathcal{D}$,

$$d \models_{\rho_0} \mathbin{-\!\!*}(l_j * p_1) \text{ implies } d \models_{\rho_0} \mathbin{-\!\!*} l_i$$

Suppose $d \models_{\rho_0} \mathbin{-\!\!*}(l_j * p_1)$. Since $\rho_0(l_j)$ and $\rho_0(p_1)$ each contain infinitely many disjoint sets, we can find $d_j \in \rho_0(l_j)$ and $d_1 \in \rho_0(p_1)$ such that d, d_j and d_1 are disjoint. Thus $d_j \circ d_1 \models_{\rho_0} l_j * p_1$. Using the equivalence (25) and Lemma 3.8, we have:

$$\begin{array}{rcl} d \circ d_j \circ d_1 & \in & \rho_0(b) \\ & = & \bigcup_{\langle L_i, n_1, n_2\rangle \Downarrow_M} \rho_0(l_i) \cdot [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \end{array}$$

By construction of ρ_0 and using the fact that $d_j \in \rho_0(l_j)$, we must have $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ for some n_1, n_2. Moreover, by Lemma 9.2, and using the fact that $d_1 \in \rho_0(p_1)$, the set $d \circ d_j \circ d_1$ uniquely determines $\langle L_j, n_1 + 1, n_2\rangle$, whence $\langle L_j, n_1 + 1, n_2\rangle \Downarrow_M$. Since $\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1 + 1, n_2\rangle$ via the increment instruction γ, we have $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

Now to see that $d \models_{\rho_0} \mathbin{-\!\!*} l_i$, let $d' \in \mathcal{D}$ be disjoint from d with $d' \in \rho_0(l_i)$. By (25) it suffices to show that $d \circ d' \in \rho_0(b)$. Since $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$, we have $d \circ d' \in [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ where $\langle L_i, n_1, n_2\rangle \Downarrow_M$, and thus $d \circ d' \in \rho_0(b)$ as required.

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

$$d \models_{\rho_0} \mathbin{-\!\!*} l_j \text{ implies } d \models_{\rho_0} \mathbin{-\!\!*}(l_i * p_1)$$

Now suppose $d \models_{\rho_0} \mathbin{-\!\!*} l_j$. As in the previous cases, we can find $d_j \in \rho_0(l_j)$ such that d_j and d are disjoint. Thus, using the equivalence (25), we have:

$$\begin{array}{rcl} d \circ d_j & \in & \rho_0(b) \\ & = & \bigcup_{\langle L_i, n_1, n_2\rangle \Downarrow_M} \rho_0(l_i) \cdot [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \end{array}$$

By the same argument as in previous cases, this implies that there exist n_1 and n_2 such that $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$, and $\langle L_j, n_1, n_2\rangle \Downarrow_M$. Since $\langle L_i, n_1 + 1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1, n_2\rangle$ via the decrement instruction γ, we have $\langle L_i, n_1 + 1, n_2\rangle \Downarrow_M$.

Now let $d' \in \mathcal{D}$ be disjoint from d with $d' \models_{\rho_0} l_i * p_1$. By (25) it suffices to show that $d \circ d' \in \rho_0(b)$. First, we have $d' = d_i \circ d_1$ where $d_i \in \rho_0(l_i)$ and $d_1 \in \rho_0(p_1)$. Since $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$, we have $d \circ d_i \circ d_1 \in [\![l_i * p_1^{n_1+1} * p_2^{n_2}]\!]_{\rho_0}$ where $\langle L_i, n_1 + 1, n_2\rangle \Downarrow_M$. Thus $d \circ d_i \circ d_1 = d \circ d' \in \rho_0(b)$ as required.

Case $\gamma = (L_i\colon \textbf{if } c_1{=}0 \textbf{ goto } L_j;)$**.** In this case, we have $\kappa(\gamma) = (\mathbin{-\!\!*}(l_j \vee l_{-1}) \mathbin{-\!\!*} \mathbin{-\!\!*} l_i)$. To show $e_0 \models_{\rho_0} \kappa(\gamma)$, we must show that, for any $d \in \mathcal{D}$,

$$d \models_{\rho_0} \mathbin{-\!\!*}(l_j \vee l_{-1}) \text{ implies } d \models_{\rho_0} \mathbin{-\!\!*} l_i$$

Now suppose that $d \models_{\rho_0} \mathbin{-\!\!*}(l_j \vee l_{-1})$. As in previous cases, we can find $d_j \in \rho_0(l_j)$ and $d_{-1} \in \rho_0(l_{-1})$ such that d, d_j and d_{-1} are all disjoint. Note that both $d_j \models_{\rho_0} l_j \vee l_{-1}$ and $d_{-1} \models_{\rho_0} l_j \vee l_{-1}$. Thus, using the equivalence (25), we have:

$$\begin{array}{rcl} d \circ d_j, d \circ d_{-1} & \in & \rho_0(b) \\ & = & \bigcup_{\langle L_i, n_1, n_2\rangle \Downarrow_M} \rho_0(l_i) \cdot [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0} \end{array}$$

By the same argument as in previous cases, there are n_1 and n_2 such that $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$, and we have both $\langle L_j, n_1, n_2\rangle \Downarrow_M$ and $\langle L_{-1}, n_1, n_2\rangle \Downarrow_M$. However, by Lemma 7.2, $\langle L_{-1}, n_1, n_2\rangle \Downarrow_M$ implies $n_1 = 0$. Thus $\langle L_i, n_1, n_2\rangle \rightsquigarrow_M \langle L_j, n_1, n_2\rangle \Downarrow_M$ by applying the zero-test instruction γ, whence $\langle L_i, n_1, n_2\rangle \Downarrow_M$. The argument that $d \models_{\rho_0} -\!\!* l_i$ is then exactly as in the goto and increment instruction cases. This completes the case, and the proof. □

LEMMA 9.4. *For any machine M we have $e_0 \models_{\rho_0} \mathrm{I} \wedge -\!\!* l_0$.*

PROOF. We trivially have $e_0 \models_{\rho_0} \mathrm{I}$. To show $e_0 \models_{\rho_0} -\!\!* l_0$, taking into account that $e_0 \circ x = x$, we must show that $x \models_{\rho_0} l_0$ implies $x \models_{\rho_0} b$, i.e. that $\rho_0(l_0) \subseteq \rho_0(b)$. This is immediate by construction of $\rho_0(b)$, since $\langle L_0, 0, 0\rangle \Downarrow_M$ by definition and $\rho_0(l_0) = [\![l_0 * \mathrm{I} * \mathrm{I}]\!]_{\rho_0} = [\![l_0 * p_1^0 * p_2^0]\!]_{\rho_0}$. □

THEOREM 9.5. *For any machine M, if the sequent*

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

is valid in the RAM-domain model $(\mathcal{D}, \circ, \{e_0\})$, then $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

PROOF. By the definition of validity and using Lemma 3.8 we have:

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

Since $e_0 \in [\![\kappa(M)]\!]_{\rho_0}$ by Lemma 9.3 and $e_0 \in [\![\mathrm{I} \wedge -\!\!* l_0]\!]_{\rho_0}$ by Lemma 9.4 we have in particular:

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

By Lemma 9.2, each of the elements of the set $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ uniquely determines the configuration $\langle L_i, n_1, n_2\rangle$, whence our construction of $\rho_0(b)$ yields $\langle L_i, n_1, n_2\rangle \Downarrow_M$. □

9.4. The general case of stack-and-heap-with-permission models

Having established as Theorem 9.5 that validity of a sequent of the form $\mathcal{F}_{M,C}$ in the basic RAM-domain model implies termination of machine M from configuration C, we now extend this result to the most sophisticated *stack-and-heap-with-permission* models from Section 2.5. As per Remark 2.1, all the models from Sections 2.1-2.5 can be seen as special instances of such models by taking appropriate locations L, record values RV, stacks S and permission algebra $(P, \bullet, \mathbb{1})$. Without loss of generality, we consider the infinite set of locations L to be given by $\mathbb{N}$. We write $\circ'$ for the composition in a stack-and-heap-with-permission model and $\cdot'$ for its extension to sets to disambiguate from the corresponding operations $\circ$ and $\cdot$ of the RAM-domain model.

Definition 9.6. Let $(S \times H, \circ', E)$ be a stack-and-heap model from Section 2.5, where S is a set of stacks and H is a set of heaps-with-permissions from locations $L = \mathbb{N}$ to pairs of record values and permissions in $RV \times P$, where $(P, \bullet, \mathbb{1})$ is a permission algebra. (Recall that $\mathbb{1} \bullet \pi$ is undefined for all $\pi \in P$.) We abuse notation slightly by writing the unit e_0 of the RAM-domain model for the heap with empty domain. Thus $E = \{\langle s, e_0\rangle \mid s \in S\}$.

Based on our valuation ρ_0 for the RAM-domain model and a given machine M in Definition 9.1, 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:

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

where $h(\ell) = \langle \_, \mathbb{1}\rangle$ means that $h(\ell) = \langle s, \mathbb{1}\rangle$ for some $s \in S$. Then for any atomic proposition p we define its valuation by:

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

LEMMA 9.7. *For any stack-and-heap model $(S \times H, \circ', E)$ and any atomic propositions p and q, we have the following identity:*

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

PROOF. First, we show that for any $d_1 \in \rho_0(p)$, $d_2 \in \rho_0(q)$,

$$[d_1 \circ d_2] = [d_1] \cdot' [d_2] \tag{26}$$

For disjoint d_1 and d_2 this is given by construction, since

$$\begin{aligned}
[d_1] \cdot' [d_2] &= \{\langle s_0, h_1\rangle \circ' \langle s_0, h_2\rangle \mid \langle s_0, h_1\rangle \in [d_1], \langle s_0, h_2\rangle \in [d_2]\} \\
&= \{\langle s_0, h_1\rangle \circ' \langle s_0, h_2\rangle \mid \mathtt{domain}(h_1) = [d_1], \mathtt{domain}(h_2) = [d_2], \\
&\qquad\qquad \forall \ell \in d_1.\ h_1(\ell) = \langle _, \mathbb{1}\rangle, \forall \ell \in d_2.\ h_2(\ell) = \langle _, \mathbb{1}\rangle\} \\
&= \{\langle s_0, h\rangle \mid \mathtt{domain}(h) = d_1 \circ d_2, \forall \ell \in d_1 \circ d_2.\ h(\ell) = \langle _, \mathbb{1}\rangle\} \\
&= [d_1 \circ d_2]
\end{aligned}$$

For overlapping d_1 and d_2, we have $d_1 \circ d_2$ undefined and thus $[d_1 \circ d_2] = \emptyset$. We show that $[d_1] \cdot' [d_2] = \emptyset$. Assume for contradiction that $\langle s_0, h_1\rangle \circ' \langle s_0, h_2\rangle$ 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 $[d_2]$, and since d_1, d_2 are overlapping, this implies $h_1(\ell) = h_2(\ell) = \langle \_, \mathbb{1}\rangle$ for some $\ell \in d_1 \cap d_2$. 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 (because in heap-with-permissions models (cf. Section 2.4) we add the permissions when composing compatible heaps), which is a contradiction. Thus $[d_1] \cdot' [d_2]$ is empty when $d_1 \circ d_2$ is undefined.
Now, using Lemma 3.8, we have as required:

$$\begin{aligned}
[\![p * q]\!]_{\rho_1} &= [\![p]\!]_{\rho_1} \cdot' [\![q]\!]_{\rho_1} \\
&= \{\langle s_0, h_1\rangle \circ' \langle s_0, h_2\rangle \mid \langle s_0, h_1\rangle \in \rho_1(p), \langle s_0, h_2\rangle \in \rho_1(q)\} \\
&= \{\langle s_0, h_1\rangle \circ' \langle s_0, h_2\rangle \mid \langle s_0, h_1\rangle \in \textstyle\bigcup_{d \in \rho_0(p)} [d], \langle s_0, h_2\rangle \in \bigcup_{d \in \rho_0(q)} [d]\} \\
&= \textstyle\bigcup_{d_1 \in \rho_0(p),\ d_2 \in \rho_0(q)} \{\langle s_0, h_1\rangle \circ' \langle s_0, h_2\rangle \mid \langle s_0, h_1\rangle \in [d_1], \langle s_0, h_2\rangle \in [d_2]\} \\
&= \textstyle\bigcup_{d_1 \in \rho_0(p),\ d_2 \in \rho_0(q)} [d_1] \cdot' [d_2] \\
&= \textstyle\bigcup_{d_1 \in \rho_0(p),\ d_2 \in \rho_0(q)} [d_1 \circ d_2] \qquad \text{(by (26))} \\
&= \textstyle\bigcup_{d \in \{d_1 \circ d_2 \mid d_1 \in \rho_0(p),\ d_2 \in \rho_0(q)\}} [d] \\
&= \textstyle\bigcup_{d \in [\![p]\!]_{\rho_0} \cdot [\![q]\!]_{\rho_0}} [d] \\
&= \textstyle\bigcup_{d \in [\![p*q]\!]_{\rho_0}} [d]
\end{aligned}$$

□

LEMMA 9.8. *For any stack-and-heap model $(S \times H, \circ', E)$ and any formula A of the form l_i, $l_i * p_k$, or $l_i \vee l_j$, we have the equivalence*

$$[\![A]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1} \ \Leftrightarrow\ \langle s_0, h\rangle \in [d] \text{ and } [\![A]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$$

where $d = \mathtt{domain}(h)$.

PROOF. First, note that by distinguishing cases on A, and using Lemma 9.7 in the case $A = l_i * p_k$, we easily have that

$$[\![A]\!]_{\rho_1} = \bigcup_{d \in [\![A]\!]_{\rho_0}} [d] \tag{27}$$

We show each of the required implications separately.

($\Rightarrow$) Suppose that $[\![A]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1}$. That is, we have:

$$\{\langle s', h'\rangle \circ' \langle s_0, h\rangle \mid \langle s', h'\rangle \models_{\rho_1} A\} \subseteq \bigcup_{d \in \rho_0(b)} [d] \tag{28}$$

Now let $d = \mathtt{domain}(h)$, and let $x \in [\![A]\!]_{\rho_0} \cdot \{d\}$. Thus $x = d \circ d'$, where d and d' are disjoint and $d' \models_{\rho_0} A$. We require to show that $d \circ d' \in [\![b]\!]_{\rho_0}$ and $\langle s_0, h\rangle \in [d]$.

Let $\langle s_0, h'\rangle \in [d']$. Since $d' \in [\![A]\!]_{\rho_0}$, we have by (27) $\langle s_0, h'\rangle \in [\![A]\!]_{\rho_1}$, i.e. $\langle s_0, h'\rangle \models_{\rho_1} A$. Since d and d' are disjoint, $\langle s_0, h\rangle \circ' \langle s_0, h'\rangle$ is defined and, by (28),

$$\langle s_0, h\rangle \circ' \langle s_0, h'\rangle \in \bigcup_{d \in \rho_0(b)} [d]$$

That is, $\langle s_0, h\rangle \circ' \langle s_0, h'\rangle \in [d'']$ for some $d'' \in \rho_0(b)$. Since the domain of $\langle s_0, h\rangle \circ' \langle s_0, h'\rangle$ is $d \circ d'$ by construction, we must have $d'' = d \circ d'$. Thus $d \circ d' \in \rho_0(b) = [\![b]\!]_{\rho_0}$ as required. Furthermore, $\langle s_0, h\rangle \in [d]$ because d and d' are disjoint, and thus h and h' do not overlap, but $\langle s_0, h\rangle \circ' \langle s_0, h'\rangle \in [d \circ d']$, which means that $h(\ell) = \langle \_, \mathbb{1}\rangle$ for all $\ell \in d$. This completes the "only if" direction.

($\Leftarrow$) Let $d = \mathtt{domain}(h)$, let $\langle s_0, h\rangle \in [d]$ and suppose $[\![A]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$. The latter inclusion means that:

$$\{d' \circ d \mid d' \models_{\rho_0} A\} \subseteq \rho_0(b) \tag{29}$$

Now suppose that $x \in [\![A]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\}$. Thus $x = \langle s', h'\rangle \circ' \langle s_0, h\rangle$ (which implies $s' = s_0$), where $\langle s', h'\rangle \models_{\rho_1} A$. We require to show that $\langle s_0, h'\rangle \circ' \langle s_0, h\rangle \in [\![b]\!]_{\rho_1}$.

Since $\langle s_0, h'\rangle \in [\![A]\!]_{\rho_1}$ by the above, we have by (27) some d' such that $d' \in [\![A]\!]_{\rho_0}$, i.e. $d' \models_{\rho_0} A$, and $\langle s_0, h'\rangle \in [d']$ (which implies $\mathtt{domain}(h') = d'$). Because $\langle s_0, h\rangle \in [d]$ and $\langle s', h'\rangle \in [d']$, we have $h(\ell) = \langle \_, \mathbb{1}\rangle$ for all $\ell \in d$ and $h'(\ell') = \langle \_, \mathbb{1}\rangle$ for all $\ell' \in d'$. As $\langle s_0, h'\rangle \circ' \langle s_0, h\rangle$ is defined by assumption, it follows that h and h' cannot overlap, and so d and d' are disjoint. Thus, by (29), $d' \circ d \in \rho_0(b)$. We easily have $\langle s_0, h'\rangle \circ' \langle s_0, h\rangle \in [d' \circ d]$, and so

$$\langle s_0, h'\rangle \circ' \langle s_0, h\rangle \in \bigcup_{d \in \rho_0(b)} [d] = \rho_1(b) = [\![b]\!]_{\rho_1}$$

as required. This completes the "if" direction, and the proof. □

LEMMA 9.9. *$\langle s_0, e_0\rangle \models_{\rho_1} \kappa(M)$ for any machine M and any stack-and-heap model $(S \times H, \circ', E)$.*

PROOF. As in Lemma 9.3, we show that $\langle s_0, e_0\rangle \models_{\rho_1} \kappa(\gamma)$ for any instruction γ in Definition 7.1. For any such γ, the formula $\kappa(\gamma)$ is of the form $(\mathbin{-\!\!*} A \mathbin{-\!\!*} \mathbin{-\!\!*} B)$ where A and B are of the form l_i, $l_i \vee l_j$ or $l_i * p_k$. Since $\langle s_0, e_0\rangle$ is a unit of the model and $\langle s_0, e_0\rangle \circ' \langle s, h\rangle$ is undefined for $s \neq s_0$, we have

$$\begin{aligned}
&\langle s_0, e_0\rangle \models_{\rho_1} \mathbin{-\!\!*} A \mathbin{-\!\!*} \mathbin{-\!\!*} B \\
\Leftrightarrow\ &\forall \langle s, h\rangle.\ \langle s_0, e_0\rangle \circ' \langle s, h\rangle \text{ defined and } \langle s, h\rangle \models_{\rho_1} \mathbin{-\!\!*} A \text{ implies } \langle s_0, e_0\rangle \circ' \langle s, h\rangle \models_{\rho_1} \mathbin{-\!\!*} B \\
\Leftrightarrow\ &\forall \langle s', h'\rangle.\ s = s_0 \text{ and } \langle s, h\rangle \models_{\rho_1} \mathbin{-\!\!*} A \text{ implies } \langle s, h\rangle \models \mathbin{-\!\!*} B \\
\Leftrightarrow\ &\forall h.\ \langle s_0, h\rangle \models_{\rho_1} \mathbin{-\!\!*} A \text{ implies } \langle s_0, h\rangle \models_{\rho_1} \mathbin{-\!\!*} B
\end{aligned}$$

Recalling that $\mathbin{-\!\!*} A = A \mathbin{-\!\!*} b$, and using Lemma 3.8, this amounts to showing, for any $h \in H$:

$$[\![A]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1} \Rightarrow [\![B]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1} \tag{30}$$

Assume $[\![A]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1}$, and let $d = \mathtt{domain}(h)$. By Lemma 9.8 we have $\langle s_0, h\rangle \in [d]$, and $[\![A]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$. By Lemma 9.3 we have $e_0 \models_{\rho_0} \kappa(\gamma)$ and thus (by the same reasoning as above) $[\![\mathbin{-\!\!*} A]\!]_{\rho_0} \subseteq [\![\mathbin{-\!\!*} B]\!]_{\rho_0}$, which means that $[\![A]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$ implies $[\![B]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$. Thus we have $[\![B]\!]_{\rho_0} \cdot \{d\} \subseteq [\![b]\!]_{\rho_0}$, whence Lemma 9.8 yields $[\![B]\!]_{\rho_1} \cdot' \{\langle s_0, h\rangle\} \subseteq [\![b]\!]_{\rho_1}$ as required. □

LEMMA 9.10. *$\langle s_0, e_0\rangle \models_{\rho_1} \mathrm{I} \wedge -\!\!* l_0$ for any machine M and any stack-and-heap model $(S \times H, \circ', E)$.*

PROOF. We have $\langle s_0, e_0\rangle \models_{\rho_1} \mathrm{I}$ because $\langle s_0, e_0\rangle \in E$. By Lemma 9.4 we have $e_0 \models_{\rho_0} -\!\!* l_0$ and thus $\rho_0(l_0) \subseteq \rho_0(b)$. Thus we have

$$\rho_1(l_0) = \bigcup_{d \in \rho_0(l_0)} [d] \;\subseteq\; \bigcup_{d \in \rho_0(b)} [d] = \rho_1(b)$$

which implies that $\langle s_0, e_0\rangle \models_{\rho_1} -\!\!* l_0$, as required. □

THEOREM 9.11. *For any machine M and configuration $\langle L_i, n_1, n_2\rangle$, if a sequent $\mathcal{F}_{M,\langle L_i,n_1,n_2\rangle}$ 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 Section 2 then $\langle L_i, n_1, n_2\rangle \Downarrow_M$.

PROOF. The case of Petri net marking models (Section 2.6) and integer partition models (Section 2.7) with their *total* $\circ$ can be covered by the original valuation ρ_0 from Definition 9.1.

For other models, by taking appropriate locations L, record values RV, set of stacks S and permission algebra P (where L is infinite), we may assume that $\mathcal{F}_{M,\langle L_i,n_1,n_2\rangle}$ is valid in a stack-and-heap model $(S \times H, \circ', E)$ as given in Definition 9.6. Thus we have by definition of validity, and using Lemma 3.8:

$$[\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -\!\!* l_0)]\!]_{\rho_1} \;\subseteq\; [\![b]\!]_{\rho_1}$$
$$\text{i.e. } [\![\kappa(M)]\!]_{\rho_1} \cdot' [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_1} \cdot' [\![\mathrm{I} \wedge -\!\!* l_0]\!]_{\rho_1} \;\subseteq\; \rho_1(b)$$

Taking into account that $\langle s_0, e_0\rangle \in [\![\kappa(M)]\!]_{\rho_1}$ and $\langle s_0, e_0\rangle \in [\![\mathrm{I} \wedge -\!\!* l_0]\!]_{\rho_1}$ according to Lemmas 9.9 and 9.10 respectively, we get:

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

Using Lemma 9.7 and our definition of $\rho_1(b)$, we obtain:

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

According to Lemma 9.2, each element from the set $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ uniquely determines the configuration $\langle L_i, n_1, n_2\rangle$, so that our construction of $\rho_0(b)$ yields $\langle L_i, n_1, n_2\rangle \Downarrow_M$. □

10. FINITE APPROXIMATIONS IN INFINITE MODELS

At first sight, 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. [Calcagno et al. 2001].

However, the crucial difference between our setting and theirs is that their decidability result is restricted to specific *finite* valuations ρ such that $\rho(p)$ is finite for every atomic proposition p. More precisely, in [Calcagno et al. 2001] each p represents one *cell*, i.e. a heap whose domain is a singleton. Their decidability result is nevertheless 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.

LEMMA 10.1. *Let $(H, \circ, \{e_0\})$ be a heap model from Section 2.2. There is an algorithm that, for any finite valuation ρ, decides whether $e_0 \models_\rho \kappa(M)$ holds or not.*

PROOF. In principle, this result can be deduced from [Calcagno et al. 2001]. The following direct construction illustrates subtleties of the problem, caused by non-totality of the composition $\circ$.

As in Lemmas 9.3 and 9.9, we have to check whether

$$e_0 \models_\rho \kappa(\gamma)$$

for any machine instruction γ taken from Definition 7.1. For any such γ, the formula $\kappa(\gamma)$ is of the form $(\mathbf{-}A \mathrel{-\!\!*} \mathbf{-}B)$ where A and B are each of the form l_i, $(l_i \vee l_j)$ or $(l_i * p_k)$. Note that, as ρ is a finite valuation, $[\![A]\!]_\rho$ and $[\![B]\!]_\rho$ are also finite.

Checking whether $e_0 \models_\rho \kappa(\gamma)$ thus means checking whether the following sentence is true or not (cf. (19)):

$$\text{for all } h \in H,\ [\![A]\!]_\rho \cdot \{h\} \subseteq \rho(b) \ \text{ implies } \ [\![B]\!]_\rho \cdot \{h\} \subseteq \rho(b) \tag{31}$$

By negating this statement, it suffices to check whether or not there is an $h \in H$ satisfying the following sentence:

$$[\![A]\!]_\rho \cdot \{h\} \subseteq \rho(b) \ \text{ and } \ [\![B]\!]_\rho \cdot \{h\} \not\subseteq \rho(b) \tag{32}$$

The tricky point, caused by the fact that $\circ$ is not total, is that the equation (32) may have an infinite number of solutions h. Nevertheless, we can construct, in advance, two *finite* sets $\mathcal{H}_1, \mathcal{H}_2 \subseteq H$ such that if the equation (32) has a solution, then it has a solution $\widetilde{h}$ belonging to the finite set $\mathcal{H}_1 \cup \mathcal{H}_2$.

In other words, the statement (31), in which the range of the quantified h is infinite, is equivalent to the following decidable statement, in which the *finite* range $\mathcal{H}_1 \cup \mathcal{H}_2$ of the quantified h can be computed in advance:

$$\text{for all } h \in \mathcal{H}_1 \cup \mathcal{H}_2,\ [\![A]\!]_\rho \cdot \{h\} \subseteq \rho(b) \ \text{ implies } \ [\![B]\!]_\rho \cdot \{h\} \subseteq \rho(b) \tag{33}$$

The practical upshot of this transformation is that checking whether $e_0 \models_\rho \kappa(\gamma)$ can be done in a finite number of steps. It just remains to show how to construct the finite sets $\mathcal{H}_1$ and $\mathcal{H}_2$.

Let $[\![A]\!]_\rho = \{f_1, f_2, .., f_m\}$ and $[\![B]\!]_\rho = \{g_1, .., g_t\}$. Given a solution h to (32), we consider two cases depending on whether or not $[\![A]\!]_\rho \cdot \{h\} = \emptyset$:

- **Case** $[\![A]\!]_\rho \cdot \{h\} \neq \emptyset$.
  In this case, the product $f_i \circ h$ must be defined for some i, and, by (32), we have $f_i \circ h \in \rho(b)$. Thus h is guaranteed to belong to the *finite* set $\mathcal{H}_1$ of all "sub-heaps" of $\rho(b)$, defined as follows:
  $$\mathcal{H}_1 = \{h \in H \mid \exists h' \in H.\ h' \circ h \in \rho(b)\}$$
- **Case** $[\![A]\!]_\rho \cdot \{h\} = \emptyset$**.**
  In this case, for each i such that $1 \leq i \leq m$, the product $f_i \circ h$ is undefined and hence there is an ℓ_i such that $\ell_i \in \mathtt{domain}(f_i) \cap \mathtt{domain}(h)$. Notice that, by (32), we have $[\![B]\!]_\rho \cdot \{h\} \not\subseteq \rho(b)$. Consequently, for some $1 \leq j \leq t$, we have
  $$\mathtt{domain}(g_j) \cap \{\, \ell_1, \ell_2, .., \ell_m \,\} = \emptyset \tag{34}$$
  (Otherwise, for all j we have that $\mathtt{domain}(g_j) \cap \{\, \ell_1, \ell_2, .., \ell_m \,\} \neq \emptyset$, and thereby $\mathtt{domain}(g_j) \cap \mathtt{domain}(h) \neq \emptyset$, resulting in $[\![B]\!]_\rho \cdot \{h\} = \emptyset$, which contradicts (32).)
  Now we construct a new solution $\widetilde{h}$ in the following way. Let $\widetilde{a}$ be a fresh element of L such that $\widetilde{a}$ does not occur in $[\![A]\!]_\rho$, $[\![B]\!]_\rho$ or $\rho(b)$, and let $\widetilde{b}$ be a fixed element of RV. Then the domain of $\widetilde{h}$ is defined as the extended set $\{\, \ell_1, \ell_2, .., \ell_m, \widetilde{a} \,\}$, and for each x from $\mathtt{domain}(\widetilde{h})$ we set:
  $$\widetilde{h}(x) = \widetilde{b}$$
  It remains to show that $\widetilde{h}$ indeed satisfies the sentence (32).

Firstly, for all $1 \leq i \leq m$, we have $\mathtt{domain}(f_i) \cap \mathtt{domain}(\widetilde{h}) \neq \emptyset$ because $\ell_i \in \mathtt{domain}(\widetilde{h})$ by construction. This implies that $[\![A]\!]_\rho \cdot \{\widetilde{h}\} = \emptyset$.

Secondly, by (34), $\mathtt{domain}(g_j)$ and $\mathtt{domain}(\widetilde{h})$ are disjoint for some $1 \leq j \leq t$, so that $g_j \circ \widetilde{h}$ is defined and, because of the choice of $\widetilde{a}$, we have $g_j \circ \widetilde{h} \notin \rho(b)$, so $[\![B]\!]_\rho \cdot \{\widetilde{h}\} \not\subseteq \rho(b)$. Therefore, we can conclude that $\widetilde{h}$ is still a solution to (32).

Now we define a set $\mathcal{H}_2$ of 'solution candidates' as follows. For each choice of ℓ_1, ℓ_2,..., ℓ_m from $\mathtt{domain}(f_1)$, $\mathtt{domain}(f_2)$,..., $\mathtt{domain}(f_m)$, respectively, $\mathcal{H}_2$ contains the heap h' defined as follows:

$$\mathtt{domain}(h') = \{\, \ell_1, \ell_2, .., \ell_m, \widetilde{a} \,\} \text{ and } \forall x \in \mathtt{domain}(h').\ h'(x) = \widetilde{b}$$

It is clear that $\mathcal{H}_2$ is finite and contains every solution of the form $\widetilde{h}$ above, as required. This completes the proof. □

THEOREM 10.2. *Let $(H, \circ, \{e_0\})$ be a heap model from Section 2.2. Then there is an algorithm that, for any finite valuation ρ, and any sequent $\mathcal{F}_{M,\langle L_i, n_1, n_2\rangle}$ 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. Given an $\mathcal{F}_{M,\langle L_i, n_1, n_2\rangle}$ we first use Lemma 10.1 to compute $[\![\kappa(M)]\!]_\rho$ and $[\![\mathrm{I} \wedge -\!\!* l_0]\!]_\rho$ (note that each of these is either empty or the singleton set $\{e_0\}$). If either of these sets is empty then trivially $\mathcal{F}_{M,\langle L_i, n_1, n_2\rangle}$ is valid under the valuation ρ.

Otherwise, $[\![\kappa(M)]\!]_\rho = [\![\,\mathrm{I} \wedge -\!\!* l_0\,]\!]_\rho = \{e_0\}$, so deciding the validity of $\mathcal{F}_{M,\langle L_i, n_1, n_2\rangle}$ under ρ means deciding the inclusion

$$\begin{aligned}
& [\![\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -\!\!* l_0)]\!]_\rho \ \subseteq\ \rho(b) \\
\text{i.e. } & [\![\kappa(M)]\!]_\rho \cdot [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \cdot [\![(\mathrm{I} \wedge -\!\!* l_0)]\!]_\rho \ \subseteq\ \rho(b) \\
\text{i.e. } & \{e_0\} \cdot [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \cdot \{e_0\} \ \subseteq\ \rho(b) \\
\text{i.e. } & [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho \ \subseteq\ \rho(b)
\end{aligned}$$

which is straightforward since both $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_\rho$ and $\rho(b)$ are finite. □

10.1. Non-compactness

COROLLARY 10.3. *We can construct a sequent $\mathcal{F}_{M,\langle L_1, n_0, 0\rangle}$ of the form*

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

so that, for each heap model $(H, \circ, \{e_0\})$ from Section 2.2,

(a) *$\mathcal{F}_{M,\langle L_1, n_0, 0\rangle}$ is not valid in this model (and hence the corresponding computation of M from $\langle L_1, n_0, 0\rangle$ does not terminate),*

(b) *but $\mathcal{F}_{M,\langle L_1, n_0, 0\rangle}$ is valid in this model under all finite valuations ρ.* □

PROOF. Take M such that $K_M =_{\text{def}} \{n \mid \langle L_1, n, 0\rangle \Downarrow_M\}$ is undecidable (the existence of such machines is guaranteed by [Minsky 1967]). K_M is recursively enumerable.

Now let $(H, \circ, \{e_0\})$ be a heap model from Section 2.2. According to Figure 1 in Section 5, we have

$$\begin{aligned}
K_M &= \{n \mid \mathcal{F}_{M,\langle L_1, n, 0\rangle} \text{ provable in Minimal BBI}\} \\
&= \{n \mid \mathcal{F}_{M,\langle L_1, n, 0\rangle} \text{ valid in all separation models}\} \\
&= \{n \mid \mathcal{F}_{M,\langle L_1, n, 0\rangle} \text{ valid in } (H, \circ, \{e_0\})\}
\end{aligned}$$

Next, we define

$$W_M =_{\text{def}} \{n \mid \mathcal{F}_{M,\langle L_1, n, 0\rangle} \text{ is not valid in } (H, \circ, \{e_0\}) \text{ under some finite valuation } \rho\}$$

By Theorem 10.2, W_M is also recursively enumerable.

However, by construction, 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 $n_0 \notin K_M \cup W_M$. We can now complete the two parts of the corollary as follows:

(a) Since $n_0 \notin K_M$, Theorem 9.11 implies that $\mathcal{F}_{M,\langle L_1,n_0,0\rangle}$ is not valid in $(H, \circ, \{e_0\})$.

(b) However, $n_0 \notin W_M$ implies that $\mathcal{F}_{M,\langle L_1,n_0,0\rangle}$ is valid in $(H, \circ, \{e_0\})$ under all finite valuations ρ. □

11. 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 the bunched logic Classical BI.

Definition 11.1. 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 $h \circ h^{-1} = e^{-1}$ for all $h \in H$.

The CBI-models we consider here form a subclass of the more general *relational* CBI-models given in [Brotherston and Calcagno 2010].

Example 11.2. Examples of CBI-models (cf. [Brotherston and Calcagno 2010]):

(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 ε 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 [Foulis and Bennett 1994], which arise in the foundations of quantum mechanics, are exactly CBI-models with indivisible units.

(d). Permission algebras $(P, \bullet, \mathbb{1})$ [Bornat et al. 2005] enriched with a 'formal unit' e and 'formal equalities' $e \bullet h = h \bullet e = e$ can be shown to be exactly non-degenerate (i.e. having more than one element) CBI-models with indivisible units.

Definition 11.3. Following Definition 4.1, we introduce a second chain of logics as follows:

$$\text{BBI} \subseteq \text{CBI} \subseteq \text{CBI+eW} \subseteq \text{CBI+W}$$

- *Classical* BI, a.k.a. CBI [Brotherston and Calcagno 2010] is obtained from BBI by extending its language with a constant $\widetilde{\text{I}}$, and adding the axiom $\sim\sim A \vdash A$, where $\sim A$ is an abbreviation for $(A \mathrel{-\!\!*} \widetilde{\text{I}})$.

- CBI+eW is obtained by extending CBI with the restricted $*$-weakening $\text{I} \wedge (A * B) \vdash A$.

- 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 given by extending the satisfaction relation in Definition 3.9 with the clause:

$$h \models_\rho \widetilde{\text{I}} \;\Leftrightarrow\; h \neq e^{-1}$$

Proposition 11.4. *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.*

PROOF. Since CBI-models are special cases of separation models, and BBI is sound wrt. separation models (Proposition 4.2), it suffices to show for the first part that the CBI axiom $\sim\sim A \vdash A$ is valid in an arbitrary CBI-model $(H, \circ, e, \cdot^{-1})$.

First, we observe that for any $h \in H$, the element h^{-1} is the *unique* element such that $h \circ h^{-1} = e^{-1}$. For, suppose that $h \circ x = e^{-1}$, then we have by associativity and commutativity of $\circ$:

$$e^{-1} \circ x = (h \circ h^{-1}) \circ x = (h \circ x) \circ h^{-1} = e^{-1} \circ h^{-1}$$

whence we obtain $x = h^{-1}$ by cancellativity[2] of $\circ$. As an immediate corollary, we also have $(h^{-1})^{-1} = h$.

Now by the definition of validity above and recalling that $\sim A =_{\text{def}} (A \mathrel{-\!\!*} \widetilde{\mathrm{I}})$, we have:

$$h \models_\rho \sim A \;\Leftrightarrow\; \forall h'.\ h \circ h' \text{ defined and } h' \models_\rho A \text{ implies } h \circ h' \neq e^{-1}$$

Since, by the above, h^{-1} is the unique element of H such that $h \circ h^{-1} = e^{-1}$, we obtain:

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

Thus, using the fact that $(h^{-1})^{-1} = h$, we have:

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

and so $\sim\sim A \vdash A$ (as well as its converse) is valid as required.

For soundness of CBI+eW, the fact that restricted $*$-weakening holds in all CBI-models with indivisible units follows from the fact that it holds in all separation models with indivisible units, as established in Proposition 4.2. □

COROLLARY 11.5. *Interpreting* $\subset$ *as strict inclusion between the set of sequents provable in each system* using only the language of BBI*, we have:*

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

PROOF. The nonstrict versions of the inclusions hold easily by construction. The non-inclusion CBI $\not\subseteq$ BBI was established in [Brotherston and Calcagno 2010]. By the same arguments as in Theorem 3.10 and Corollary 4.4, restricted $*$-weakening is not valid in all CBI-models, and unrestricted $*$-weakening is not valid in all CBI-models with indivisible units. Thus the non-inclusions CBI+W $\not\subseteq$ CBI+eW $\not\subseteq$ CBI hold by Proposition 11.4. □

PROPOSITION 11.6. CBI+W *is ordinary classical logic.*

PROOF. By the same argument as in Proposition 4.7, the presence of unrestricted $*$-weakening forces the equivalences $A * B \equiv A \wedge B$, $A \mathrel{-\!\!*} B \equiv A \to B$ and $\mathrm{I} \equiv \top$.

It just remains to show that $\widetilde{\mathrm{I}} \equiv \bot$. To see this, first note that $\bot \vdash \widetilde{\mathrm{I}}$ is trivially provable by the usual *ex falso quodlibet* of classical logic. For the reverse direction, first note that $(\bot \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \vdash \top$ is trivially provable. As a consequence, $(\top \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \vdash ((\bot \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \mathrel{-\!\!*} \widetilde{\mathrm{I}})$ is also provable. By the CBI axiom, we have $((\bot \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \vdash \bot$, whence by transitivity of $\vdash$ we obtain $(\top \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \vdash \bot$. By the equivalence $\mathrm{I} \equiv \top$, we then have $(\mathrm{I} \mathrel{-\!\!*} \widetilde{\mathrm{I}}) \vdash \bot$. Since $\widetilde{\mathrm{I}} \vdash (\mathrm{I} \mathrel{-\!\!*} \widetilde{\mathrm{I}})$ is easily derivable, we have $\widetilde{\mathrm{I}} \vdash \bot$ provable by transitivity as required.

We observe that, as a consequence, $\sim A \equiv A \to \bot \equiv \neg A$. □

Since (Minimal) BBI-provability implies CBI-provability, to establish undecidability for CBI it suffices (see Figure 1) to prove the analogue of Theorem 9.5 for a CBI-model.

[2]The original definition of a CBI-model in [Brotherston and Calcagno 2010] requires as an axiom that the dual h^{-1} be unique for each h, whereas here we drop this requirement in favour of cancellativity of $\circ$, which is arguably more natural.

Definition 11.7. We introduce the *RAM-codomain 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 the union of disjoint sets (and undefined for non-disjoint sets), the unit e_0 is $\emptyset$ and $\cdot^{-1}$ is set complement.

By extending the valuation ρ_0 in Definition 9.1, 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

$$\rho_C(b) = \rho_0(b) \cup \{d \in \mathcal{D}^+ \mid d \text{ is co-finite}\}$$

LEMMA 11.8. *$e_0 \models_{\rho_C} \kappa(M)$ for any machine M.*

PROOF. As in Lemma 9.3, we must show $e_0 \models_{\rho_C} \kappa(\gamma)$ for any instruction γ. Here we only examine the case of an increment instruction $\gamma = (L_i\colon c_k{+}{+}; \textbf{goto } L_j;)$ for $k = 1$. We note that we have, as in Lemma 9.3,

$$d \models_{\rho_C} -\!\!\!* A \;\Leftrightarrow\; \forall d'.\ d, d' \text{ disjoint and } d' \models_{\rho_C} A \text{ implies } d \circ d' \in \rho_C(b) \tag{35}$$

We have $\kappa(\gamma) = (-\!\!\!*(l_j * p_1) \mathrel{-\!\!\!*} -\!\!\!* l_i)$. To show $e_0 \models_{\rho_C} \kappa(\gamma)$, we must show for any $d \in \mathcal{D}^+$, as in the corresponding case of Lemma 9.3,

$$d \models_{\rho_C} -\!\!\!*(l_j * p_1) \text{ implies } d \models_{\rho_C} -\!\!\!* l_i$$

Assuming that $d \models_{\rho_C} -\!\!\!*(l_j * p_1)$, there are two cases to consider.

First, if d is finite then we can find (as in the corresponding case of Lemma 9.3) $d_j \in \rho_C(l_j)$ and $d_1 \in \rho_C(p_1)$ such that d, d_j and d_1 are disjoint, and thus in particular $d_j \circ d_1 \models_{\rho_C} l_j * p_1$. Using the equivalence (35), we have:

$$\begin{aligned} d \circ d_j \circ d_1 \;&\in\; \rho_C(b) \\ &=\; \rho_0(b) \cup \{d \in \mathcal{D}^+ \mid d \text{ is co-finite}\} \end{aligned}$$

Since d, d_j and d_1 are finite, we must have $d \circ d_j \circ d_1 \in \rho_0(b)$ and thus, by the same argument as in the corresponding case of Lemma 9.3, $d \in [\![p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$ where $\langle L_j, n_1 + 1, n_2\rangle \Downarrow_M$ and thus $\langle L_i, n_1, n_2\rangle \Downarrow_M$ by applying the instruction γ. Consequently, $d \circ d' \in \rho_0(b) \subseteq \rho_C(b)$ for any d' such that d, d' are disjoint and $d' \in \rho_C(l_i)$. Thus by (35) we have $d \models_{\rho_C} -\!\!\!* l_i$ as required.

If, on the other hand, d is co-finite then, for any choice of $d' \in \rho_C(l_i)$ with d, d' disjoint, $d \circ d'$ must also be co-finite, in which case $d \circ d' \in \rho_C(b)$ because $\rho_C(b)$ contains *all* co-finite sets. Thus, again by (35), we have $d \models_{\rho_C} -\!\!\!* l_i$ as required.

The other cases follow from their analogues in Lemma 9.3 in a similar fashion. □

LEMMA 11.9. *$e_0 \models_{\rho_C} \mathrm{I} \wedge -\!\!\!* l_0$ for any machine M.*

PROOF. Similar to Lemma 9.4. □

THEOREM 11.10. *If the sequent $\kappa(M) * l_i * p_1^{n_1} * p_2^{n_2} * (\mathrm{I} \wedge -\!\!\!* l_0) \vdash b$ 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, and using Lemma 3.8, we have:

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

By Lemmas 11.8 and 11.9 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, we have $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_C} = [\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_0}$, so that $\langle L_i, n_1, n_2\rangle$ is uniquely determined according to Lemma 9.2. In particular, $[\![l_i * p_1^{n_1} * p_2^{n_2}]\!]_{\rho_C}$ is finite, so our construction of $\rho_C(b)$ yields $\langle L_i, n_1, n_2\rangle \Downarrow_M$. □

Again, based on Figure 1, we can assert the following:

COROLLARY 11.11. *The following properties of formulas are undecidable, even when restricted to the language* $(\wedge, \rightarrow, \mathrm{I}, *, -\!\!*)$ *of Minimal* BBI[3]*:*

(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 RAM-codomain model $(\mathcal{D}^+, \circ, e_0, \cdot^{-1})$*.*

PROOF. Similar to Corollary 5.1. □

COROLLARY 11.12. *Neither* CBI *nor* CBI+eW *has the finite model property.*

PROOF. Similar to Corollary 5.2. □

12. CONCLUDING REMARKS

Our main contribution in this paper is that heap-like memory models, of the type considered in Section 2, have a very complicated logical structure even at the propositional level:

(A) Separation logic, which provides us with an effective language for reasoning about such memory models, is undecidable even at the *purely propositional* level.

(B) Moreover, we have established that for any *particular* heap-like memory model drawn from the literature, validity in that model is *undecidable* even for purely propositional formulas in this language. 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 Section 2 (with L infinite and RV, S, P possibly degenerate), we *always* get an undecidable model.

(C) We also have established the undecidability of validity in various classes of separation models, and of provability in BBI and CBI and their siblings.

(D) In fact, to obtain a new exhibit for the 'Undecidability Zoo' - that is, a minimal version of BBI that is "as simple as possible" - we only need to add classical conjunction and implication to the multiplicatives, without invoking either classical negation or falsum (see Section 6).

(E) From a technical point of view, we kill *all* our 'undecidable birds' with one 'stone': a single direct encoding of Minsky machines. This single encoding suffices to cover all cases of interest (see Figure 1).

□

One of the above problems, namely whether or not the specific system BBI is decidable, was widely considered to be open for some time (see e.g. [Galmiche and Larchey-Wendling 2006]). As we see, undecidability of BBI is an immediate corollary of our general results. However, immediately prior to publication of our LICS paper, we discovered that undecidability of BBI can in fact be deduced from the undecidability of equational theories over certain algebras stated in [Kurucz et al. 1995]. An alternative proof of BBI undecidability has also been given independently in [Larchey-Wendling and Galmiche 2010].

[3]On the surface, it seems that we are within the framework of Corollary 5.1, since formulas are restricted to the language $(\wedge, \rightarrow, \mathrm{I}, *, -\!\!*)$, which does not include the new CBI constant $\widetilde{\mathrm{I}}$. However, the problem remains non-trivial because of the fact that CBI is *not* a conservative extension of BBI.

However, our paper overlaps with [Kurucz et al. 1995] and [Larchey-Wendling and Galmiche 2010] *only* with respect to the isolated result of undecidability of BBI (which is obtained by very different techniques in all cases). Here, 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 *concrete* heap-like memory model of practical interest. There is no *ad hoc* connection between these problems and BBI.

12.1. New principles for separation logic

In proving our main undecidability results on separation logic, we have found a number of new principles which should be taken into account at the level of the multiplicative unit I.

(i) Though $*$*-contraction*, $A \vdash A * A$, is not generally valid in separation logic, in Lemma 6.3 we deduce the following *restricted* $*$*-contraction*

$$\mathrm{I} \wedge A \vdash A * A \tag{36}$$

(ii) As for the systems BBI+eW and CBI+eW newly established in this paper, we remark that the memory models of theoretical/practical importance employed in the literature (cf. Section 2) all have indivisible units in the sense of Definition 3.4, which is exactly axiomatized by the *restricted* $*$*-weakening* principle of BBI+eW and CBI+eW (see Theorem 3.10):

$$\mathrm{I} \wedge (A * B) \vdash A \tag{37}$$

This principle is in accordance with the following law on 'conservation of matter':

"*The empty memory cannot be split into non-empty pieces*".

□

12.2. Separation logic vs. linear logic

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 the two. 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 6.3, but this too fails in linear logic.

Finally, while adding the unrestricted $*$-weakening $(A * B) \vdash A$ to linear logic gives us a rich structure of *affine logic*, adding it to BBI forces a collapse into classical logic (Proposition 4.7).

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 [O'Hearn and Pym 1999]. 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. □

12.3. Undecidability vs. decidable fragments of memory models

Finally, our undecidability results for *concrete* heap-like models give new insights into the nature of decidable fragments of memory models such as those given in [Berdine et al. 2004; Calcagno et al. 2001], 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 arbitrary infinite valuations (as in [Calcagno et al. 2001]) or restrict the formula language (as in [Berdine et al. 2004]). □

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 [Foulis and Bennett 1994], and to Tadeusz Litak who has attracted our attention to [Kurucz et al. 1995]. We owe special thanks to the referees for their scrupulous and insightful comments, which have allowed us to significantly improve the exposition.

REFERENCES

Ahmed, A., Jia, L., and Walker, D. 2003. Reasoning about hierarchical storage. In *Proceedings of LICS-18*. IEEE Computer Society, 33–44.

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. 1974. *The Design and Analysis of Computer Algorithms*. Addison-Wesley.

Andrews, G. E. 1976. *The theory of partitions*. Encyclopedia of mathematics and its applications. Addison-Wesley.

Benton, N. P., Bierman, G. M., de Paiva, V., and Hyland, M. 1993. A term calculus for intuitionistic linear logic. In *Proceedings of TLCA-1*. Springer, 75–90.

Berdine, J., Calcagno, C., and O'Hearn, P. 2004. A decidable fragment of separation logic. In *Proceedings of FSTTCS-24*. Springer, 97–109.

Bornat, R., Calcagno, C., O'Hearn, P., and Parkinson, M. 2005. Permission accounting in separation logic. In *Proceedings of POPL-32*. 59–70.

Brotherston, J. 2012. Bunched logics displayed. *Studia Logica: Special Issue on Recent Developments related to Residuated Lattices and Substructural Logics 100,* 6, 1223–1254.

Brotherston, J. and Calcagno, C. 2009. Classical BI (A logic for reasoning about dualising resources). In *Proceedings of POPL-36*. ACM, 328–339.

Brotherston, J. and Calcagno, C. 2010. Classical BI: Its semantics and proof theory. *Logical Methods in Computer Science 6,* 3.

Brotherston, J. and Kanovich, M. 2010. Undecidability of propositional separation logic and its neighbours. In *Proceedings of LICS-25*. IEEE Computer Society, 137–146.

Calcagno, C., Distefano, D., O'Hearn, P., and Yang, H. 2011. Compositional shape analysis by means of bi-abduction. *Journal of the ACM 58,* 6.

Calcagno, C., O'Hearn, P., and Yang, H. 2007. Local action and abstract separation logic. In *Proceedings of LICS-22*. IEEE Computer Society, 366–378.

Calcagno, C., Yang, H., and O'Hearn, P. W. 2001. Computability and complexity results for a spatial assertion language for data structures. In *Proceedings of FSTTCS-21*. Springer, 108–119.

Distefano, D. and Parkinson, M. 2008. jStar: Towards practical verification for Java. In *Proceedings of OOPSLA-23*. ACM, 213–226.

Dodds, M., Feng, X., Parkinson, M., and Vafeiadis, V. 2009. Deny-guarantee reasoning. In *Proceedings of ESOP-18*. Springer, 363–377.

Foulis, D. and Bennett, M. 1994. Effect algebras and unsharp quantum logics. *Foundations of Physics 24*, 1331–1352.

Galmiche, D. and Larchey-Wendling, D. 2006. Expressivity properties of Boolean BI through relational models. In *Proceedings of FSTTCS-26*. Springer, 357–368.

Galmiche, D., Méry, D., and Pym, D. 2005. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science 15*, 1033–1088.

Gardner, P., Maffeis, S., and Smith, G. D. 2012. Towards a program logic for JavaScript. In *Proceedings of POPL-39*. 31–44.

Girard, J.-Y. and Lafont, Y. 1987. Linear logic and lazy computation. In *Proceedings of TAPSOFT'87*. Springer-Verlag, 52–66.

Gotsman, A., Cook, B., Parkinson, M., and Vafeiadis, V. 2009. Proving that non-blocking algorithms don't block. In *Proceedings of POPL-36*. ACM, 16–28.

Ishtiaq, S. and O'Hearn, P. W. 2001. BI as an assertion language for mutable data structures. In *Proceedings of POPL-28*. ACM, 14–26.

Kanovich, M. 1992. Horn programming in linear logic is NP-complete. In *Proceedings of LICS-7*. IEEE Computer Society, 200–210.

Kanovich, M. 1995. The direct simulation of Minsky machines in linear logic. In *Advances in Linear Logic*, London Mathematical Society Lecture Notes Series, vol. 222. Cambridge University Press, 123–145.

Kurucz, A., Nemeti, I., Sain, I., and Simon, A. 1995. Decidable and undecidable modal logics with a binary modality. *Journal of Logic, Language and Information 4*, 191–206.

Larchey-Wendling, D. and Galmiche, D. 2010. The undecidability of Boolean BI through phase semantics. In *Proceedings of LICS-25*. IEEE Computer Society, 140–149.

Minsky, M. 1967. *Computation: finite and infinite machines*. Prentice-Hall, Inc.

Murata, T. 1989. Petri nets: properties, analysis and applications. *Proceedings of the IEEE 77,* 4, 541–580.

O'Hearn, P. W. and Pym, D. J. 1999. The logic of bunched implications. *Bulletin of Symbolic Logic 5,* 2, 215–244.

Parkinson, M. and Bierman, G. 2008. Separation logic, abstraction and inheritance. In *Proceedings of POPL-35*. ACM, 75–86.

Parkinson, M., Bornat, R., and Calcagno, C. 2006. Variables as resource in Hoare logics. In *Proceedings of LICS-21*. IEEE Computer Society, 137–146.

Peterson, J. L. 1981. *Petri Net Theory and the Modeling of Systems*. Prentice Hall.

Pym, D. 2002. *The Semantics and Proof Theory of the Logic of Bunched Implications*. Applied Logic Series. Kluwer. Errata and remarks (Pym 2008) maintained at `http://www.abdn.ac.uk/~csc335/BI-monograph-errata.pdf`.

Pym, D., O'Hearn, P., and Yang, H. 2004. Possible worlds and resources: The semantics of BI. *Theoretical Computer Science 315,* 1, 257–305.

Reynolds, J. C. 2002. Separation logic: A logic for shared mutable data structures. In *Proceedings of LICS-17*. IEEE Computer Society, 55–74.

Statman, R. 1979. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science 9*, 67–72.

Yang, H., Lee, O., Berdine, J., Calcagno, C., Cook, B., Distefano, D., and O'Hearn, P. 2008. Scalable shape analysis for systems code. In *Proceedings of CAV-20*. Springer, 385–398.