Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 336 (2018) 101–118

www.elsevier.com/locate/entcs

A Stone-type Duality Theorem for Separation Logic Via its Underlying Bunched Logics

Simon Docherty[1] and David Pym [2]

Department of Computer Science
University College London
London, United Kingdom

Abstract

Stone-type duality theorems, which relate algebraic and relational/topological models, are important tools in logic because — in addition to elegant abstraction — they strengthen soundness and completeness to a categorical equivalence, yielding a framework through which both algebraic and topological methods can be brought to bear on a logic. We give a systematic treatment of Stone-type duality theorems for the structures that interpret bunched logics, starting with the weakest systems, recovering the familiar Boolean BI, and concluding with Separation Logic. Our results encompass all the known existing algebraic approaches to Separation Logic and prove them sound with respect to the standard store-heap semantics. We additionally recover soundness and completeness theorems of the specific truth-functional models of these logics as presented in the literature. This approach synthesises a variety of techniques from modal, substructural and categorical logic and contextualises the 'resource semantics' interpretation underpinning Separation Logic amongst them. As a consequence, theory from those fields — as well as algebraic and topological methods — can be applied to both Separation Logic and the systems of bunched logics it is built upon. Conversely, the notion of *indexed resource frame* (generalizing the standard model of Separation Logic) and its associated completeness proof can easily be adapted to other non-classical predicate logics.

Keywords: Separation logic, bunched logic, substructural logic, program logic, categorical logic, algebraic logic, representation, Stone duality, complex systems, hyperdoctrine, relational semantics, topological semantics, completeness.

1 Introduction

Bunched logics, beginning with O'Hearn and Pym's **BI** [36], have proved to be exceptionally useful tools in modelling and reasoning about computational and information-theoretic phenomena such as resources, the structure of complex systems, and access control [14,15,23]. Perhaps the most striking example is Separation Logic [38,41] (via BI Pointer Logic [31]), a specific theory of first-order Boolean BI with primitives for mutable data structures. Other examples include layered graph

[1] Email: simon.docherty.14@ucl.ac.uk

[2] Email: d.pym@ucl.ac.uk

https://doi.org/10.1016/j.entcs.2018.03.018

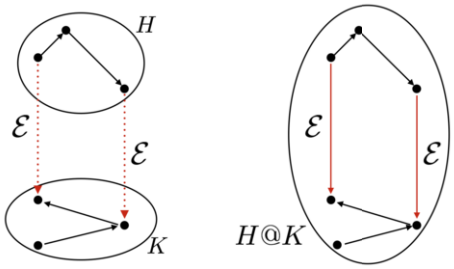

Fig. 1. A layered graph $H \,@_{\mathcal{E}}\, K$

logics [14,15,23], modal and epistemic systems [20,26], and Hennessy–Milner-style process logics that have applications in security [15] and systems modelling [16,2].

The weakest bunched systems are the so-called layered graph logics [14,23]. These logics have a multiplicative conjunction that is neither associative nor commutative, together with its associated implications, and additives that may be classical or intuitionistic. These systems can be used to describe the decomposition of directed graphs into layers (see Fig 1), with applications such as complex systems modelling (e.g., [14,23]) and issues in security concerning the relationship of policies and the systems to which they are intended to apply (e.g., [15,23]). Strengthening the multiplicative conjunction to be associative and commutative yields **BI**, for intuitionistic additives, and Boolean BI (**BBI**), for classical additives. Further extensions include additive and multiplicative modalities and, with the addition of parametrization of modalities on actions, Hennessy–Milner-syle process logics [16,2]. Yet further extensions include additive and multiplicative epistemic modalities [26], with applications in security modelling.

All of the applications of bunched logics to reasoning about computational and information-theoretic phenomena essentially rely on the interpretation of the truth-functional models of these systems known as *resource semantics*. Truth-functional models of bunched logics are, essentially, constructed from pre- or partially ordered partial monoids [29] which, in resource semantics, are interpreted as describing how resource-elements can be combined (monoid composition) and compared (order). The program logic known as *Separation Logic* [31,38,41] is a specific theory of first-order Boolean BI (**FOBBI**) based on the partial monoid of elements of the heap (with the order being simply equality). Separation Logic has found industrial-strength application to static analysis through Facebook's Infer tool (`fbinfer.com`).

Stone's representation theorem for Boolean algebras [39] establishes that every Boolean algebra is isomorphic to a field of sets. Specifically, every Boolean algebra $\mathbb{A}$ is isomorphic to the algebra of clopen subsets of its associated *Stone space* [32] $S(\mathbb{A})$. This result generalizes to a family of Stone-type duality theorems which establish equivalences between certain categories of topological spaces and categories of partially ordered sets. From the logical point of view, Stone-type dualities strengthen the semantic equivalence of truth-functional (such as **BI**'s resource semantics or Kripke's semantics for intuitionistic logic) and algebraic (such as BI algebras or Heyting algebras) models to a dual equivalence of categories. This is useful for

a number of reasons: on the one hand, it provides a theoretically convenient abstract characterization of semantic interpretations and, on the other, it provides a systematic approach to soundness and completeness theorems, via the close relationship between the algebraic structures and Hilbert-type proof systems. Beyond this, Stone-type dualities set up a framework through which techniques from both algebra and topology can be brought to bear on a logic.

In this paper, we give a systematic account of resource semantics via a family of Stone-type duality theorems that encompass the range of systems from the layered graph logics, via Boolean BI, to Separation Logic. Our analysis can also be extended to the intuitionistic variants of each logic, variants with additional multiplicatives [6,7,10] and, we conjecture, the modal and epistemic systems described in [20,26]. As corollaries we retrieve the soundness and completeness of the standard truth-functional models in the literature.

Soundness and completeness theorems for bunched logics and their extensions tend to be proved through labelled tableaux countermodel procedures [29,34,20,26] that must be specified on a logic-by-logic basis, or by lengthy translations into auxilliary modal logics axiomatized by Sahlqvist formulae [12,7,10]. A notable exception to this (and precursor of the completeness result for **BBI** given in the present work) is [27]. We predict our framework will increase the ease with which completeness theorems can be proved, as the family of duality theorems can be extended in a modular fashion. Our results also yield the equivalence of labelled tableaux systems for bunched logics with sequential proof systems that directly present the algebraic semantics [6], as well as provide a foundation for a direct, Sahlqvist-style notion of canonicity for bunched logics, via the canonical extension construction we employ. More generally, the notion of *indexed resource frame* (generalizing the standard model of Separation Logic) and its associated completeness proof can easily be adapted to other non-classical predicate logics.

All of the structures given in existing algebraic approaches to Separation Logic — including [13], [24], [21], [8] and [25] — are instances of the structures used in the present work. Thus these approaches are all proved sound with respect to the standard semantics on store-heap pairs by the results of this paper. In particular, we strengthen the result of [3] interpreting Separation Logic in BI hyperdoctrines. To do so we synthesise a variety of related work from modal [33], relevant [1], substructural [4] and categorical logic [18]. Much of the theory these areas enjoy is produced by way of algebraic and topological arguments. We hope that by recontextualizing the resource semantics of bunched logics in this way similar theory can be given for both Separation Logic and its underlying systems.

In Section 2, we introduce **LGL**, **BBI** and Separation Logic. In Section 3, we define the algebraic, relational and topological structures suitable for interpreting **LGL** and **BBI** and give representation and duality theorems relating them. In Section 4, we strengthen the results of the previous section to Separation Logic by considering **FOBBI**. We recall how **FOBBI** can be interpreted on hyperdoctrines and define new structures called *indexed resource frames*. Crucially, we show that the standard model of Separation Logic is an instantiation of an indexed resource

1. $\phi \vdash \phi$
2. $\phi \vdash \top$
3. $\bot \vdash \phi$
4. $(\phi \to \bot) \to \bot \vdash \phi$
5. $\dfrac{\eta \vdash \phi \quad \eta \vdash \psi}{\eta \vdash \phi \wedge \psi}$
6. $\dfrac{\phi \vdash \psi_1 \wedge \psi_2}{\phi \vdash \psi_i}$
7. $\dfrac{\eta \vdash \psi \quad \phi \vdash \psi}{\eta \vee \phi \vdash \psi}$
8. $\dfrac{\phi \vdash \psi_i}{\phi \vdash \psi_1 \vee \psi_2}$
9. $\dfrac{\eta \wedge \phi \vdash \psi}{\eta \vdash \phi \to \psi}$
10. $\dfrac{\eta \vdash \phi \to \psi \quad \eta \vdash \phi}{\eta \vdash \psi}$
11. $\dfrac{\phi \vdash \psi}{\eta \wedge \phi \vdash \psi}$
12. $\dfrac{\xi \vdash \phi \quad \eta \vdash \psi}{\xi \blacktriangleright \eta \vdash \phi \blacktriangleright \psi}$
13. $\dfrac{\eta \blacktriangleright \phi \vdash \psi}{\eta \vdash \phi \twoheadrightarrow \psi}$
14. $\dfrac{\xi \vdash \phi \twoheadrightarrow \psi \quad \eta \vdash \phi}{\xi \blacktriangleright \eta \vdash \psi}$
15. $\dfrac{\eta \blacktriangleright \phi \vdash \psi}{\phi \vdash \eta \twoheadleftarrow \psi}$
16. $\dfrac{\xi \vdash \phi \twoheadleftarrow \psi \quad \eta \vdash \phi}{\eta \blacktriangleright \xi \vdash \psi}$

Fig. 2. The **LGL** Hilbert system, $\mathrm{LGL_H}$. In 6. and 8. $i = 1, 2$.

frame. We show that the semantics on hyperdoctrines and indexed resource frames are equivalent and strengthen this relationship to a dual equivalence of categories. In Section 5, we consider possibilities for further work as a result of the duality theorems. Proofs of the main results of the paper can be found in an extended research note [22].

2 Preliminaries

2.1 Layered Graph Logic

We begin by presenting the classical logic of layered graphs, **LGL** [14]. The intuitionistic version of **LGL**, **ILGL**, is presented in [23]. We begin with a formal, graph-theoretic definition of layered graph that, we claim, captures the concept as used in modelling complex systems [14,15,23]. Informally, two layers in a directed graph are connected by a specified set of edges, each element of which starts in the upper layer and ends in the lower layer.

Given a directed graph, $\mathcal{G}$, we refer to its *vertex set* and its *edge set* by $V(\mathcal{G})$ and $E(\mathcal{G})$ respectively, while its set of subgraphs is denoted $Sg(\mathcal{G})$, with $H \subseteq \mathcal{G}$ iff $H \in Sg(\mathcal{G})$. For a *distinguished edge set* $\mathcal{E} \subseteq E(\mathcal{G})$, the *reachability relation* $\rightsquigarrow_{\mathcal{E}}$ on $Sg(\mathcal{G})$ is defined $H \rightsquigarrow_{\mathcal{E}} K$ iff a vertex of K can be reached from a vertex of H by an $\mathcal{E}$-edge. This generates a partial composition $@_{\mathcal{E}}$ on subgraphs, with $H @_{\mathcal{E}} K \downarrow$ (where $\downarrow$ denotes definedness) iff $V(H) \cap V(K) = \emptyset, H \rightsquigarrow_{\mathcal{E}} K$ and $K \not\rightsquigarrow_{\mathcal{E}} H$. Output is given by the graph union of the two subgraphs and the $\mathcal{E}$-edges between them. We say G is a *layered graph* (with respect to $\mathcal{E}$) if there exist H, K such that $H @_{\mathcal{E}} K \downarrow$ and $G = H @_{\mathcal{E}} K$ (see Fig 1). Layering is evidently neither commutative nor associative.

Let Prop be a set of atomic propositions, ranged over by p. The set of all

$$G \vDash \mathrm{p} \text{ iff } G \in \mathcal{V}(\mathrm{p}) \qquad G \vDash \top \text{ always} \qquad G \vDash \bot \text{ never}$$
$$G \vDash \phi \wedge \psi \text{ iff } G \vDash \phi \text{ and } G \vDash \psi \qquad G \vDash \phi \vee \psi \text{ iff } G \vDash \phi \text{ or } G \vDash \psi$$
$$G \vDash \phi \to \psi \text{ iff } G \vDash \phi \text{ implies } G \vDash \psi$$
$$G \vDash \phi \blacktriangleright \psi \text{ iff there exists } G_1, G_2 \text{ s.t. } G = G_1 \mathbin{@_{\mathcal{E}}} G_2, G_1 \vDash \phi \text{ and } G_2 \vDash \psi$$
$$G \vDash \phi \mathrel{-\!\!\blacktriangleright} \psi \text{ iff for all } H,\ G \mathbin{@_{\mathcal{E}}} H \downarrow \text{ and } H \vDash \phi \text{ implies } G \mathbin{@_{\mathcal{E}}} H \vDash \psi$$
$$G \vDash \phi \mathrel{\blacktriangleleft\!\!-} \psi \text{ iff for all } H,\ H \mathbin{@_{\mathcal{E}}} G \downarrow \text{ and } H \vDash \phi \text{ implies } H \mathbin{@_{\mathcal{E}}} G \vDash \psi$$

Fig. 3. Satisfaction on layered graphs for **LGL**

formulae of **LGL** is generated by the following grammar:

$$\phi ::= \mathrm{p} \mid \top \mid \bot \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \phi \blacktriangleright \phi \mid \phi \mathrel{-\!\!\blacktriangleright} \phi \mid \phi \mathrel{\blacktriangleleft\!\!-} \phi.$$

The connectives above are the standard (classical additive) logical connectives, together with (non-commutative and non-associative) multiplicative conjunction, $\blacktriangleright$, and its associated implications $\mathrel{-\!\!\blacktriangleright}$ and $\mathrel{\blacktriangleleft\!\!-}$. We define $\neg\phi$ as $\phi \to \bot$. A Hilbert-type system for the logic is given in Fig 2.

LGL is interpreted on layered structures called *scaffolds*. A scaffold is a structure $\mathcal{X} = (\mathcal{G}, \mathcal{E}, X)$ where $\mathcal{G}$ is a directed graph, $\mathcal{E}$ is a distinguished edge set and $X \subseteq Sg(\mathcal{G})$ is such that, if $H \mathbin{@_{\mathcal{E}}} K \downarrow$, $H, K \in X$ iff $H \mathbin{@_{\mathcal{E}}} K \in X$. Given a scaffold $\mathcal{X}$ and a valuation $\mathcal{V} : \mathrm{Prop} \to \mathcal{P}(X)$ (where $\mathcal{P}(X)$ is the power set of X) the satisfaction relation $\vDash$ is inductively defined in Fig 3.

2.2 Boolean BI

Let Prop be a set of atomic propositions, ranged over by p. The set of all formulae of **BBI** is generated by the following grammar:

$$\phi ::= \mathrm{p} \mid \top \mid \bot \mid \mathrm{I} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \phi * \phi \mid \phi \mathrel{-\!\!*} \phi.$$

Once again we have the standard classical additives, this time joined by a multiplicative conjunction $*$ and implication $\mathrel{-\!\!*}$, as well as a constant I. By extending rules 1–11 of Fig 2 with the rules of Fig 4 we obtain a system for **BBI**. These rules enforce commutativity and associativity of the multiplicative conjunction $*$, as well as specifying that I is a unit for $*$.

BBI is interpreted on *partial resource monoids* $\mathrm{R} = (\mathrm{Res}, \circ, e)$, where Res is a set of resources, $\circ : \mathrm{Res} \times \mathrm{Res} \to \mathcal{P}(\mathrm{Res})$ is a *non-deterministic composition* satisfying commutativity and associativity, and e is a unit for $\circ$: for all $r \in \mathrm{Res}$, $r \circ e = \{r\}$. Given a partial resource monoid R and a valuation $\mathcal{V} : \mathrm{Prop} \to \mathcal{P}(\mathrm{Res})$, the satisfaction relation $\vDash$ is inductively defined in Fig 5.

2.3 Separation Logic

Separation Logic [35], introduced by Ishtiaq and O'Hearn [31], and Reynolds [38], is an extension of Hoare's program logic which addresses reasoning about programs that access and mutate data structures. The usual presentation of Separation Logic is based on Hoare triples — for reasoning about the state of imperative programs

$$12'.\ \frac{\xi \vdash \phi \quad \eta \vdash \psi}{\xi * \eta \vdash \phi * \psi} \qquad 13'.\ \frac{\eta * \phi \vdash \psi}{\eta \vdash \phi \mathrel{-\!\!*} \psi}$$

$$14'.\ \frac{\xi \vdash \phi \mathrel{-\!\!*} \psi \quad \eta \vdash \phi}{\xi * \eta \vdash \psi} \qquad 15'.\ \phi * (\psi * \xi) \dashv\vdash (\phi * \psi) * \xi$$

$$16'.\ \phi * \psi \vdash \psi * \phi \qquad 17.\ \phi * \mathrm{I} \dashv\vdash \phi$$

Fig. 4. Rules for the **BBI** Hilbert System, $\mathrm{BBI_H}$

$$r \vDash \mathrm{p} \text{ iff } r \in \mathcal{V}(\mathrm{p}) \qquad r \vDash \top \text{ always} \qquad r \vDash \bot \text{ never}$$
$$r \vDash \phi \wedge \psi \text{ iff } r \vDash \phi \text{ and } r \vDash \psi \qquad r \vDash \phi \vee \psi \text{ iff } r \vDash \phi \text{ or } r \vDash \psi$$
$$r \vDash \phi \to \psi \text{ iff } r \vDash \phi \text{ implies } r \vDash \psi$$
$$r \vDash \mathrm{I} \text{ iff } r = e$$
$$r \vDash \phi * \psi \text{ iff there exists } r_1, r_2 \text{ s.t. } r \in r_1 \circ r_2,\ r_1 \vDash \phi \text{ and } r_2 \vDash \psi$$
$$r \vDash \phi \mathrel{-\!\!*} \psi \text{ iff for all } r', r'' \text{ s.t. } r'' \in r \circ r',\ r' \vDash \phi \text{ implies } r'' \vDash \psi$$

Fig. 5. Satisfaction on partial resource monoids for **BBI**

$$s, h \vDash \top \text{ always} \qquad s, h \models \bot \text{ never} \qquad s, h \models E = E' \text{ iff } \{\{E\}\}s = \{\{E'\}\}s$$
$$s, h \models E \mapsto F \text{ iff } \{\{E\}\}s = \mathrm{dom}(h) \text{ and } h(\{\{E\}\}s) = \{\{\ \mathrm{F}\ \}\}\mathrm{s}$$
$$s, h \models \mathtt{emp} \text{ iff } h = [\,] \text{ (the empty heap)}$$
$$s, h \models \phi * \psi \text{ iff there are } h_0, h_1 \text{ s.t. } h_0 \# h_1,\ h_0 \cdot h_1 = h, s, h_0 \models \phi \text{ and } s, h_1 \models \psi$$
$$s, h \models \phi \mathrel{-\!\!*} \psi \text{ iff for all } h',\ h \# h' \text{ and } s, h' \models \phi, \text{ implies } s, h \cdot h' \models \psi$$
$$s, h \models \phi \to \psi \quad \text{iff} \quad s, h \models \phi \text{ implies } s, h \models \psi$$
$$s, h \models \exists v\,.\,\phi \quad \text{iff} \quad \text{for some } a \in \mathrm{Val},\ [s \mid v \mapsto a], h \models \phi$$
The remaining classical connectives are defined in the usual way: $\neg\phi = \phi \to \bot$; $\phi \vee \psi = (\neg\phi) \to \psi$; $\phi \wedge \psi = \neg(\neg\phi \vee \neg\psi)$; and $\forall x\,.\,\phi = \neg\exists x\,.\,\neg\phi$.

Fig. 6. Satisfaction for BI Pointer Logic

— of the form $\{\,\phi\,\}\,C\,\{\,\psi\,\}$, where C is a program command, ϕ is pre-condition for C, and ψ is a post-condition for C. Reynolds' programming language is a simple language of commands with a Lisp-like set-up for creating and accessing cons cells: $C ::= x := E \mid x := E.i \mid E.i := E' \mid x := \mathit{cons}(E_1, E_2) \mid \ldots$. Here the expressions E of the language are built up using booleans, variables, etc., *cons* cells, and atomic expressions. Separation Logic thus facilitates verification procedures for programs that alter the heap.

A key feature of Separation Logic is the local reasoning provided by the so-called Frame Rule,

$$\frac{\{\,\phi\,\}\,C\,\{\,\psi\,\}}{\{\,\phi * \chi\,\}\,C\,\{\,\psi * \chi\,\}},$$

where χ does not include any free variables modified by the program C. Static analysis procedures based on the Frame Rule form the basis of Facebook's Infer tool (`fbinfer.com`) that is deployed in its code production. The decomposition of the analysis that is facilitated by the Frame Rule is critical to the practical deployability of Infer.

Separation Logic can usefully and safely be seen (see [41] for the details) as a

presentation of BI Pointer Logic [31]. The semantics of BI Pointer Logic, a theory of (first-order) **BBI**, is an instance of **BBI**'s resource semantics in which the monoid of resources is constructed from the program's heap. In detail, this model has two components, the store and the heap. The store is a partial function mapping from variables to values $a \in \mathrm{Val}$, such as integers, and the heap is a partial function from natural numbers to values. In logic, the store is often called the valuation, and the heap is a possible world. In programming languages, the store is sometimes called the environment. Within this set-up, the atomic formulae of BI Pointer Logic include equality between expressions, $E = E'$, and, crucially, the points-to predicate, $E \mapsto F$.

We use the following additional notation: $dom(h)$ denotes the domain of definition of a heap h and $dom(s)$ is the domain of a store s; $h\#h'$ denotes that $dom(h) \cap dom(h') = \emptyset$; $h \cdot h'$ denotes the union of functions with disjoint domains, which is undefined if the domains overlap; $[f \mid v \mapsto a]$ is the partial function that is equal to f except that v maps to a; expressions E are built up from variables and constants, and so determine denotations $\{\{E\}\}s \in \mathrm{Val}$. With this basic data, the satisfaction relation for BI Pointer Logic is defined as in Figure 6. The judgement, $s, h \vDash \phi$, says that the assertion ϕ holds for a given store and heap, assuming that the free variables of ϕ are contained in the domain of s.

Note that the semantics of $E \mapsto F$ requires that E be the only active address in the current heap. Descriptions of larger heaps can be built up using $*$: this corresponds to the local reasoning provided by the Frame Rule. For example, $(9 \mapsto 5) * (10 \mapsto 7)$ describes two adjacent cells whose contents are 5 and 7.

3 Representation and Duality for LGL and BBI

By abstracting from the Hilbert systems and the semantics given in Section 2 we can obtain algebraic and relational semantics (respectively) for the logics **LGL** and **BBI**. We begin with algebraic semantics.

Definition 3.1

(i) A *layered algebra* $\mathbb{A}$ is an algebra $\mathbb{A} = (A, \wedge, \vee, \neg, \top, \bot, \blacktriangleright, \mathrel{-\!\!\blacktriangleright}, \blacktriangleleft\!\!-)$ such that $(A, \wedge, \vee, \neg, \top, \bot)$ is a Boolean algebra and $\blacktriangleright, \mathrel{-\!\!\blacktriangleright}$ and $\blacktriangleleft\!\!-$ are binary operations on A satisfying, for all $a, b, c \in A$, $a \blacktriangleright b \leq c$ iff $a \leq b \mathrel{-\!\!\blacktriangleright} c$ iff $b \leq a \blacktriangleleft\!\!- c$.

(ii) A *resource algebra* is a layered algebra $\mathbb{A}$ extended with a constant I such that a) $\blacktriangleright$ is associative and commutative; and b) for all $a \in \mathbb{A}$, $a \blacktriangleright \mathrm{I} = a$.

We note that for resource algebras, commutativity of $\blacktriangleright$ entails $\mathrel{-\!\!\blacktriangleright} = \blacktriangleleft\!\!-$. LayAlg (ResAlg) denotes the category of layered (resource) algebras and homomorphisms between them. □

Given a valuation $\mathcal{V} : \mathrm{Prop} \to \mathbb{A}$ on a layered algebra, we obtain an interpretation $[\![-]\!]$ for **LGL** on $\mathbb{A}$ as follows: $[\![\mathrm{p}]\!] = \mathcal{V}(\mathrm{p})$, $[\![\top]\!] = \top$, $[\![\bot]\!] = \bot$, $[\![\phi \to \psi]\!] = \neg[\![\phi]\!] \vee [\![\psi]\!]$, and $[\![\phi \circ \psi]\!] = [\![\phi]\!] \circ [\![\psi]\!]$ for $\circ \in \{\wedge, \vee, \blacktriangleright, \mathrel{-\!\!\blacktriangleright}, \blacktriangleleft\!\!-\}$. For a valuation on a resource algebra $\mathbb{A}$ we similarly obtain an interpretation $[\![-]\!]$ for **BBI** on $\mathbb{A}$: in this case we

set $[\![\phi * \psi]\!] = [\![\phi]\!] \blacktriangleright [\![\psi]\!]$, $[\![\phi \mathrel{-\!\!*} \psi]\!] = [\![\phi]\!] \blacktriangleleft\!\!- [\![\psi]\!]$ and $[\![\mathrm{I}]\!] = \mathrm{I}$.

An interpretation $[\![-]\!]$ on a layered (resource) algebra *satisfies* ϕ if $[\![\phi]\!] = \top$. ϕ is valid on layered algebras if it is satisfied under all interpretations. By forming Lindenbaum-Tarski algebras from the Hilbert-type systems given in Figures 2 and 4 we obtain soundness and completeness for this semantics.

Theorem 3.2 *For all formulae ϕ of **LGL** (**BBI**), $\phi \vdash \psi$ is provable in* LGL_H (BBI_H) *iff, for all algebraic interpretations $[\![-]\!]$, $[\![\phi]\!] \leq [\![\psi]\!]$.* □

We now move to the relational structures generalizing the semantics of **LGL** and **BBI**.

Definition 3.3

(i) A *layered frame* $\mathcal{X}$ is a pair $\mathcal{X} = (X, R)$, where X is a set and R is a ternary relation on X.

(ii) A *resource frame* $\mathcal{X}$ is a triple $\mathcal{X} = (X, R, E)$, where (X, R) is a layered frame, $E \subseteq X$ and, for all $x, y, z, t \in X$, the following properties are satisfied:
- (Assoc) $\exists t'(Rxyt'$ and $Rt'zt)$ iff $\exists t'(Ryzt'$ and $Rxt't)$;
- (Comm) $Rxyz$ iff $Ryxz$;
- (Unit) $\exists e \in E, Rexx$ and $\forall e \in E, Rexy$ implies $x = y$. □

It is straightforward to see that these definitions generalize the structures defined in Section 2 to interpret **LGL** and **BBI**. Given a scaffold $(\mathcal{G}, X, \mathcal{E})$, we obtain a layered frame $(X, R_\mathcal{E})$ by defining $R_\mathcal{E}HKG$ iff $H \mathbin{@_\mathcal{E}} K \downarrow$ and $H \mathbin{@_\mathcal{E}} K = G$. Similarly, for a partial resource monoid $(Res, \circ, e)$, we obtain a resource frame $(Res, R_\circ, \{e\})$ by defining $R_\circ r_0 r_1 r$ iff $r \in r_0 \circ r_1$. Using these substitutitions one can reconfigure the semantics given in Figures 3 and 5 to give a satisfaction relation $\vDash$ on frames. For **BBI**, we make one additional adjustment to take care of the move from a single unit e to a set of units E: $x \vDash \mathrm{I}$ iff $x \in E$.

Resource frames are the weakest relational structures that can soundly and completely interpret **BBI**, a fact that is formally captured by the duality theorem 3.12. The notion is closely related to two other types of relational structure from the **BBI** literature — *multi-unit separation algebras* [24] and *relational frames* [27] — and coincides with two others, *BBI frames* [9] and *non-deterministic monoids* [28]. Resource frames have multiple units like multi-unit separation algebras, but drop the cancellativity requirement of the partial composition. In contrast, they are distinguished from relational frames *because of* the fact they have multiple units.

These distinctions are crucial for what follows: the representation and duality theorems do not hold when we restrict to frames satisfying either of these properties. This is also witnessed by the fact that **BBI** is not expressive enough to distinguish between cancellative/non-cancellative models and single unit/multi-unit models [9], all of which define the same notion of validity [28].

To obtain categories LayFr and ResFr we define morphisms for frames.

Definition 3.4 (cf. [9]) Given layered frames $\mathcal{X}$ and $\mathcal{X}'$, a *layered p-morphism* $f : \mathcal{X} \to \mathcal{X}'$ is a function $f : X \to X'$ satisfying the following:

(i) $\forall x, y, z$, if $Rxyz$, then $R'f(x)f(y)f(z)$;

(ii) $\forall x', y', z$, if $R'x'y'f(z)$, then $\exists x, y \in X$ s.t. $Rxyz$, $f(x) = x'$ and $f(y) = y'$;

(iii) $\forall x', y, z'$, if $R'x'f(y)z'$, then $\exists x, z \in X$ s.t. $Rxyz$, $f(x) = x'$ and $f(z) = z'$;

(iv) $\forall x, y', z'$, if $R'f(x)y'z'$, then $\exists y, z \in X$ s.t. $Rxyz$, $f(y) = y'$ and $f(z) = z'$.

A *resource p-morphism* $f : \mathcal{X} \to \mathcal{X}'$ between resource frames $\mathcal{X}$ and $\mathcal{X}'$ is a layered p-morphism that additionally satisfies
(v) $\forall x$, $x \in E$ iff $f(x) \in E'$. □

3.1 Representation and Duality

We now give representation and duality theorems for layered and resource algebras. As a corollary, we obtain the equivalence of the relational semantics to the algebraic semantics, as well as its completeness with respect to the Hilbert systems of Section 2. The soundness and completeness of resource semantics can thus be understood as a consequence of this topological duality.

Definition 3.5 Given a layered frame $\mathcal{X}$, the *complex algebra* of $\mathcal{X}$ is given by $Com(\mathcal{X}) = (\mathcal{P}(X), \cap, \cup, \setminus, X, \emptyset, \blacktriangleright_R, \twoheadrightarrow_R, \blacktriangleleft_R)$, where $\blacktriangleright_R, \twoheadrightarrow_R$ and $\blacktriangleleft_R$ are defined as follows:

$$A \blacktriangleright_R B = \{z \mid \text{there exists } x \in A, y \in B \text{ s.t. } Rxyz\}$$

$$A \twoheadrightarrow_R B = \{x \mid \text{for all } y, z \in X, \text{if } Rxyz \text{ and } y \in A, \text{ then } z \in B\}$$

$$A \blacktriangleleft_R B = \{x \mid \text{for all } y, z \in X, \text{if } Ryxz \text{ and } y \in A, \text{ then } z \in B\}.$$

For a resource frame $\mathcal{X}$, the complex algebra $Com(\mathcal{X})$ is given by extending the complex algebra of the underlying layered frame with the set E. □

Lemma 3.6 *The complex algebra $Com(\mathcal{X})$ of a layered (resource) frame $\mathcal{X}$ is a layered (resource) algebra.* □

We can also define a layered (resource) frame from any layered (resource) algebra. We first recall the notion of (ultra)filter. A *filter* on a Boolean algebra $\mathbb{A}$ is a subset $F \subseteq A$ satisfying, for all $x, y \in A$, (i) $x \in F$ and $x \leq y$ implies $y \in F$; (ii) $x, y \in F$ implies $x \wedge y \in F$. It is *proper* if $\bot \notin F$. An *ultrafilter* is a proper filter that additionally satisfies (iii) $x \vee y \in F$ implies $x \in F$ or $y \in F$. An ultrafilter of a layered (resource) algebra $\mathbb{A}$ is an ultrafilter of its underlying Boolean algebra.

Definition 3.7 Given a layered algebra $\mathbb{A}$, the *ultrafilter frame* $Ult(\mathbb{A})$ is defined $Ult(\mathbb{A}) = (Uf(A), R_{Ult(\mathbb{A})})$, where $Uf(A)$ is the set of ultrafilters on $\mathbb{A}$ and $R_{Ult(\mathbb{A})}$ is defined by $R_{Ult(\mathbb{A})}F_0F_1F_2$ iff, for all $x \in F_0$ and $y \in F_1, x \blacktriangleright y \in F_2$. For a resource algebra $\mathbb{A}$, the ultrafilter frame is given by extending $Ult(\mathbb{A})$ by $E_{R_{Ult(\mathbb{A})}} = \{F \in Uf(A) \mid \mathrm{I} \in F\}$. □

Lemma 3.8 *Given a layered (resource) algebra $\mathbb{A}$, the ultrafilter frame $Ult(\mathbb{A})$ is a layered (resource) frame.* □

We now extend the Stone representation theorem for Boolean algebras to take account of the additional residuated structure of layered/resource algebras. For layered algebras this is not a new result exactly: it can be derived as a specific case of an analogous theorem for Boolean gaggles ([4], Theorem 1.4.16) and is related to representation theorems for algebras with operators ([33], [30]). The difference with the latter results is the use of a single relation R for the operator $\blacktriangleright$ *and* its non-operator adjoints $\blacktriangleright\!\!\rightarrow$ and $\blacktriangleleft\!\!\blacktriangleright$. The derived structure required to take care of these adjoints was not investigated in the frameworks of Jonsson-Tarski or Goldblatt. In addition, the application to the semantics of **LGL** and **BBI** is new.

Theorem 3.9 (Representation Theorem for Layered/Resource Algebras) *Every layered (resource) algebra is isomorphic to a subalgebra of a complex algebra. Specifically, the map* $h_{\mathbb{A}} : \mathbb{A} \to Com(Ult(\mathbb{A}))$ *given by* $h_{\mathbb{A}}(a) = \{F \in Uf(A) \mid a \in F\}$ *is an embedding.* □

Now given an interpretation $[\![-]\!]$ on a layered (resource) algebra $\mathbb{A}$ we can give a valuation $\mathcal{V}_{[\![-]\!]}$ on the ultrafilter frame by $\mathcal{V}_{[\![-]\!]}(\mathrm{p}) = h_{\mathbb{A}}([\![\mathrm{p}]\!])$. Similarly, any valuation $\mathcal{V}$ on a layered (resource) frame $\mathcal{X}$ generates an interpretation on its complex algebra. As $h_{\mathbb{A}}$ is a homomorphism and the definition of the operations of the complex algebra matches the clauses for the relational semantics, we obtain the following corollary.

Corollary 3.10

(i) *For all formulae* ϕ *of* ***LGL*** *(****BBI****),* ϕ *is satisfiable/valid on layered (resource) algebras iff* ϕ *is satisfiable/valid on layered (resource) frames.*

(ii) *The relational semantics of* ***LGL*** *(****BBI****) is sound and complete.* □

Similarly to Stone's representation theorem, our results extend to categorical dualities. As with the representation theorem, for layered algebras this is not a new result: it can be obtained as a specific case of the duality theorem for Boolean gaggles ([4], Theorem 9.2.22).

Definition 3.11

(i) A *layered space* is a structure $\mathcal{X} = (X, \mathcal{O}, R)$ such that
  (a) $(X, \mathcal{O})$ is a *Stone space* [32] and (X, R) a layered frame,
  (b) the clopen sets of $(X, \mathcal{O})$, $\mathcal{CL}(X)$, are closed under $\blacktriangleright_R$, $\blacktriangleright\!\!\rightarrow_R$ and $\blacktriangleleft\!\!\blacktriangleright_R$, and
  (c) if $Rxyz$ does not hold, then there exist clopen sets O_0 and O_1 such that $x \in O_0$, $y \in O_1$ and $z \notin O_0 \blacktriangleright_R O_1$.

(ii) A *resource space* is a structure $\mathcal{X} = (X, \mathcal{O}, R, E)$ such that $(X, \mathcal{O}, R)$ is a layered space, (X, R, E) is a resource frame and E is a clopen set. □

A morphism of layered (resource) spaces $f : \mathcal{X} \to \mathcal{X}'$ is thus a *continuous layered (resource) p-morphism*. This yields categories LaySp and ResSp. Given a layered (resource) algebra $\mathbb{A}$, we can equip its ultrafilter frame with the topology generated by the base $\{h_{\mathbb{A}}(a) \mid a \in A\}$. This yields a layered (resource) space and underpins the categorical duality: a proof can be found in an extended research note [22].

Theorem 3.12 (Duality Theorem for Layered/Resource Algebras) *The categories* LayAlg (ResAlg) *and* LaySp (ResSp) *are dually equivalent.* □

4 A Duality Theorem For Separation Logic

We now extend the duality theorem for resource algebras to the algebraic and relational structures suitable for interpreting Separation Logic. First, we must consider first-order BBI (**FOBBI**). A Hilbert-type proof system is obtained by extending that given for **BBI** in Section 2 with the usual rules for quantifiers (see, e.g., [40]). Second, to give the semantics for the quantifiers of **FOBBI**, we must expand our definitions from the propositional case with category-theoretic structure. As these semantic structures support it, we consider a many-sorted first-order logic. We start on the algebraic side with resource hyperdoctrines.

Definition 4.1 (cf. [3]) A *resource hyperdoctrine* is a tuple

$$(\mathbb{P} : \mathrm{C}^{op} \to \mathrm{Poset}, (=_X)_{X \in Ob(\mathrm{C})}, (\exists X_\Gamma, \forall X_\Gamma)_{\Gamma, X \in Ob(\mathrm{C})}) \quad \text{such that,}$$

(i) C is a category with finite products;

(ii) $\mathbb{P} : \mathrm{C}^{op} \to \mathrm{Poset}$ is a functor such that, for each object X in C, $\mathbb{P}(X)$ is a resource algebra, and, for each morphism f in C, $\mathbb{P}(f)$ is a homomorphism;

(iii) For each object X in C and each diagonal morphism $\Delta_X : X \to X \times X$ in C, $=_X \in \mathbb{P}(X \times X)$ is such that, for all $a \in \mathbb{P}(X \times X)$, $\top \leq \mathbb{P}(\Delta_X)(a)$ iff $=_X \leq a$;

(iv) For each pair of objects Γ, X in C and each projection $\pi_{\Gamma,X} : \Gamma \times X \to \Gamma$ in C, $\exists X_\Gamma$ and $\forall X_\Gamma$ are monotone maps $\exists X_\Gamma : \mathbb{P}(\Gamma \times X) \to \mathbb{P}(\Gamma)$ and $\forall X_\Gamma : \mathbb{P}(\Gamma \times X) \to \mathbb{P}(\Gamma)$ such that, for all $a, b \in \mathbb{P}(\Gamma)$, $\exists X_\Gamma(a) \leq b$ iff $a \leq \mathbb{P}(\pi_{\Gamma,X})(b)$ and $\mathbb{P}(\pi_{\Gamma,X})(b) \leq a$ iff $b \leq \forall X_\Gamma(a)$. This assignment of morphisms is additionally natural in Γ: given a morphism $s : \Gamma \to \Gamma'$, the following diagrams commute:

$$\begin{array}{ccc} \mathbb{P}(\Gamma' \times X) & \xrightarrow{\mathbb{P}(s \times id_X)} & \mathbb{P}(\Gamma \times X) \\ {\scriptstyle \exists X_{\Gamma'}}\downarrow & & \downarrow{\scriptstyle \exists X_\Gamma} \\ \mathbb{P}(\Gamma') & \xrightarrow[\mathbb{P}(s)]{} & \mathbb{P}(\Gamma) \end{array} \qquad \begin{array}{ccc} \mathbb{P}(\Gamma' \times X) & \xrightarrow{\mathbb{P}(s \times id_X)} & \mathbb{P}(\Gamma \times X) \\ {\scriptstyle \forall X_{\Gamma'}}\downarrow & & \downarrow{\scriptstyle \forall X_\Gamma} \\ \mathbb{P}(\Gamma') & \xrightarrow[\mathbb{P}(s)]{} & \mathbb{P}(\Gamma) \end{array}$$

□

Resource hyperdoctrines have appeared elsewhere in the literature as *BI hyperdoctrines* where they were used to prove the existence of models of higher-order variants of Separation Logic [3]. The Boolean quantale [21] and formal power series [25] approaches to algebraic Separation Logic are instantiations of this structure.

To specify an interpretation $[\![-]\!]$ of **FOBBI** in a resource hyperdoctrine, $\mathbb{P}$, we assign each type X an object $[\![X]\!]$ of C, and for each context $\Gamma = \{v_1 : X_1, \ldots, v_n : X_n\}$ we have $[\![\Gamma]\!] = [\![X_1]\!] \times \cdots \times [\![X_n]\!]$. Each function symbol $f : X_1 \times \cdots X_n \to X$ is assigned a morphism $[\![f]\!] : [\![X_1]\!] \times \cdots [\![X_n]\!] \to [\![X]\!]$. This allows us to inductively assign to every term of type X in context Γ a morphism $[\![t]\!] : [\![\Gamma]\!] \to [\![X]\!]$ in the

standard way (see [37]). We additionally assign, for each m-ary predicate symbol P of type $X_1, \ldots, X_m$, $[\![P]\!] \in \mathbb{P}([\![X_1]\!] \times \cdots \times [\![X_m]\!])$. Then the structure of the hyperdoctrine allows us to extend $[\![-]\!]$ to **FOBBI** formulae ϕ in context Γ as follows:

$$[\![Pt_1 \ldots t_m]\!] = \mathbb{P}(\langle [\![t_1]\!], \ldots, [\![t_m]\!] \rangle)([\![P]\!]) \quad [\![t =_X t']\!] = \mathbb{P}(\langle [\![t]\!], [\![t']\!] \rangle)(=_{[\![X]\!]})$$
$$[\![C]\!] = C_{\mathbb{P}([\![\Gamma]\!])} \quad [\![\phi \circ \psi]\!] = [\![\phi]\!] \circ_{\mathbb{P}([\![\Gamma]\!])} [\![\psi]\!] \quad [\![Qv : X\phi]\!] = Q[\![X]\!]_{[\![\Gamma]\!]}([\![\phi]\!])$$

where $C \in \{\top, \bot, I\}$, $\circ \in \{\wedge, \vee, \to, *, -\!\!*\}$ and $Q \in \{\exists, \forall\}$). Substitution of terms is given by $[\![\phi(t/x)]\!] = \mathbb{P}([\![t]\!])([\![\phi]\!])$. ϕ is satisfied by an interpretation $[\![-]\!]$ if $[\![\phi]\!] = \top$. ϕ is valid if it is satisfied by all interpretations.

Theorem 4.2 *[37,3]* ***FOBBI*** *is sound and complete on resource hyperdoctrines.*□

On the relational side, we introduce a new structure: *indexed resource frames*. This definition is adapted from the notion of indexed Stone space presented in [18] as a topological dual for Boolean hyperdoctrines. In contrast to the duality presented there, we additionally consider (typed) equality and universal quantification.

Definition 4.3 An *indexed resource frame* is a functor $\mathcal{R} : \mathrm{C} \to \mathrm{ResFr}$ such that

(i) C is a category with finite products;

(ii) For all objects Γ, Γ' and X in C, all morphisms $s : \Gamma \to \Gamma'$ and all product projections $\pi_{\Gamma,X}$, for the following commutative square

$$\begin{array}{ccc} \mathcal{R}(\Gamma \times X) & \xrightarrow{\mathcal{R}(\pi_{\Gamma,X})} & \mathcal{R}(\Gamma) \\ \downarrow{\scriptstyle \mathcal{R}(s \times id_X)} & & \downarrow{\scriptstyle \mathcal{R}(s)} \\ \mathcal{R}(\Gamma' \times X) & \xrightarrow[\mathcal{R}(\pi_{\Gamma',X})]{} & \mathcal{R}(\Gamma') \end{array}$$

the induced map $\mathcal{R}(\Gamma \times X) \to \mathcal{R}(\Gamma) \times_{\mathcal{R}(\Gamma')} \mathcal{R}(\Gamma' \times X)$ is an epimorphism. This is known as the *quasi-pullback* or *epi-pullback* property.

Given an arbitrary indexed resource frame $\mathcal{R} : \mathrm{C} \to \mathrm{ResFr}$ and an object X we denote the resource frame at X by $\mathcal{R}(X) = (\mathcal{R}(X), R_{\mathcal{R}(X)}, E_{\mathcal{R}(X)})$. □

We now give a truth-functional semantics for **FOBBI** on indexed resource frames. An interpretation $[\![-]\!]$ is given in precisely the same way as for resource hyperdoctrines, except for the key difference that each m-ary predicate symbol P of type $X_1, \ldots, X_m$, is assigned $[\![P]\!] \subseteq \mathcal{R}([\![X_1]\!] \times \cdots \times [\![X_m]\!])$.

Then for formulae ϕ of **FOBBI** in context Γ with $x \in \mathcal{R}([\![\Gamma]\!])$ the satisfaction relation $\vDash^{\Gamma}$ is inductively defined in Fig 7. There, $Ran(\mathcal{R}(\Delta_{[\![X]\!]})) = \{y \mid \exists z(\mathcal{R}(\Delta_{[\![X]\!]})(z) = y)\}$. We note that bound variables are renamed to be fresh throughout, in an order determined by quantifier depth.

4.1 The Pointer Model as an Indexed Resource Frame

Although at first sight it doesn't seem so, indexed resource frames and the semantics based upon them are a generalization of the standard store–heap semantics of Separation Logic.

$$x, [\![-]\!] \vDash^{\Gamma} Pt_1 \dots t_m \text{ iff } \mathcal{R}(\langle [\![t_1]\!], \dots, [\![t_m]\!] \rangle)(x) \in [\![P]\!]$$
$$x, [\![-]\!] \vDash^{\Gamma} t =_X t' \text{ iff } \mathcal{R}(\langle [\![t]\!], [\![t']\!] \rangle)(x) \in Ran(\mathcal{R}(\Delta_{[\![X]\!]}))$$
$$x, [\![-]\!] \vDash^{\Gamma} \top \text{ always} \quad x, [\![-]\!] \vDash^{\Gamma} \bot \text{ never}$$
$$x, [\![-]\!] \vDash^{\Gamma} \phi \wedge \psi \text{ iff } x, [\![-]\!] \vDash^{\Gamma} \phi \text{ and } x, [\![-]\!] \vDash^{\Gamma} \psi$$
$$x, [\![-]\!] \vDash^{\Gamma} \phi \vee \psi \text{ iff } x, [\![-]\!] \vDash^{\Gamma} \phi \text{ or } x, [\![-]\!] \vDash^{\Gamma} \psi$$
$$x, [\![-]\!] \vDash^{\Gamma} \phi \to \psi \text{ iff } x, [\![-]\!] \nvDash^{\Gamma} \phi \text{ or } x, [\![-]\!] \vDash^{\Gamma} \psi$$
$$x, [\![-]\!] \vDash^{\Gamma} I \text{ iff } x \in E_{\mathcal{R}([\![\Gamma]\!])}$$
$$x, [\![-]\!] \vDash^{\Gamma} \phi * \psi \text{ iff there exists } y, z \in \mathcal{R}([\![\Gamma]\!]) \text{ such that } R_{\mathcal{R}([\![\Gamma]\!])} yzx \text{ and}$$
$$y, [\![-]\!] \vDash^{\Gamma} \phi \text{ and } z, [\![-]\!] \vDash^{\Gamma} \psi$$
$$x, [\![-]\!] \vDash^{\Gamma} \phi \mathrel{-\!\!*} \psi \text{ iff, for all } y, z \in \mathcal{R}([\![\Gamma]\!]), \text{ if } R_{\mathcal{R}([\![\Gamma]\!])} yxz \text{ and}$$
$$y, [\![-]\!] \vDash^{\Gamma} \phi, \text{ then } z, [\![-]\!] \vDash^{\Gamma} \psi$$
$$x, [\![-]\!] \vDash^{\Gamma} \exists v_{n+1} : X \phi \text{ iff there exists } x' \in \mathcal{R}([\![\Gamma]\!] \times [\![X]\!]) \text{ such that}$$
$$\mathcal{R}(\pi_{[\![\Gamma]\!],[\![X]\!]})(x') = x \text{ and } x', [\![-]\!] \vDash^{\Gamma \cup \{v_{n+1}:X\}} \phi$$
$$x, [\![-]\!] \vDash^{\Gamma} \forall v_{n+1} : X \phi \text{ iff, for all } x' \in \mathcal{R}([\![\Gamma]\!] \times [\![X]\!]), \text{ if } \mathcal{R}(\pi_{[\![\Gamma]\!],[\![X]\!]})(x') = x,$$
$$\text{then } x', [\![-]\!] \vDash^{\Gamma \cup \{v_{n+1}:X\}} \phi$$

Fig. 7. Satisfaction on indexed resource frames for **FOBBI**

Consider the resource frame Heap $= (H, \uplus, \{[\,]\})$, where H is the set of heaps, $[\,]$ is the empty heap and $\uplus$ is defined by $\uplus h_0 h_1 h_2$ iff $h_0 \# h_1$ and $h_0 \cdot h_1 = h_2$. This is the resource frame corresponding to the partial monoid of heaps.

We define an indexed resource frame Store : Set $\to$ ResFr by $\mathrm{Store}(X) = (X \times H, \uplus_X,\ X \times \{[\,]\})$, where $\uplus_X(x_0, h_0)(x_1, h_1)(x_2, h_2)$ iff $x_0 = x_1 = x_2$ and $\uplus h_0 h_1 h_2$, and $\mathrm{Store}(f : X \to Y)(x, h) = (f(x), h)$. It is straightforward to see this defines a functor: for arbitrary X, $\mathrm{Store}(X)$ inherits the resource frame properties from Heap and for arbitrary $f : X \to Y$, $\mathrm{Store}(f)$ is trivially a resource p-morphism as it is identity on the structure that determines the back and forth conditions. The quasi-pullback property is also satisfied so this defines an indexed resource frame.

The interpretation $[\![-]\!]$ on Store that yields the standard model of Separation Logic is as follows. We have one type Val and we set $[\![\mathrm{Val}]\!] = \mathbb{Z}$, with the arithmetic operations $[\![+]\!], [\![-]\!] : [\![\mathrm{Val}]\!]^2 \to [\![\mathrm{Val}]\!]$ defined as one would expect. Term morphisms $[\![t]\!] : [\![\mathrm{Val}]\!]^n \to [\![\mathrm{Val}]\!]$ in context $\Gamma = \{v_1, \dots v_n\}$ are then defined as usual, with each constant n assigned the morphism $[\![n]\!] : [\![\Gamma]\!] \longrightarrow \{*\} \xrightarrow{n} [\![\mathrm{Val}]\!]$. Finally, the points-to predicate $\mapsto$ is assigned

$$[\![\mapsto]\!] = \{((a, a'), h) \mid dom(h) = \{a\} \text{ and } h(a) = a'\} \subseteq \mathrm{Store}([\![\mathrm{Val}]\!]^2).$$

In the indexed resource frame Store : Set $\to$ ResFr with the interpretation just defined, a store is represented as an n-place vector of values over $[\![\mathrm{Val}]\!]$. That is, the store $s = \{(v_1, a_1), \dots, (v_n, a_n)\}$ is given by the element $(a_1, \dots, a_n) \in [\![\mathrm{Val}]\!]^n$. By a simple inductive argument we have the following result:

Theorem 4.4 *For all formulae ϕ of pointer logic, all stores $s = \{(v_1, a_1), \dots, (v_n, a_n)\}$ and all heaps h, $s, h \vDash \phi$ iff $((a_1, \dots, a_n), h), [\![-]\!] \vDash^{\Gamma} \phi$.* □

The notion of indexed resource frame and its associated semantics are therefore a natural generalization of the standard Separation Logic model.

4.2 Equivalence of Semantics and Duality

We now extend the results given for resource algebras to resource hyperdoctrines. To do so we give analogous structures to complex algebras and ultrafilter frames. To specify complex *hyperdoctrines* we first require an auxiliary definition. Given a function $f : X \to Y$, the *dual image* $f_* : \mathcal{P}(X) \to \mathcal{P}(Y)$ is defined $f_*(A) = \{x \mid \text{for all } y : \text{ if } f(y) = x, \text{ then } y \in A\}$.

Definition 4.5 Given an indexed resource frame $\mathcal{R} : \mathrm{C} \to \mathrm{ResFr}$, the *complex hyperdoctrine* of $\mathcal{R}$, $Com(\mathcal{R}(-)) : \mathrm{C}^{op} \to \mathrm{ResAlg}$ is defined by extending Definition 3.5 to morphisms with $Com(\mathcal{R}(f)) = (\mathcal{R}(f))^{-1}$ and setting $Ran(\mathcal{R}(\Delta_X))$ as $=_X$, the direct image $\mathcal{R}(\pi_{\Gamma,X})$ as $\exists X_\Gamma$, and $\mathcal{R}(\pi_{\Gamma,X})_*$ as $\forall X_\Gamma$. □

Lemma 4.6 *Given an indexed resource frame $\mathcal{R} : \mathrm{C} \to \mathrm{ResFr}$, the complex hyperdoctrine $Com(\mathcal{R}(-))$ is a resource hyperdoctrine.* □

Definition 4.7 Given a resource hyperdoctrine $\mathbb{P} : \mathrm{C}^{op} \to \mathrm{Poset}$ the *indexed ultrafilter frame* $Ult(\mathbb{P}(-)) : \mathrm{C} \to \mathrm{ResFr}$ is given by extending Definition 3.7 to morphisms by setting $Ult(\mathbb{P}(f)) = (\mathbb{P}(f))^{-1}$. □

Lemma 4.8 *Given a resource hyperdoctrine $\mathbb{P} : \mathrm{C}^{op} \to \mathrm{Poset}$ the indexed ultrafilter frame $Ult(\mathbb{P}(-))$ is an indexed resource frame.* □

Given an interpretation $[\![-]\!]$ on an indexed resource frame $\mathcal{R}$ we immediately obtain an interpretation on its complex hyperdoctrine, as for each m-ary predicate symbol P of type $X_1, \ldots, X_m$, $[\![P]\!]$ is an element of $Com(\mathcal{R}([\![[\![X_1]\!] \times \cdots \times [\![X_m]\!]]\!])$, as required. Correspondingly, given an interpretation $[\![-]\!]$ on a resource hyperdoctrine $\mathbb{P}$, we automatically obtain an interpretation $\widetilde{[\![-]\!]}$ on its indexed ultrafilter frame. $\widetilde{[\![-]\!]}$ is the same as $[\![-]\!]$ except $\widetilde{[\![P]\!]} = h_{\mathbb{P}([\![X_1]\!] \times \cdots \times [\![X_m]\!])}([\![P]\!])$ for m-ary predicate symbols of type $X_1, \ldots, X_m$.

Theorem 4.9 (i) *For all formulae ϕ of **FOBBI**: ϕ is satisfiable (valid) on resource hyperdoctrines iff ϕ is satisfiable (valid) on indexed resource frames.*

(ii) *The indexed resource frame semantics of **FOBBI** is sound and complete.* □

This can be strengthened to prove a duality theorem for resource hyperdoctrines. First we augment Definition 4.3 with topological structure.

Definition 4.10 An *indexed resource space* is a functor $\mathcal{R} : \mathrm{C} \to \mathrm{ResSp}$ such that

(i) C is a category with finite products,

(ii) $Ran(\mathcal{R}(\Delta_X))$ is clopen, and

(iii) for all objects Γ, Γ' and X in C, all morphisms $s : \Gamma \to \Gamma'$ and all product projections $\pi_{\Gamma,X}$, the following square is a quasi-pullback:

$$\begin{array}{ccc} \mathcal{R}(\Gamma \times X) & \xrightarrow{\mathcal{R}(\pi_{\Gamma,X})} & \mathcal{R}(\Gamma) \\ \Big\downarrow{\scriptstyle \mathcal{R}(s \times id_X)} & & \Big\downarrow{\scriptstyle \mathcal{R}(s)} \\ \mathcal{R}(\Gamma' \times X) & \xrightarrow[\mathcal{R}(\pi_{\Gamma',X})]{} & \mathcal{R}(\Gamma') \end{array}$$

With the additional conditions that $\mathcal{R}(\pi_{\Gamma,X})$ maps open sets to open sets and $\mathcal{R}(\pi_{\Gamma,X})_*$ maps closed sets to closed sets. □

We can now combine **BBI** duality with the transformations between indexed resource frames and resource hyperdoctrines to give a dual equivalence of categories. First, we give notions of morphism for resource hyperdoctrines and indexed resource frames to obtain categories ResHyp and IndResSp. For hyperdoctrines, we adapt the definition of coherent hyperdoctrine morphism given in [19].

Definition 4.11 Given resource hyperdoctrines $\mathbb{P} : \mathrm{C}^{op} \to \mathrm{Poset}$ and $\mathbb{P}' : \mathrm{D}^{op} \to \mathrm{Poset}$, a *resource hyperdoctrine morphism* $(K, \tau) : \mathbb{P} \to \mathbb{P}'$ is a pair such that

(i) $K : \mathrm{C} \to \mathrm{D}$ is a finite product preserving functor,

(ii) $\tau : \mathbb{P} \to \mathbb{P}' \circ K$ is a natural transformation,

(iii) for all objects X in C: $\tau_{X \times X}(=_X) = ='_{K(X)}$

(iv) for all objects Γ and X in C, the following squares commute:

$$\begin{array}{ccc} \mathbb{P}(\Gamma \times X) & \xrightarrow{\tau_{\Gamma \times X}} & \mathbb{P}'(K(\Gamma) \times K(X)) \\ {\scriptstyle \exists X_\Gamma}\Big\downarrow & & \Big\downarrow{\scriptstyle \exists' K(X)_{K(\Gamma)}} \\ \mathbb{P}(\Gamma) & \xrightarrow{\tau_\Gamma} & \mathbb{P}'(K(\Gamma)) \end{array} \qquad \begin{array}{ccc} \mathbb{P}(\Gamma \times X) & \xrightarrow{\tau_{\Gamma \times X}} & \mathbb{P}'(K(\Gamma) \times K(X)) \\ {\scriptstyle \forall X_\Gamma}\Big\downarrow & & \Big\downarrow{\scriptstyle \forall' K(X)_{K(\Gamma)}} \\ \mathbb{P}(\Gamma) & \xrightarrow{\tau_\Gamma} & \mathbb{P}'(K(\Gamma)) \end{array}$$

The composition of the resource hyperdoctrine morphisms $(K, \tau) : \mathbb{P} \to \mathbb{P}'$ and $(K', \tau') : \mathbb{P}' \to \mathbb{P}''$ is given by $(K' \circ K, \tau'_{K(-)} \circ \tau)$. □

Definition 4.12 Given indexed resource spaces $\mathcal{R} : \mathrm{C} \to \mathrm{ResSp}$ and $\mathcal{R}' : \mathrm{D} \to \mathrm{ResSp}$, an *indexed resource space morphism* $(L, \lambda) : \mathcal{R} \to \mathcal{R}'$ is a pair (L, λ) such that

(i) $L : \mathrm{D} \to \mathrm{C}$ is a finite product preserving functor,

(ii) $\lambda : \mathcal{R} \circ L \to \mathcal{R}'$ is a natural transformation,

(iii) (Lift Property) if there exist x and y such that $\lambda_{X \times X}(x) = \mathcal{R}'(\Delta_X)(y)$, then there exists y' such that $\mathcal{R}(L(\Delta_X))(y') = x$, and

(iv) for all objects Γ and X in C, the following square is a quasi-pullback:

$$\begin{array}{ccc} \mathcal{R}(L(\Gamma) \times L(X)) & \xrightarrow{\lambda_{\Gamma \times X}} & \mathcal{R}(\Gamma \times X) \\ \Big\downarrow{\scriptstyle \mathcal{R}(L(\pi_{\Gamma,X}))} & & \Big\downarrow{\scriptstyle \mathcal{R}'(\pi_{\Gamma,X})} \\ \mathcal{R}(L(\Gamma)) & \xrightarrow{\lambda_\Gamma} & \mathcal{R}(\Gamma) \end{array}$$

The composition of the indexed resource space morphisms $(L, \lambda) : \mathcal{R} \to \mathcal{R}'$ and $(L', \lambda') : \mathcal{R}' \to \mathcal{R}''$ is given by $(L \circ L', \lambda' \circ \lambda_{L'(-)})$. □

Duality is given on objects by composing a resource hyperdoctrine/indexed re-

source frame with the corresponding functor from **BBI** duality. On morphisms, we take the inverse image of the natural transformation in both resource hyperdoctrine and indexed resource frame morphisms. A full proof can be found in the extended research note [22].

Theorem 4.13 (Duality Theorem for Resource Hyperdoctrines) *The categories* ResHyp *and* IndResSp *are dually equivalent.* □

5 Conclusions and Further Work

We have given a systematic treatment of Stone-type duality for the structures that interpret bunched logics, starting with the weakest systems, recovering the familiar **BBI**, and concluding with Separation Logic. Our results encompass all the known existing algebraic approaches to Separation Logic and prove them sound with respect to the standard store-heap semantics. As corollaries, we uniformly recover soundness and completeness theorems for the systems we consider.

We have also obtained analogous results for the intuitionistic variant of **LGL** (**ILGL**, developed in [23]), **BI** [36] and intuitionistic **FOBBI**, of which intuitionistic Separation Logic [31] is a specific model. Our theorems can also be extended to the bunched logics with additional multiplicatives corresponding to negation and disjunction: **dMBI** [6], **CBI** [7] and the full range of sub-classical bunched logics [10]. These results will be presented elsewhere. We conjecture that the treatment can additionally encompass a range of bunched modal and epistemic systems (e.g., [20], [9], and [26]), as well as higher-order variants of Separation Logic via general hyperdoctrines [3]. We believe this treatment will simplify completeness arguments for bunched logics by providing a modular framework within which existing results can be extended. More generally, the notion of indexed resource frame and its associated completeness argument can easily be adapted for a wide range of non-classical predicate logics.

We identify two areas of interest for further work. First, in extending our framework to encompass the breadth of the bunched logic literature we hope to give an account of multiplicative (or *bunched*) modalities [20] and quantification [17], areas which have yet to be explored algebraically. This would require the formulation of *resource algebra with operators* and a *reformulation* of resource hyperdoctrine in which the operators and adjoints (respectively) satisfy certain compatibility conditions with the monoidal structure of resource algebras. We believe the present work provides the mathematical foundation to explore these ideas.

Second, we conjecture that our approach can be extended to account for the operational semantics of program execution given by Hoare triples. As a consequence, we aim to interpret computational approaches to the Frame Rule such as bi-abduction [11] within our semantics. We believe the evident extension of our framework with the duality-theoretic approach to Hoare logic [5] can facilitate this. We wish to investigate if the duality theorems can be used to bring algebraic or topological methods to bear on these important properties of Separation Logic.

References

[1] A.R. Anderson, N. Belnap, and J.M. Dunn. *Entailment. The Logic of Relevance and Necessity. Vol. II.* Princeton University Press, Princeton, NJ, 1992. With contributions by Kit Fine, Alasdair Urquhart et al, Includes a bibliography of entailment by Robert G. Wolf.

[2] G. Anderson and D. Pym. A Calculus and Logic of Bunched Resources and Processes. *Theoretical Computer Science*, 614:63–96, 2016.

[3] B. Biering, L. Birkedal, and N. Torp-Smith. BI Hyperdoctrines and Higher-order Separation Logic. In *Proc. 14th ESOP*, 233–247, Springer-Verlag, 2005.

[4] K. Bimbó and J.M. Dunn. *Generalized Galois Logics. Relational Semantics of Nonclassical Logical Calculi*, *CSLI Lecture Notes*, Volume 188. CSLI Publications, Stanford, December 2008.

[5] C. Brink and I. Rewitzky. *A Paradigm for Program Semantics: Power Structures and Duality, Studies in Logic, Language and Information.* CSLI Publications, Stanford, CA, 2001.

[6] J. Brotherston Bunched Logics Displayed. *Studia Logica* 100(6):1223-1254, 2012.

[7] J. Brotherston and C. Calcagno. Classical BI: Its semantics and proof theory. *Logical Methods in Computer Science*, 6 (3)1-42. 2010. doi=10.2168/LMCS-6(3:3)2010.

[8] J. Brotherston and M. Kanovich. Undecidability of Propositional Separation Logic and Its Neighbours. *Journal of the ACM*, 61(2):14:1–14:43, 2014.

[9] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *SIGPLAN Notices*, 49(1):453–464, 2014.

[10] J. Brotherston and J. Villard. Sub-Classical Boolean Bunched Logics and the Meaning of Par. *Proceedings of CSL-24*, LIPlcs, Dagstuhl, 325–342, 2015.

[11] C. Calcagno, D. Distefano, P. O'Hearn, and H. Yang. Compositional Shape Analysis by Means of Bi-abduction. *Journal of the ACM*, 58(6): 66, 2011.

[12] C. Calcagno, P. Gardner, and U. Zarfaty. Context logic as modal logic: Completeness and parametric inexpressivity. in *Proc. POPL-34*. ACM, 123 –134, 2007.

[13] C. Calcagno, P. O'Hearn, and H. Yang. Local Action and Abstract Separation Logic. In *Proc. 22nd LICS*, IEEE, 2007, 366–378.

[14] M. Collinson, K. McDonald, and D. Pym. A Substructural Logic for Layered Graphs. *Journal of Logic and Computation*, 24(4):953–988, 2014.

[15] M. Collinson, K. McDonald, and D. Pym. Layered Graph Logic as an Assertion Language for Access Control Policy Models. *Journal of Logic and Computation*, 2015. doi:10.1093/logcom/exv020.

[16] M. Collinson and D. Pym. Algebra and logic for resource-based systems modelling. *Mathematical Structures in Computer Science*, 19:959–1027, 2009. doi:10.1017/S0960129509990077.

[17] M. Collinson, D. Pym and E. Robinson. Bunched Polymorphism. *Mathematical Structures in Computer Science*, 18(6):1091–1132, 2008.

[18] D. Coumans. Duality for first-order logic. http://www.math.ru.nl/~coumans/talkAC.pdf. Accessed 5 June 2017.

[19] D. Coumans. Generalising Canonical Extension to the Categorical Setting. *Annals of Pure and Applied Logic*, 163(12):1940 – 1961, 2012.

[20] J.-R. Courtault, D. Galmiche, and D. Pym. A Logic of Separating Modalities. *Theoretical Computer Science*, 637:30–58, 2016.

[21] H.-H. Dang and P. Höfner and B. Möller. Algebraic Separation Logic. *The Journal of Logic and Algebraic Programming*, 80(6):221–247, 2011.

[22] S. Docherty and D. Pym. A Stone-type Duality Theorem for Separation Logic via its Underlying Bunched Logics. Research note RN/17/06, Department of Computer Science, UCL, http://www.cs.ucl.ac.uk/fileadmin/UCL-CS/research/Research_Notes/RN_17_06.pdf, 2017.

[23] S. Docherty and D. Pym. Intuitionistic Layered Graph Logic. In *Proc IJCAR '16*. LNAI 9706:469–486, 2016.

[24] R. Dockins, A. Hobor, and A.W. Appel. A Fresh Look at Separation Algebras and Share Accounting. In *Proc. of the 7th Asian Symposium on Programming Languages and Systems*, APLAS '09, 161–177, Berlin, Heidelberg, 2009. Springer-Verlag.

[25] B. Dongol, V. Gomes, and G. Struth. A Program Construction and Verification Tool for Separation Logic. In R. Hinze and J. Voigtländer, editors, LNCS 9129, 137–158. Springer, 2015.

[26] D. Galmiche, P. Kimmel, and D. Pym. A Substructural Epistemic Resource Logic. in *Proc ICLA '17*, LNCS 10119:106–122, 2017.

[27] D. Galmiche and D. Larchey-Wendling. Expressivity Properties of Boolean BI Through Relational Models. in *Proc FSTTCS '06*, Springer Berlin Heidelberg, 357–368, 2006.

[28] D. Galmiche and D. Larchey-Wendling. Looking at Separation Algebras with Boolean BI-eyes. in *Proc TCS '14*, Springer Berlin Heidelberg, 326 –340, 2014.

[29] D. Galmiche, D. Méry, and D. Pym. The Semantics of **BI** and Resource Tableaux. *Mathematical Structures in Computer Science*, 15:1033–1088, 2005.

[30] R. Goldblatt. Varieties of Complex Algebras. *Annals of Pure and Applied Logic*, 44(3):173-242, 1989.

[31] S. Ishtiaq and P. O'Hearn. **BI** as an Assertion Language for Mutable Data Structures. In *28th ACM-SIGPLAN Symposium on Principles of Programming Languages, London*, 14–26. ACM, 2001.

[32] P. Johnstone. *Stone Spaces*. Cambridge University Press, 1986.

[33] B. Jonsson and A. Tarski. Boolean algebras with operators. Part I. *American Journal of Mathematics*, 73(4):891–939, 1951.

[34] D. Larchey-Wendling. The Formal Strong Completeness of Partial Monoidal Boolean BI. *Journal of Logic and Computation*, 26(2):605–640, 2016.

[35] P. O'Hearn. A Primer on Separation Logic. *Software Safety and Security; Tools for Analysis and Verification. NATO Science for Peace and Security Series*, 33:286–318, 2012.

[36] P. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, June 1999.

[37] A. Pitts. Categorical Logic. In *Handbook of Logic in Computer Science, Volume 5*, S. Abramsky and D. Gabbay and T. Maibaum, editors, Oxford University Press, 2000. 39–128.

[38] J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. In *Proceedings of the Seventeenth Annual IEEE Symposium on Logic in Computer Science, Copenhagen, Denmark, July 22-25, 2002*, 55–74. IEEE Computer Society Press, 2002.

[39] M. H. Stone. The Theory of Representations of Boolean Algebras. *Transactions of the American Mathematical Society* 40, 37 – 111. 1936.

[40] A. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.

[41] H. Yang and P. O'Hearn. A Semantic Basis for Local Reasoning. In *Proc. FOSSACS'02*, 2002. doi:10.1.1.10.8768.