

A Unified Display Proof Theory for Bunched Logic

James Brotherston¹

*Dept. of Computing
Imperial College
London, UK*

Abstract

We formulate a unified display calculus proof theory for the four principal varieties of bunched logic by combining display calculi for their component logics. Our calculi satisfy cut-elimination, and are sound and complete with respect to their standard presentations. We show that the standard sequent calculus for BI can be seen as a reformulation of its display calculus, and argue that analogous sequent calculi for the other varieties of bunched logic seem very unlikely to exist.

Keywords: bunched logic, display calculus, proof theory, cut-elimination, substructural logic

1 Introduction

Bunched logics, originating in O’Hearn and Pym’s BI [18], are a relatively recent addition to the menagerie of substructural logics and are increasingly attracting interest amongst computer science researchers as well as logicians. Of their better-established cousins, bunched logics most resemble *relevant logics* [21] in that they feature both multiplicative (or intensional) and additive (or extensional) logical connectives, with the difference between the two types characterised as a matter of which structural principles are admitted by each. However, while in relevant logics certain of the additive connectives are barred in order to exclude various philosophically controversial theorems from the logic, in bunched logics one simply takes a *complete* set of additive connectives in addition to a set of multiplicatives. Thus bunched logics can be seen as the result of freely combining a standard (additive) propositional logic with a (multiplicative) linear logic. This simple-minded treatment of the additives gives rise to a Tarskian *resource interpretation* of formulas

¹ Research supported by an EPSRC Postdoctoral Fellowship. Thanks to Rajeev Goré, Greg Restall and the East London Massive (as was) for illuminating discussions and advice, and to the anonymous referees for suggestions on improving the paper.

in bunched logics, which are read as true or false relative to resources: roughly speaking, the additives have their standard propositional meanings while the multiplicatives denote resource composition properties [20]. In computer science, such resource readings of bunched logic have very successfully been exploited to obtain logics for program analysis. Most notably, *separation logic* [24], which is based upon an interpretation of resources as portions of heap memory, has spawned a host of program analysis applications that discover and reason about the structure of heap memory during program execution (recent examples include [7,8,10]). Bunched logic has also been variously employed in addressing other computing problems such as polymorphic abstraction [9], tree update [6] and typed reference update and disposal [2].

In this paper, we examine bunched logic from the general proof-theoretic perspective. While there has been considerable interest in the semantics of bunched logics, justified in no small part by the computational significance of the resulting models [11,12,14,20], their proof theory by contrast has received comparatively little attention. As observed by Pym [19], it is natural to consider four principal varieties of bunched logic, characterised by the presence or otherwise of classical negation in the additive and multiplicative fragments or, equivalently, by the underlying additive and multiplicative algebras (see Figure 1). However, to date there has been no proof-theoretical analysis corresponding to this general characterisation. On the one hand, O’Hearn and Pym’s original bunched logic BI is known to possess both a complete natural deduction proof system satisfying normalisation, and a complete sequent calculus satisfying cut-elimination [19]. On the other hand, analogues of these syntactic proof systems for well-known variants of BI such as Boolean BI (BBI) have been conspicuously absent from the literature². Instead, proof systems for BBI are usually obtained in a crude manner by adding a sufficiently powerful axiom or inference rule to the corresponding proof system for BI. Such additions typically break normalisation and cut-elimination properties, which is less than ideal from the theoretical point of view but also from a practical perspective, since separation logic and many of the aforementioned related program analysis tools are based on BBI. However, extending the BI sequent calculus to BBI without breaking cut-elimination is highly problematic.

An alternative route to a disciplined proof theory for general bunched logics is suggested by Belnap’s *display logic* [1], which was historically employed as a device for giving consecution calculi *à la* Gentzen for relevant and modal logics. The distinguishing feature of display calculi is the *display property*: any consecution can always be rearranged so that a given part appears alone on the appropriate side of the proof turnstile. To ensure this property we need both a richer form of consecution than that of typical Gentzen-style sequents, and a set of auxiliary “display” rules for rearranging them in the required fashion. Compensation for this extra complexity comes in the form of an elegant, symmetric presentation of the calculus, analogous to that of Gentzen’s sequent calculi. Furthermore, cut-

² A tableau proof system for a variant of BBI based on functional rather than the usual relational monoidal semantics is given in [16], but it does not correspond directly to a cut-free sequent calculus, owing to its reliance on semantic constraint labels.

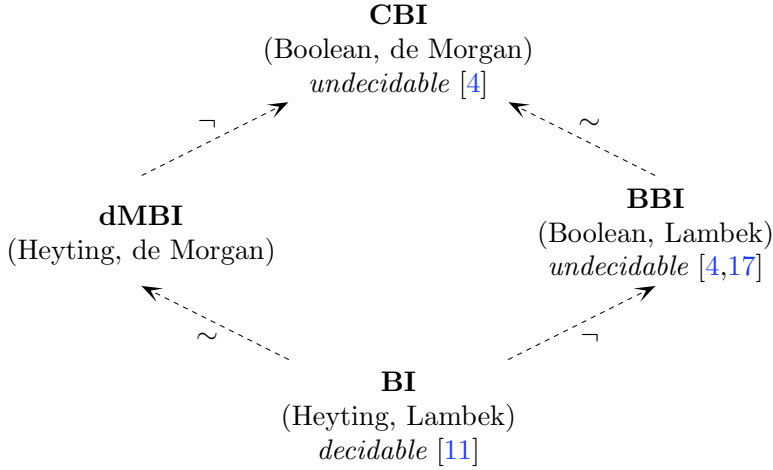


Fig. 1. The bunched logic family. The (additive, multiplicative) subtitles indicate the underlying additive and multiplicative algebras. The arrows denote the addition of either additive (\neg) or multiplicative (\sim) classical negation.

elimination is guaranteed for any display calculus whose rules obey a set of easily verifiable syntactic conditions.

In earlier work with Calcagno, we showed that the “classical” bunched logic CBI could be naturally presented as a display calculus [3]. In this paper, we obtain a *unified* display calculus proof theory for all four principal bunched logics in Figure 1. First, we formulate display calculi for the elementary logics which characterise the additive and multiplicative components of the various logics. Since Belnap’s original display apparatus does not adapt to the intuitionistic components (because it relies on the presence of classical negation), we instead exploit the residual relationship between conjunction and implication to obtain a display property, *à la* Goré [13] and Restall [23]. Then, we obtain display calculi for the bunched logics by combining the display calculi for the corresponding elementary components. This composition of the elementary calculi preserves several of their desirable structural properties, chiefly cut-elimination and soundness / completeness with respect to standard presentations of the corresponding logics. Additionally, we establish a translation between cut-free proofs in our display calculus for BI and those in its standard sequent calculus, showing that the latter system can be seen as a refined version of the former. The fact that our display calculi for the other bunched logics seemingly cannot be similarly optimised into sequent presentations — due to their essential use of *unary* structural connectives as well as the usual binary ones — goes some way to explaining why well-behaved sequent calculi for these logics have been so elusive.

The remainder of this paper is organised as follows. In Section 2 we define the four main bunched logics in Figure 1 in terms of their elementary component logics. Section 3 introduces the apparatus of display logic. In Section 4 we give display calculi for the bunched logics via display calculi for their elementary component logics. In Section 5 we compare the resulting display calculus for BI with its bunched sequent calculus. Section 6 concludes.

2 From elementary logics to bunched logics

In this section, we define the four principal bunched logics (cf. Figure 1) as free combinations of well-known elementary logics.

We assume a fixed infinite set \mathcal{V} of propositional variables. *Formulas* are constructed from propositional variables using the logical connectives given in Figure 2: any $P \in \mathcal{V}$ is a formula, and so is the result of applying a logical connective to the appropriate number of formulas. We restrict the syntax of formulas in a particular logic by stipulating which formula connectives are permitted to occur. We write F, G, H , etc., to range over formulas.

Additive symbol:	\top	\perp	\neg	\wedge	\vee	\rightarrow
Multiplicative symbol:	\top^*	\perp^*	\sim	$*$	\wp	\multimap
Arity:	0	0	1	2	2	2

Fig. 2. Logical connectives.

We regard a *logic* \mathcal{L} as being specified by: (a) the set of logical connectives which may occur in formulas of the logic; and (b) a basic proof system for entailments of the form $F \vdash G$, where F and G are formulas. We write $F \dashv\vdash G$ to abbreviate two axioms $F \vdash G$ and $G \vdash F$. We specify four well-known elementary logics, which form the principal components of the bunched logics in Figure 1, as follows:

- Intuitionistic logic, IL, has as logical connectives \top , \perp , \wedge , \vee and \rightarrow . Classical logic, CL, adds the negation \neg . We present IL and CL in Figure 3.
- Lambek multiplicative logic, LM (a.k.a. multiplicative intuitionistic linear logic), has as logical connectives the multiplicative \top^* , $*$ and \multimap . De Morgan multiplicative logic, dMM (a.k.a. multiplicative classical linear logic), extends these by \perp^* , \sim and \wp . We present LM and dMM in Figure 4.

$F \vdash F$	$F \vdash \top$	$\perp \vdash F$	$G_i \vdash G_1 \vee G_2 \ (i \in \{1, 2\})$	$G_1 \wedge G_2 \vdash G_i \ (i \in \{1, 2\})$
$F \vdash H \quad G \vdash H$	$F \vdash G \quad F \vdash H$	$F \vdash G \rightarrow H$	$F \wedge G \vdash H$	$F \vdash G \quad G \vdash H$
$F \vee G \vdash H$	$F \vdash G \wedge H$	$F \wedge G \vdash H$	$F \vdash G \rightarrow H$	$F \vdash H$
.....				
$\neg F \dashv\vdash F \rightarrow \perp$		$\neg\neg F \vdash F$		

Fig. 3. IL and CL. The axioms below the dotted line are present in CL only.

$F \vdash F$	$F * (G * H) \dashv\vdash (F * G) * H$	$F * G \vdash G * F$	$F * \top^* \dashv\vdash F$
$F_1 \vdash G_1 \quad F_2 \vdash G_2$	$F * G \vdash H$	$F \vdash G \multimap H$	$F \vdash G \quad G \vdash H$
$F_1 * F_2 \vdash G_1 * G_2$	$F \vdash G \multimap H$	$F * G \vdash H$	$F \vdash H$
.....			
$\perp^* \dashv\vdash \sim \top^*$	$F \wp G \dashv\vdash \sim(\sim F * \sim G)$	$\sim F \dashv\vdash F \multimap \perp^*$	$\sim\sim F \vdash F$

Fig. 4. LM and dMM. The axioms below the line are present in dMM only.

We write $\mathcal{E} = \{\text{IL}, \text{CL}, \text{LM}, \text{dMM}\}$ for this set of elementary logics. By the *free combination* $\mathcal{L}_1 + \mathcal{L}_2$ of two logics $\mathcal{L}_1, \mathcal{L}_2 \in \mathcal{E}$, we mean the logic whose logical connectives and presentation are the unions of, respectively, the logical connectives

Additive	Multiplicative	Arity	Antecedent meaning	Consequent meaning
\emptyset	\emptyset	0	truth	falsity
\sharp	\flat	1	negation	negation
$;$	$,$	2	conjunction	disjunction
\Rightarrow	\multimap	2	undefined	implication

Fig. 5. Structural connectives.

and the presentations of \mathcal{L}_1 and \mathcal{L}_2 . The bunched logics $\mathcal{B} = \{\text{BI}, \text{BBI}, \text{dMBI}, \text{CBI}\}$ can then be defined very straightforwardly in terms of their elementary components:

- BI, a.k.a. the logic of bunched implications (cf. [18,20]), is given by IL + LM;
- BBI, a.k.a. Boolean BI (cf. [12]), is given by CL + LM;
- dMBI, standing for “de Morgan BI”, is given by IL + dMM;
- CBI, a.k.a. Classical BI (cf. [3]), is given by CL + dMM.

Our definition of the logics $\mathcal{E} \cup \mathcal{B}$ above will be taken as the baseline with respect to which our display calculi for these logics are later proven correct. This has the benefit of freeing our analysis from unnecessary semantic considerations. We note that our definitions of \mathcal{B} can be seen to be in agreement with those found elsewhere in the literature, as well as the characterisation in Figure 1. For example, our presentations of BI and BBI agree with their counterparts in [20] and [12] respectively. (To our knowledge, dMBI has not appeared in the literature before, while CBI was presented in [3] via a display calculus, which we will reconstruct as part of our unified proof theory for \mathcal{B} .)

3 Display calculus fundamentals

In this section we present the basic notions that we require in order to specify a *display calculus* in the spirit of Belnap [1].

Structures are constructed from formulas using the structural connectives given by Figure 5: any formula is a structure, and so is the result of applying a structural connective to the appropriate number of structures. We write W, X, Y, Z , etc., to range over structures. If X and Y are structures then $X \vdash Y$ is called a *consecution*. There is a classification of the structure occurrences in a consecution into *antecedent* and *consequent* parts, which generalises the simple left-right division of sequent calculus.

Definition 3.1 (Antecedent / consequent part) *Substructure occurrences in a structure X are classified as either positive or negative in X , as follows:*

- X is positive in X ;
- if Z is negative (positive) in Y then Z is positive (negative) in $\sharp Y$ and $\flat Y$;
- if Z is positive (negative) in X_1 or X_2 then Z is positive (negative) in $X_1 ; X_2$ and X_1 , X_2 ;
- if Z is negative (positive) in X_1 or positive (negative) in X_2 , then Z is positive (negative) in $X_1 \Rightarrow X_2$ and $X_1 \multimap X_2$.

Z is said to be an antecedent (consequent) part of a consecution $X \vdash Y$ if it is positive (negative) in X or negative (positive) in Y .

Consecutions are interpreted as entailments between formulas as follows.

Definition 3.2 (Consecution validity) For any structure X we define the formulas Ψ_X and Υ_Y by mutual structural induction as follows:

$$\begin{array}{llll}
 \Psi_F = F & & \Upsilon_F = F & \\
 \Psi_\emptyset = \top & \Psi_\emptyset = \top^* & \Upsilon_\emptyset = \perp & \Upsilon_\emptyset = \perp^* \\
 \Psi_{\sharp X} = \neg \Upsilon_X & \Psi_{\flat X} = \sim \Upsilon_X & \Upsilon_{\sharp X} = \neg \Psi_X & \Upsilon_{\flat X} = \sim \Psi_X \\
 \Psi_{X;Y} = \Psi_X \wedge \Psi_Y & \Psi_{X,Y} = \Psi_X * \Psi_Y & \Upsilon_{X;Y} = \Upsilon_X \vee \Upsilon_Y & \Upsilon_{X,Y} = \Upsilon_X \dot{\vee} \Upsilon_Y \\
 \Psi_{X \Rightarrow Y} = \text{undefined} & \Psi_{X \multimap Y} = \text{undefined} & \Upsilon_{X \Rightarrow Y} = \Psi_X \rightarrow \Upsilon_Y & \Upsilon_{X \multimap Y} = \Psi_X \multimap \Upsilon_Y
 \end{array}$$

$X \vdash Y$ is said to be valid in a logic \mathcal{L} iff $\Psi_X \vdash \Upsilon_Y$ is provable in \mathcal{L} .

We remark that, in any given display calculus, we restrict the form of consecutions by stipulating which of the structural connectives may appear as the main (i.e. outermost) connective of an antecedent or consequent part. In doing so, we ensure that the restrictions on the structural connectives match the available formula connectives, so that validity of consecutions is always well defined. In particular, neither \Rightarrow nor \multimap will ever be permitted to appear as the main connective of an antecedent part of a consecution.

The defining feature of any true display calculus is the availability of an equivalence relation on consecutions, called a *display-equivalence*, that facilitates their rearrangement into an equivalent consecution in which a given antecedent (consequent) part appears as the *entire* antecedent (consequent).

Definition 3.3 (Display-equivalence) Let \equiv_D be the least equivalence generated by a set of rules of the form $C \lessgtr_D C'$, where C, C' are consecutions. We say that \equiv_D is a display-equivalence if, for any antecedent (consequent) part Z of $X \vdash Y$, one can construct a structure W such that $X \vdash Y \equiv_D Z \vdash W$ ($X \vdash Y \equiv_D W \vdash Z$). The process of rearranging $X \vdash Y$ into $Z \vdash W$ or $W \vdash Z$ via display-equivalence is called displaying Z .

A display calculus $DL_{\mathcal{L}}$ for a logic \mathcal{L} is then specified by the following:

Antecedent / consequent structural connectives: The structural connectives that are permitted to appear as the main connective of an antecedent / consequent part of a consecution, respectively.

Display postulates: A set of rules of the form $C \lessgtr_D C'$ such that the least equivalence \equiv_D generated by the rules is a display-equivalence.

Logical rules: Proof rules for the formula connectives, typically divided into pairs of left- and right-introduction rules for each logical connective in the manner familiar from sequent calculus (though, like in sequent systems, some connectives may have only one introduction rule). Note that, since we can appeal to the display-equivalence \equiv_D , these rules may be written so that the formula introduced by a rule is displayed (alone) in its conclusion.

Structural rules: Proof rules for the structural connectives. We write a rule with a double-line between premise and conclusion to indicate that it is *symmetric*,

Antecedent structure connectives: \emptyset ;

Consequent structure connectives: \Rightarrow

Display postulates: $X ; Y \vdash Z \text{ } <>_D \text{ } X \vdash Y \Rightarrow Z \text{ } <>_D \text{ } Y ; X \vdash Z$

Logical rules:

$$\begin{array}{c}
 \frac{}{\perp \vdash X} (\perp L) \quad \frac{\emptyset \vdash X}{\top \vdash X} (\top L) \quad \frac{F ; G \vdash X}{F \wedge G \vdash X} (\wedge L) \quad \frac{F \vdash X \quad G \vdash X}{F \vee G \vdash X} (\vee L) \quad \frac{X \vdash F \quad G \vdash Y}{F \rightarrow G \vdash X \Rightarrow Y} (\rightarrow L) \\
 \\
 \frac{}{X \vdash \top} (\top R) \quad \frac{X \vdash F \quad X \vdash G}{X \vdash F \wedge G} (\wedge R) \quad \frac{X \vdash F_i \quad i \in \{1, 2\}}{X \vdash F_1 \vee F_2} (\vee R) \quad \frac{X ; F \vdash G}{X \vdash F \rightarrow G} (\rightarrow R)
 \end{array}$$

Structural rules:

$$\frac{\emptyset ; X \vdash Y}{X \vdash Y} (\emptyset L) \quad \frac{W ; (X ; Y) \vdash Z}{(W ; X) ; Y \vdash Z} (AAL) \quad \frac{X \vdash Z}{X ; Y \vdash Z} (WkL) \quad \frac{X ; X \vdash Y}{X \vdash Y} (CtrlR)$$

Fig. 6. Specification of DL_{IL} .

i.e., that the premise and conclusion may be exchanged.

In addition to the logical and structural proof rules given by their specification, all of our display calculi share a common set of *identity rules*:

$$\begin{array}{c}
 P \vdash P \quad (Id) \quad \frac{X \vdash F \quad F \vdash Y}{X \vdash Y} (Cut) \quad \frac{X' \vdash Y'}{X \vdash Y} \quad X \vdash Y \equiv_D X' \vdash Y' \quad (\equiv_D)
 \end{array}$$

where P ranges over propositional variables. We remark that a display calculus so specified is not guaranteed to obey any particular proof-theoretic properties over and above the availability of display-equivalence; as is well-known, display calculi may fail to enjoy cut-elimination, interpolation, or decidability. However, cut-elimination is guaranteed for display calculi with sufficiently well-behaved logical and structural rules [1].

4 Display calculi for bunched logics

In this section we give display calculi for the elementary logics \mathcal{E} (see Section 2) and combine them to obtain display calculi for the bunched logics \mathcal{B} . We give display calculus specifications for the elementary logics IL, CL, LM and dMM in Figures 6, 7, 8 and 9 respectively.

Some remarks on our formulation of these elementary display calculi are in order. Firstly, the display postulates for the classical logics CL and dMM essentially follow Belnap [1]. These postulates do not adapt to the intuitionistic IL and LM because they lack suitable involutive negations necessary to interpret \sharp and \flat . Instead, we employ the meta-level implications \Rightarrow and \multimap , and exploit their residual connections with the conjunctions to obtain suitable display postulates (an idea also employed by Goré [13] and Restall [23]).

Secondly, because the structural connectives \emptyset and $;$ may occur only in antecedent positions in DL_{IL} consecutions, we must use structure-free formulations of the rules $(\vee L)$ and $(\perp L)$ (for convenience, we do the same for $(\wedge R)$ and $(\top R)$, and use the same rules for DL_{CL}). Also, we could have written a single version of $(\rightarrow L)$

Antecedent structure connectives: $\emptyset \quad \sharp \quad ;$					
Consequent structure connectives: $\emptyset \quad \sharp \quad ;$					
Display postulates: $X ; Y \vdash Z \quad <>_D \quad X \vdash \sharp Y ; Z \quad <>_D \quad Y ; X \vdash Z$					
$X \vdash Y ; Z \quad <>_D \quad X ; \sharp Y \vdash Z \quad <>_D \quad X \vdash Z ; Y$					
$X \vdash Y \quad <>_D \quad \sharp Y \vdash \sharp X \quad <>_D \quad \sharp \sharp X \vdash Y$					
Logical rules:			Structural rules:		
$\frac{X \vdash \emptyset}{X \vdash \perp} (\perp R)$	$\frac{\sharp F \vdash X}{\neg F \vdash X} (\neg L)$	$\frac{X \vdash \sharp F}{X \vdash \neg F} (\neg R)$	$\frac{X \vdash Y ; \emptyset}{X \vdash Y} (\emptyset R)$		
$\frac{X \vdash F \quad G \vdash Y}{F \rightarrow G \vdash \sharp X ; Y} (\rightarrow L)$	$\frac{X \vdash F ; G}{X \vdash F \vee G} (\vee R)$				

Fig. 7. Specification of DL_{CL} . The logical and structural rules extend those of DL_{IL} (Figure 6), except for $(\vee R)$ and $(\rightarrow L)$ which replace their DL_{IL} counterparts.

Antecedent structure connectives: $\emptyset \quad ,$					
Consequent structure connectives: \multimap					
Display postulates: $X, Y \vdash Z \quad <>_D \quad X \vdash Y \multimap Z \quad <>_D \quad Y, X \vdash Z$					
Logical rules:			Structural rules:		
$\frac{\emptyset \vdash X}{\top^* \vdash X} (\top^* L)$	$\frac{F, G \vdash X}{F * G \vdash X} (*L)$	$\frac{X \vdash F \quad G \vdash Y}{F \multimap G \vdash X \multimap Y} (*L)$	$\frac{\emptyset, X \vdash Y}{X \vdash Y} (\emptyset L)$		
$\frac{X \vdash F \quad Y \vdash G}{X, Y \vdash F * G} (*R)$	$\frac{X, F \vdash G}{X \vdash F \multimap G} (*R)$		$\frac{W, (X, Y) \vdash Z}{(W, X), Y \vdash Z} (MAL)$		

Fig. 8. Specification of DL_{LM} .

common to both DL_{IL} and DL_{CL} with conclusion $X ; F \rightarrow G \vdash Y$, and similarly for $(\multimap L)$ in DL_{LM} and DL_{dMM} . We use separate versions in order to maintain the pleasant property that the formula introduced by a logical rule is always displayed in its conclusion.

Finally, we note that the right-hand analogues of the structural rules (AAL), (CtrL) and (WkL) are derivable in DL_{CL} . Likewise, the right-hand analogue of (MAL) is derivable in DL_{dMM} .

Now we obtain display calculi for \mathcal{B} by defining, for $\mathcal{L}_1, \mathcal{L}_2 \in \mathcal{E}$:

$$DL_{\mathcal{L}_1 + \mathcal{L}_2} =_{\text{def}} DL_{\mathcal{L}_1} + DL_{\mathcal{L}_2}$$

where $DL_{\mathcal{L}_1} + DL_{\mathcal{L}_2}$ is the display calculus whose antecedent and consequent structure connectives, display postulates, and logical and structural rules are in each case those of $DL_{\mathcal{L}_1}$ plus those of $DL_{\mathcal{L}_2}$. We observe that DL_{CBI} as presented here is equivalent to its earlier formulation in [3], while DL_{BI} , DL_{BBI} and DL_{dMBI} are new. However, DL_{dMBI} can be seen to be very nearly equivalent to Restall's display calculus for the well-known relevant logic **RW** (a.k.a. C) obtained from **R** by removing the multiplicative contraction rule [23]. The two calculi differ only because **RW** lacks the additive intuitionistic \rightarrow and \perp of **dMBI** (which can however be added conservatively).

We now demonstrate that each of our specifications does indeed give rise to a true display calculus, in the sense that the display property holds.

Antecedent structure connectives: $\emptyset \quad \flat$,	
Consequent structure connectives: $\emptyset \quad \flat$,	
Display postulates: $X, Y \vdash Z \text{ } <>_D \text{ } X \vdash \flat Y, Z \text{ } <>_D \text{ } Y, X \vdash Z$ $X \vdash Y, Z \text{ } <>_D \text{ } X, \flat Y \vdash Z \text{ } <>_D \text{ } X \vdash Z, Y$ $X \vdash Y \text{ } <>_D \text{ } \flat Y \vdash \flat X \text{ } <>_D \text{ } \flat \flat X \vdash Y$	
Logical rules:	Structural rules:
$\frac{}{\perp^* \vdash \emptyset} (\perp^*L) \quad \frac{\flat F \vdash X}{\sim F \vdash X} (\sim L) \quad \frac{F \vdash X \quad G \vdash Y}{F \flat G \vdash X, Y} (\flat L) \quad \frac{X \vdash F \quad G \vdash Y}{F \multimap G \vdash \flat X, Y} (\multimap L)$	$\frac{X \vdash Y, \emptyset}{X \vdash Y} (\emptyset R)$
$\frac{X \vdash \emptyset}{X \vdash \perp^*} (\perp^*R) \quad \frac{X \vdash \flat F}{X \vdash \sim F} (\sim R) \quad \frac{X \vdash F, G}{X \vdash F \flat G} (\flat R)$	

Fig. 9. Specification of DL_{dMM} . The logical and structural rules extend those of DL_{LM} above, except for $(\multimap L)$ which replaces its DL_{LM} counterpart.

Proposition 4.1 (Display) *For all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$, the least equivalence \equiv_D induced by the display postulates of $DL_{\mathcal{L}}$ is a display-equivalence for $DL_{\mathcal{L}}$.*

Proof. (Sketch) The required display property (Defn. 3.3) follows from the fact that, for any consecution $X \vdash Y$, the display postulates of $DL_{\mathcal{L}}$ facilitate the display of each of the immediate substructures of X and Y (as the antecedent or consequent as appropriate); it follows by induction that arbitrary substructures of $X \vdash Y$ can be displayed. This fact may be verified essentially by eye for each $\mathcal{L} \in \mathcal{E}$, noting that the structure of consecutions and display postulates in DL_{IL} and DL_{CL} is isomorphic to that of DL_{LM} and DL_{dMM} respectively. It follows immediately that the immediate substructures of $X \vdash Y$ can be displayed for each $\mathcal{L} \in \mathcal{B}$, using the display postulates from the appropriate component calculus. \square

Proposition 4.2 (Soundness) *For all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$, if a consecution of $DL_{\mathcal{L}}$ is provable in $DL_{\mathcal{L}}$ then it is valid.*

Proof. (Sketch) It suffices to prove that each rule of $DL_{\mathcal{L}}$ is locally sound in that, if all of its premises are valid, then so is its conclusion. In practice this means deriving the rule in \mathcal{L} under the translation from consecutions to formula entailments given by Defn. 3.2. This may straightforwardly, if somewhat tediously, be carried out for each of the elementary logics $\mathcal{L} \in \mathcal{E}$. For $\mathcal{L} \in \mathcal{B}$, the local soundness property is then immediate since it is obvious that, if any $DL_{\mathcal{L}_i}$ rule is \mathcal{L}_i -derivable under translation for $i \in \{1, 2\}$, then the rules of $DL_{\mathcal{L}_1} + DL_{\mathcal{L}_2}$ are derivable under translation in $\mathcal{L}_1 + \mathcal{L}_2$. \square

Lemma 4.3 (Identity) *For all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$, and for any formula F of \mathcal{L} , the consecution $F \vdash F$ is provable in $DL_{\mathcal{L}}$.*

Proof. (Sketch) By structural induction on F , distinguishing a case for every possible logical connective of \mathcal{L} . In the case $F = P \in \mathcal{V}$ we are immediately done by (Id). The connective cases follow straightforwardly by applying the left- and right-introduction rules for the connective together with the induction hypothesis (for the additives the structural rules are also required). \square

Lemma 4.4 *For all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$, and for any consecution $X \vdash Y$ of $\text{DL}_{\mathcal{L}}$, the consecutions $X \vdash \Psi_X$ and $\Upsilon_Y \vdash Y$ are $\text{DL}_{\mathcal{L}}$ -provable, where Ψ_- and Υ_- are the functions given in Definition 3.2.*

Proof. (Sketch) By induction on the number of structural connectives in $X \vdash Y$, distinguishing a case for every possible structural connective of X and Y in $\text{DL}_{\mathcal{L}}$. In the base case X and Y are formulas of \mathcal{L} , and we are done by Lemma 4.3. We show a typical connective case, $Y = (X' \Rightarrow Y')$, in which case $\text{DL}_{\mathcal{L}}$ contains DL_{IL} and we have $\Upsilon_Y = \Psi_{X'} \rightarrow \Upsilon_{Y'}$. As $X' \Rightarrow Y'$ is a consequent part of $X \vdash Y$ by the case assumption, X' is an antecedent part and Y' a consequent part, so $X' \vdash Y'$ is a consecution of $\text{DL}_{\mathcal{L}}$ and thus by induction hypothesis $X' \vdash \Psi_{X'}$ and $\Upsilon_{Y'} \vdash Y'$ are both $\text{DL}_{\mathcal{L}}$ -provable. Thus by applying the DL_{IL} rule ($\rightarrow\text{L}$) we obtain the required $\text{DL}_{\mathcal{L}}$ proof of $\Psi_{X'} \rightarrow \Upsilon_{Y'} \vdash X' \Rightarrow Y'$. \square

Theorem 4.5 (Completeness) *For all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$, if a consecution of $\text{DL}_{\mathcal{L}}$ is valid then it is provable in $\text{DL}_{\mathcal{L}}$.*

Proof. (Sketch) Suppose that $X \vdash Y$ is $\text{DL}_{\mathcal{L}}$ -valid, i.e. that $\Psi_X \vdash \Upsilon_Y$ is \mathcal{L} -provable. To show that $X \vdash Y$ is $\text{DL}_{\mathcal{L}}$ -provable, it suffices by Lemma 4.4 and (Cut) to show that $\Psi_X \vdash \Upsilon_Y$ is $\text{DL}_{\mathcal{L}}$ -provable. In practice this simply entails showing that each proof rule of \mathcal{L} is $\text{DL}_{\mathcal{L}}$ -derivable, which is a routine exercise for each $\mathcal{L} \in \mathcal{E}$. The result then follows immediately for all $\mathcal{L} \in \mathcal{B}$ because it is clear that, if $\text{DL}_{\mathcal{L}_1}$ and $\text{DL}_{\mathcal{L}_2}$ can derive every rule of \mathcal{L}_1 and \mathcal{L}_2 respectively, then $\text{DL}_{\mathcal{L}_1} + \text{DL}_{\mathcal{L}_2}$ can derive every rule of $\mathcal{L}_1 + \mathcal{L}_2$. \square

We call a display calculus proof *cut-free* if it contains no instances of (Cut).

Theorem 4.6 (Cut-elimination) *For all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$, any $\text{DL}_{\mathcal{L}}$ proof of $X \vdash Y$ can be transformed into a cut-free proof of $X \vdash Y$.*

Proof. (Sketch) Given the display property (Proposition 4.1), it suffices to verify for each $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$ that the proof rules of $\text{DL}_{\mathcal{L}}$ meet Belnap’s well-known conditions C1–C8 guaranteeing cut-elimination [1]. Moreover, since conditions C1–C7 are properties of individual rules rather than sets of rules, these may be verified for all $\mathcal{L} \in \mathcal{E} \cup \mathcal{B}$ in one pass, simply by examining each rule appearing in Figures 6–9. Condition C8 (elimination of principal cuts) depends essentially on the form of the logical rule pairs $(\bullet\text{L})$ and $(\bullet\text{R})$ for each logical connective \bullet (though the structural rules are also sometimes required to eliminate principal cuts in DL_{IL} and DL_{CL}). C8 may be easily verified for each $\mathcal{L} \in \mathcal{E}$, and then follows immediately for each $\mathcal{L} \in \mathcal{B}$ since every principal cut in $\text{DL}_{\mathcal{L}}$ is a principal cut in one of its elementary component calculi. \square

We remark that cut-free proofs in our display calculi enjoy a traditional subformula property, as can be seen by inspection of the proof rules. However, the analogous “substructure” property does not hold: the premise of a rule instance may contain structures which are not substructures of any structure in the conclusion³.

³ Of course exactly the same is true, e.g., of any sequent calculus with a contraction rule.

Thus, like in linear logic, cut-elimination does not necessarily entail decidability or interpolation. Indeed, in collaboration with Kanovich we have recently shown both BBI and CBI, among other formalisms, to be undecidable [4]⁴. Cut-elimination in display calculi remains meaningful for the reason that it eliminates the infinite branching points provided by the cut rule; as is clear by inspection of the proof rules, for any consecution there are only finitely many ways of applying a proof rule backwards⁵. Thus an exhaustive proof search is finitely branching, albeit possibly non-terminating. Moreover, for certain display calculi it is possible to ensure that such a proof search is indeed terminating [23] — although it is sadly impossible to generally determine which ones [15].

5 Relationship to bunched sequent calculi

Of all the bunched logics \mathcal{B} , only BI is known to possess a sequent calculus with cut-elimination, given by Pym [19]. Thus it is natural to compare this calculus, LBI, with our display calculus DL_{BI} . The sequents of LBI are of the form $\Gamma \vdash F$ where F is a BI-formula and Γ is a *bunch*, given by:

$$\Gamma ::= F \mid \emptyset \mid \emptyset \mid \Gamma ; \Gamma \mid \Gamma , \Gamma$$

where F ranges over BI-formulas. A *coherent equivalence* \equiv is defined on bunches as the least congruence closed under associativity and commutativity of the semicolon and comma, and under the equations $(\emptyset ; F) \equiv F \equiv (\emptyset , F)$. The right-introduction rules for the logical connectives have standard intuitionistic formulations. The left-introduction rules, and the structural rules, are written so as to apply to formulas occurring at arbitrary positions within a bunch, using the notation $\Gamma(\Delta)$ for a bunch Γ with a distinguished sub-bunch occurrence Δ . For example, the rules for \multimap in LBI are:

$$\begin{array}{ccc} \Delta \vdash F_1 & \Gamma(F_2) \vdash F & \\ \Gamma(\Delta , F_1 \multimap F_2) \vdash F & \text{(\multimap L)} & \Gamma , F \vdash G \\ & & \Gamma \vdash F \multimap G \text{ (\multimap R)} \end{array}$$

We demonstrate a correspondence between cut-free proofs in LBI and in DL_{BI} .

Lemma 5.1 *There is an injective, constructive map from LBI proofs to DL_{BI} proofs. Moreover, this map preserves cut-freeness of proofs.*

Proof. (Sketch) First note that every LBI sequent is a DL_{BI} consecution, as bunches are exactly the structures that can occur as antecedent parts of the latter. We show that each of the proof rules of LBI is derivable in DL_{BI} . The left-introduction rules can be seen in DL_{BI} as a macro for first displaying the active part of the conclusion, then applying the corresponding left-introduction rule of DL_{BI} and finally reversing the original display process to restore the bunch context. E.g.,

⁴ An alternative proof of the undecidability of BBI was produced independently [17].

⁵ Provided that one is careful to prevent arbitrary “stacking” of \sharp and \flat , cf. [15,23].

we derive the $(\multimap L)$ rule of LBI as follows:

$$\begin{array}{c} \Gamma(F_2) \vdash F \\ \hline \Delta \vdash F_1 \quad F_2 \vdash X \quad (\equiv_D) \\ \Delta, F_1 \multimap F_2 \vdash X \quad (\multimap L) \\ \hline \Gamma(\Delta, F_1 \multimap F_2) \vdash F \quad (\equiv_D) \end{array}$$

where X is a placeholder for the consequent structure that results from displaying Z in $\Gamma(Z) \vdash F$. The other left-rules are similar. The right-introduction rules of LBI have direct equivalents in DL_{BI} , and applications of the rule for coherent equivalence \equiv are translated into DL_{BI} as combinations of the display-equivalence rule (\equiv_D) , the associativity rules (AAL) and (MAL) and the unit rules $(\emptyset L)$ and $(\emptyset R)$. \square

Lemma 5.2 *There is a constructive map from DL_{BI} proofs to LBI proofs. Moreover, this map preserves cut-freeness of proofs.*

Proof. (Sketch) For any DL_{BI} consecution $X \vdash Y$ define its *display-normal form* $\lceil X \vdash Y \rceil$ to be the consecution obtained by applying the transformations $(X \vdash Y \Rightarrow Z) \mapsto (X ; Y \vdash Z)$ and $(X \vdash Y \multimap Z) \mapsto (X, Y \vdash Z)$ until no further transformations are possible. Note that for any DL_{BI} consecution $X \vdash Y$ we have that X is a bunch and $\lceil X \vdash Y \rceil$ is a unique LBI sequent of the form $\Gamma(X) \vdash F$. We show that each proof rule of DL_{BI} is derivable in LBI under the translation $\lceil - \rceil$. For example, in the case of the DL_{BI} rule $(\multimap L)$ we have:

$$\begin{array}{c} \lceil X \vdash F \rceil \quad \lceil G \vdash Y \rceil \\ \hline \lceil F \multimap G \vdash X \multimap Y \rceil \end{array} = \begin{array}{c} X \vdash F \quad \Gamma(G) \vdash H \\ \hline \Gamma(X, F \multimap G) \vdash H \end{array}$$

and we are immediately done since the translated rule instance is simply the $(\multimap L)$ rule of LBI. The other rules are similar. Note that for the display rule (\equiv_D) we simply treat each display postulate individually: applications of display postulates either collapse under $\lceil - \rceil$ or boil down to the commutativity of the comma or semicolon, which is handled by the bunch equivalence \equiv . \square

Proposition 5.3 *Any cut-elimination procedure for DL_{BI} may be constructively transformed into a cut-elimination procedure for LBI, and vice versa.*

Proof. Immediate from Lemmas 5.1 and 5.2. \square

While Lemma 5.1 demonstrates that a cut-free LBI proof is essentially a cut-free DL_{BI} proof with some display steps omitted, Lemma 5.2 indicates the converse: any cut-free DL_{BI} proof can be viewed as a cut-free LBI proof by first bringing each consecution into a “display-normal form”. We suggest that this normal form probably does not exist for the display calculi for the other bunched logics in any meaningful sense (and so Lemma 5.2 does not adapt), because of the seemingly essential presence of the structural negations \sharp and/or \flat in these calculi. For example, if we consider the DL_{BBI} consecution $F, \sharp G \vdash H$, then it is clear that there is no display-rearrangement of this consecution which could be called “bunched-like” in

the sense that \sharp does not occur in it. Thus any cut-free sequent calculus for BBI *without* such a unary negative structuring must represent cut-free DL_{BBI} proofs in a rather non-trivial way, and it appears quite plausible that attempts to formulate such a calculus are fundamentally doomed — an observation borne out by our own experience and that of others [19]. Similar remarks apply to dMBI and CBI . (Of course, this does not rule out other, less syntax-directed approaches such as labelled deduction based on tableaux [11,16] or hybrid logics [22].)

6 Conclusion

Our main contribution in this paper is a unified proof theory for the principal varieties of bunched logic, based on display calculus (incidentally substantiating O’Hearn and Pym’s suggestion that this apparatus might apply to BI [18]). In particular, we provide the first cut-free proof system for BBI, which underlies separation and spatial logics employed in program analysis. Evidence for the canonicity of our unified proof theory is provided by cut-elimination for each of our calculi, as well as soundness and completeness with respect to the basic presentations of the corresponding logics.

The fact that each bunched logic can individually be presented as a display calculus is relatively unsurprising in the light of the earlier display calculus for CBI presented in [3], and the intuitionistic display technology, based on residual pairs of connectives, to be found in [13,23]. As well as realising these calculi explicitly, we make two additional contributions in this paper. First, we obtain our proof theory in a unified and economical way, by first formulating and then combining calculi for the elementary additive and multiplicative components of the bunched logics. Our treatment takes advantage of the compositionality of key structural properties of display calculi: given that the properties hold for two “elementary” display calculi, it is straightforward to establish that the same properties hold of the display calculus obtained by combining them. Second, having formulated our display calculi, we are in a position to immediately establish a translation between cut-free proofs in our display calculus for BI and those in its standard bunched sequent calculus. By doing so, we observe not only that this sequent calculus can be seen as an optimised display calculus, but also that the display calculi for the other bunched logics cannot be pared down to a sequent calculus in the same way. Both observations provide strong evidence that our formulation of the proof theory of bunched logics in terms of display calculi is indeed canonical.

Though complete cut-free proof systems for bunched logic are of clear theoretical interest, from the practical perspective it remains to be seen whether our proof theory will find application in automated theorem-proving tools. The need for such tools is quite real, e.g., in the setting of separation logic, which is based on BBI, but since both separation logic and BBI are fundamentally undecidable [4], compromises are clearly necessary⁶. We suggest that our work might be applied in two

⁶ In fact, separation logic is significantly more complicated than pure BBI, as it must also account for specific properties of the heap-like models on which it is based.

main directions. First, the display property intuitively corresponds to “pointing” or “focusing” in a proof attempt, where one selects part of a subgoal to work on. Thus our display calculi might well find application in semi-automated proof assistants, where the proof search is partially or wholly guided by humans. Second, it might be possible to obtain useful fully-automated but incomplete proof search tools by imposing constraints on the use of structural rules. A further possibility might be to look at obtaining *deep inference* calculi, which abandon the distinction between logical connectives and structural ones [5], for bunched logics by attempting to extract formula-rewriting rules from their cut-free display calculi. Our approach may also open new avenues for display-based proof theories for other logics with relevance for computer science.

References

- [1] Nuel D. Belnap, Jr. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.
- [2] Josh Berdine and Peter O’Hearn. Strong update, disposal and encapsulation in bunched typing. In *Proceedings of MFPS, ENTCS*. Elsevier, 2006.
- [3] James Brotherston and Cristiano Calcagno. Classical BI (A logic for reasoning about dualising resource). In *Proceedings of POPL-36*, pages 328–339, 2009.
- [4] James Brotherston and Max Kanovich. Undecidability of propositional separation logic and its neighbours. To appear in *Proceedings of LICS-25*, 2010.
- [5] Kai Brännler. Deep inference and its normal form of derivations. In *Proceedings of CiE*, volume 3988 of *LNCS*, pages 65–74, 2006.
- [6] Cristiano Calcagno, Philippa Gardner, and Uri Zarfaty. Context logic as modal logic: Completeness and parametric inexpressivity. In *Proceedings of POPL-34*, 2007.
- [7] Bor-Yuh Evan Chang and Xavier Rival. Relational inductive shape analysis. In *Proceedings of POPL-35*, 2008.
- [8] Wei-Ngan Chin, Cristina David, Huu Hai Nguyen, and Shengchao Qin. Enhancing modular OO verification with separation logic. In *Proceedings of POPL-35*, 2008.
- [9] Matthew Collinson, David Pym, and Edmund Robinson. Bunched polymorphism. *Mathematical Structures in Computer Science*, 18(6):1091–1132, 2008.
- [10] Dino Distefano and Matthew Parkinson. jStar: Towards practical verification for Java. In *Proceedings of OOPSLA*, pages 213–226. ACM, 2008.
- [11] D. Galmiche, D. Mery, and D. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, 15:1033–1088, 2005.
- [12] Didier Galmiche and Dominique Larchey-Wendling. Expressivity properties of Boolean BI through relational models. In *Proceedings of FSTTCS*, 2006.
- [13] Rajeev Goré. Gaggles, Gentzen and Galois: How to display your favourite substructural logic. *Logic Journal of the IGPL*, 6(5):669–694, 1998.
- [14] Samin Ishtiaq and Peter W. O’Hearn. BI as an assertion language for mutable data structures. In *Proceedings of POPL-28*, 2001.
- [15] Marcus Kracht. Power and weakness of the modal display calculus. In Heinrich Wansing, editor, *Proof Theory of Modal Logic*, pages 93–121. Kluwer Academic Publishers, 1996.
- [16] Dominique Larchey-Wendling and Didier Galmiche. Exploring the relation between intuitionistic BI and Boolean BI: An unexpected embedding. *Mathematical Structures in Computer Science*, 19:1–66, 2009.
- [17] Dominique Larchey-Wendling and Didier Galmiche. The undecidability of Boolean BI through phase semantics. To appear in *Proceedings of LICS-25*, 2010.

- [18] P.W. O’Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, June 1999.
- [19] David Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*. Applied Logic Series. Kluwer, 2002.
- [20] David Pym, Peter O’Hearn, and Hongseok Yang. Possible worlds and resources: The semantics of BI. *Theoretical Computer Science*, 315(1):257–305, 2004.
- [21] Stephen Read. *Relevant Logic: A Philosophical Examination*. Basil Blackwell, 1987.
- [22] Jason Reed. A Hybrid Logical Framework. PhD thesis, Carnegie Mellon University, 2009.
- [23] Greg Restall. Displaying and deciding substructural logics 1: Logics with contraposition. *Journal of Philosophical Logic*, 27:179–216, 1998.
- [24] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of 17th LICS*, 2002.