A Semantic Proof of Generalised Cut Elimination for Deep Inference[*]

Robert Atkey[a,1]   Wen Kokke[a,2]

[a] *Mathematically Structured Programming Group*
Computer and Information Sciences
University of Strathclyde
Glasgow, Scotland, UK

Abstract

Multiplicative-Additive System Virtual (MAV) is a logic that extends *Multiplicative-Additive Linear Logic* with a self-dual non-commutative operator expressing the concept of "before" or "sequencing". MAV is also an extenson of the the logic *Basic System Virtual* (BV) with additives. Formulas in BV have an appealing reading as processes with parallel and sequential composition. MAV adds internal and external choice operators. BV and MAV are also closely related to *Concurrent Kleene Algebras*.

Proof systems for MAV and BV are *Deep Inference* systems, which allow inference rules to be applied anywhere inside a structure. As with any proof system, a key question is whether proofs in MAV can be reduced to a normal form, removing detours and the introduction of structures not present in the original goal. In Sequent Calculi systems, this property is referred to as *Cut Elimination*. Deep Inference systems have an analogous Cut rule and other rules that are not present in normalised proofs. Cut Elimination for Deep Inference systems has the same metatheoretic benefits as for Sequent Calculi systems, including consistency and decidability.

Proofs of Cut Elimination for BV, MAV, and other Deep Inference systems present in the literature have relied on intricate syntactic reasoning and complex termination measures. For Linear Logic, Okada has pioneered semantic proofs of Cut Elimination, using Girard's Phase Space model of Linear Logic and techniques akin to Normalisation by Evaluation in λ-calculus, which avoid this intricate reasoning, but these have not been extended to the Deep Inference systems for BV and MAV.

In this work, we present a concise semantic proof that all MAV proofs can be reduced to a normal form avoiding the Cut rule and other "non analytic" rules. Due to the self-dual "before" connective, we cannot use Okada's Phase Space technique, which relies on closure under double negation. We build the model more directly using closed lower sets and the Chu construction. We also develop soundness and completeness proofs of MAV (and BV) with respect to a class of models. We have mechanised all our proofs in the Agda proof assistant, which provides both assurance of their correctness as well as yielding an executable normalisation procedure.

Keywords: Linear Logic, Deep Inference, Algebraic Semantics, Metatheory

1 Introduction

We present an algebraic semantics and semantic proof of generalised cut-elimination for the multiplicative-additive system MAV [22], which extends the basic system BV [3] [17,18] with the additives of multiplicative-additive linear logic [15, MALL]. The proof technique also extends to the additive units and the exponentials. Our proofs are constructive and mechanised in Agda [2].

1.1 BV, MAV, and Deep Inference

MAV and BV are *Deep Inference* systems. Deep Inference [19] is generalisation of Gentzen's methodology for designing proof systems, which arose from Guglielmi's attempts to relate process algebra [24,25, CCS] to Classical Linear Logic [15, CLL]. The problem with such a relation is that, while the multiplicative connectives of Linear Logic capture parallel composition, no connective of Linear Logic captures *sequential composition*. Eventually, Guglielmi's attempts yielded BV, which extends Multiplicative Linear Logic [15, MLL] with a self-dual non-commutative connective for sequential composition. Such a connective was already present in another extension of Linear Logic, Pomset logic [29], where it arose from the study of coherence space semantics of Linear Logic [16, Ch. 8]. Recently, Nguyễn and Straßburger [27,26] showed that, while BV is similar to Pomset logic, the two are not the same, as the theorems of BV form a proper subset of the theorems of Pomset logic. Neither BV nor Pomset logic has a sequent calculus. Tiu [33] showed that no sequent calculus can capture BV, and it is assumed that this result extends to Pomset logic.

Cut-elimination, or *admissibility of Cut*, is the fundamental property of Gentzen Sequent Calculi systems, which states that proofs using the Cut rule to introduce "detours" can be normalised to ones without. Crucial properties such as consistency and decidability follow from Cut-elimination. The Deep Inference analogue of Cut-elimination is the admissibility of the whole *up* fragment of the calculus, which includes the Deep Inference form of Cut (which we describe in Section 2 below) as well as duals of most of the other rules of the calculus. Admissibility of the *up* fragment has the same metatheoretic benefits for Deep Inference systems as it does for Sequent Calculus ones.

Guglielmi [19, Sec. 4.1] proves admissibility via the Splitting Theorem, which shows that proofs of conjoined structures can be split into separate subproofs. This is proved by a detailed syntactic analysis of proofs. Horne [22] gave a syntactic proof of the admissibility of the up fragment for MAV, that extends Guglielmi's technique with further reasoning about the additives. The proof is quite lengthy and involves intricate syntactic reasoning and the subtle and complex termination measures.

We present an alternative proof of cut-elimination via a semantic model. This proof avoids some of the intricacy of Horne's proof. We believe our proof to be more robust in the presence of extensions, due to our use of standard constructions such as Day monoids, order ideals, and the Chu construction. We demonstrate this by scaling down to plain BV and up to MAV with additive units, and also to BV with exponentials (Guglielmi and Straßburger's System NEL [31,20]).

The technique of demonstrating Cut Elimination by construction of a semantic model for MALL is due to Okada [28], who shows that the phase space model of MALL, described by Girard [15, Sec. 4.1] and Troelstra [34, Ch. 8], can be constructed from cut-free proofs. The completeness of this model directly yields the existence of a cut-free proof for every proof constructible in the MALL sequent calculus. The same technique was used by Abrusci [1] for non-commutative linear logic, by De, Jafarrahmani, and Saurin [13] for MALL with fixed points, and by Frumin [14] for Bunched Implications.

The phase semantic proof of cut-elimination does not easily extend to include the kind of self-dual connective present in BV and its extensions. The phase space model derives duality by means of double negation with respect to the monoidal structure, which means that any connective has a derived dual. Attempts to extend the model with the non-commutative connective result in two distinct but dual non-commutative connectives, reminiscent of the non-commutative tensor and par introduced by Slavnov [30].

---

[3] BV stands for <u>B</u>asic System <u>V</u>irtual, owing to an early interpretation of CCS interaction as the pairwise production and annihilation of virtual particles in physics [22, Fn. 2].

1.2 *Contribution and Content of this Paper*

To our knowledge, all prior work on the metatheory of Deep Inference systems like BV and MAV has been carried out using syntactic techniques such as rewriting with termination measures, or translations into other logics with known Cut-Elimination properties.

Our main contribution is the use of semantic techniques to derive the admissibility of identity expansion, cut, and the other *co-* rules of MAV. To this end, we have developed a number of results concerning the semantics of BV and MAV:

(i) In Section 3.2, we propose MAV-algebras as the algebraic counterpart of MAV. In short, an MAV-algebra is a *-autonomous partial order with meets, with another partially ordered monoid structure that is *duoidal* with respect to the *-autonomous structure.

(ii) Normal proofs (our name for the "up" fragment) as we define them in the next section do not *prima facie* support all the structure of an MAV-algebra, so we define the weaker notion of *MAV-frame* (Definition 3.13). These play the same role as Kripke structures do for modal and substructural logics. The first main technical contribution of the paper is in showing that every MAV-frame can be completed to an MAV-algebra.

(iii) We then show that MAV-frames are strongly complete (in the terminology of Okada) for MAV by proving that the MAV-frame constructed from normal proofs can be used to deduce that all MAV-provable structures have normal proofs in Theorem 4.2. Completeness in the usual sense also follows in Theorem 4.3.

We describe the MAV Deep Inference system in Section 2 and motivate the idea for readers not familiar with such systems.

As far as we are aware, the semantics of BV and its extensions have not been considered before in the literature. The crucial part of the proof is that the standard Chu construction extends to the self-dual non-commutative connective of BV and MAV (Proposition 3.43 and Proposition 3.47). Our development of the semantics of MAV and BV also opens the possibility of using these logics as sound and complete systems for reasoning about the MAV-frame structures we define.

All of our proofs have been mechanised and checked in the Agda proof assistant [2]. We briefly discuss the mechanisation in Section 5 and provide a hyperlinked guide to the proof relating it to our mathematical development in Appendix A. Aside from the benefits of checking the proof, the Agda proof is executable and yields a program for actually normalising proofs.

We present our proof in the full generality of MAV, but note that essentially the same proof applies to the subsystem BV as well. We have also extended the proof technique to include the additive units, and to BV with exponentials (System NEL, analysed by Guglielmi and Straßburger [31,20]) We discuss further extensions in Section 6.

2 The system MAV

In Deep Inference terminology, proofs operate on *structures*, which simultaneously play the role of formulas and sequents in traditional Sequent Calculus systems. There are a number of different notations in the literature for the structures of BV and related systems. For familiarity's sake, we opt for a notation similar to the formulas of normal Linear Logic, albeit extended with the self-dual non-commutative connective $\lhd$.

The structures of MAV are formed from positive and negative atoms (α and $\overline{\alpha}$) drawn from some set of atomic propositions, units (I), the non-commutative connective *seq* ($\lhd$), the multiplicative connectives *tensor* and *par* ($\otimes$ and $\parr$) and additive connectives *with* and *plus* ($\binampersand$ and $\oplus$).

$$P, Q, R, S ::= \alpha \mid \overline{\alpha} \mid I \mid P \lhd Q \mid P \otimes Q \mid P \parr Q \mid P \binampersand Q \mid P \oplus Q$$

Duality ($\overline{P}$) is an involutive function on structures that obeys the De Morgan laws for the multiplicative

and additive connectives and preserves the self-dual connective $\lhd$.

$$\overline{I} = I \qquad \overline{P \otimes Q} = \overline{P} \mathbin{\bindnasrepma} \overline{Q} \qquad \overline{P \mathbin{\bindnasrepma} Q} = \overline{P} \otimes \overline{Q}$$
$$\overline{\overline{\alpha}} = \alpha \qquad \overline{P \lhd Q} = \overline{P} \lhd \overline{Q} \qquad \overline{P \mathbin{\&} Q} = \overline{P} \oplus \overline{Q} \qquad \overline{P \oplus Q} = \overline{P} \mathbin{\&} \overline{Q}$$

Structures are considered equivalent modulo the equality $\simeq$, which is the smallest congruence defined by the associativity, commutativity, and identity laws that ensure that $(\lhd, I)$ forms a monoid, and $(\otimes, I)$ and $(\bindnasrepma, I)$ form commutative monoids.

$$P \lhd I \simeq P \qquad I \lhd P \simeq P \qquad P \lhd (Q \lhd R) \simeq (P \lhd Q) \lhd R$$
$$P \otimes I \simeq P \qquad P \otimes Q \simeq Q \otimes P \qquad P \otimes (Q \otimes R) \simeq (P \otimes Q) \otimes R$$
$$P \bindnasrepma I \simeq P \qquad P \bindnasrepma Q \simeq Q \bindnasrepma P \qquad P \bindnasrepma (Q \bindnasrepma R) \simeq (P \bindnasrepma Q) \bindnasrepma R$$

Structure contexts are one-hole contexts over structures. Plugging ($\mathcal{C}\{P\}$) replaces the hole in $\mathcal{C}$ with P.

$$\mathcal{C}, \mathcal{D} ::= \{\,\} \mid \mathcal{C} \lhd Q \mid P \lhd \mathcal{D} \mid \mathcal{C} \otimes Q \mid P \otimes \mathcal{D} \mid \mathcal{C} \bindnasrepma Q \mid P \bindnasrepma \mathcal{D} \mid \mathcal{C} \mathbin{\&} Q \mid P \mathbin{\&} \mathcal{D} \mid \mathcal{C} \oplus Q \mid P \oplus \mathcal{D}$$

The inference rules of MAV are presented as a *rewriting system* on structures. As this may be surprising to readers unfamiliar with deep inference, let us examine how this presentation relates to the usual presentation of linear logic. Rule (1), shown below, is the axiom rule in the usual one-sided presentation of linear logic. In the one-sided presentation, the turnstile is vestigial syntax, and can be removed. In BV, the $\bindnasrepma$ connective plays the same role as the comma does in the antecedent of a linear logic sequent, and the I plays the same role as the empty sequent, which would give us rule (2) for BV. However, BV's inference rules can work arbitrarily deep in the structure. (Hence, *deep* inference.) Hence, the axiom for BV is rule (3), where $\mathcal{C}$ is a one-hole structure context.

$$\frac{}{\vdash P, \overline{P}}\,(1) \qquad \frac{I}{P \bindnasrepma \overline{P}}\,(2) \qquad \frac{\mathcal{C}\{I\}}{\mathcal{C}\{P \bindnasrepma \overline{P}\}}\,(3)$$

Rule (4) is the cut rule in the usual one-sided presentation of linear logic. In rule (4), as in any branching inference rule, the branching enforces the *disjointness* of the premise derivations. In BV, disjointness is internalised by the $\otimes$ connective. Hence, it plays the same role as branching does in sequent calculus. This would give us rule (5) for BV. However, as BV's inference rules can work arbitrarily deep in the structure, and the system contains the (SWITCH) rule, the surrounding context of $\bindnasrepma$s is unnecessary—and too restrictive. Hence, the cut for BV is rule (6).

$$\frac{\vdash \Gamma, P, \Gamma' \quad \vdash \Delta, \overline{P}, \Delta'}{\vdash \Gamma, \Gamma', \Delta, \Delta'}\,(4) \qquad \frac{(\Gamma \bindnasrepma P \bindnasrepma \Gamma') \otimes (\Delta \bindnasrepma \overline{P} \bindnasrepma \Delta')}{\Gamma \bindnasrepma \Gamma' \bindnasrepma \Delta \bindnasrepma \Delta'}\,(5) \qquad \frac{\mathcal{C}\{P \otimes \overline{P}\}}{\mathcal{C}\{I\}}\,(6)$$

Beautifully, internalising branching makes the symmetry between the axiom and cut plain to see. In BV, to acknowledge this symmetry and the connection with CSS, the axiom and cut rules are referred to as *interaction* and *co-interaction*.

Proof trees are a cumbersome presentation for BV's derivations—they are convenient for branching sequent proofs, but BV derivations are sequences of structures. *Rewriting systems*, on the other hand, are a well-known and convenient notation for such derivations. Hence, in MAV, inference rules are presented as rewrite rules.

Inference, written $\longrightarrow$, is the smallest relation defined by the following axioms:

$P \mathbin{⅋} \overline{P}$	$\longrightarrow I$	INTERACT
$(P \otimes Q) \mathbin{⅋} R$	$\longrightarrow P \otimes (Q \mathbin{⅋} R)$	SWITCH
$I \mathbin{\&} I$	$\longrightarrow I$	TIDY
$(P \triangleleft Q) \mathbin{⅋} (R \triangleleft S)$	$\longrightarrow (P \mathbin{⅋} R) \triangleleft (Q \mathbin{⅋} S)$	SEQUENCE
$P \oplus Q$	$\longrightarrow P$	LEFT
$P \oplus Q$	$\longrightarrow Q$	RIGHT
$(P \mathbin{\&} Q) \mathbin{⅋} R$	$\longrightarrow (P \mathbin{⅋} R) \mathbin{\&} (Q \mathbin{⅋} R)$	EXTERNAL
$(P \triangleleft Q) \mathbin{\&} (R \triangleleft S)$	$\longrightarrow (P \mathbin{\&} R) \triangleleft (Q \mathbin{\&} S)$	MEDIAL
I	$\longrightarrow P \otimes \overline{P}$	COINTERACT
I	$\longrightarrow I \oplus I$	COTIDY
$(P \otimes R) \triangleleft (Q \otimes S)$	$\longrightarrow (P \triangleleft Q) \otimes (R \triangleleft S)$	COSEQUENCE
P	$\longrightarrow P \mathbin{\&} Q$	COLEFT
Q	$\longrightarrow P \mathbin{\&} Q$	CORIGHT
$(P \otimes R) \oplus (Q \otimes R)$	$\longrightarrow (P \oplus Q) \otimes R$	COEXTERNAL
$(P \oplus R) \triangleleft (Q \oplus S)$	$\longrightarrow (P \triangleleft Q) \oplus (R \triangleleft S)$	COMEDIAL
$\mathcal{C}\{P\} \longrightarrow \mathcal{C}\{Q\}$	if $P \longrightarrow Q$	MONO

If $P \longrightarrow Q$, we say that P can be inferred from Q, *i.e.* the arrow points from conclusion to premise.

N.B. $P \longrightarrow Q$ is an inference rule, *not a sequent*. In sequent calculus notation, it is $\frac{Q}{P}$, *not* $P \vdash Q$.

Derivation, written $\longrightarrow^{\star}$ is the reflexive, transitive closure of inference. *Invertible derivation*, written $\longleftrightarrow^{\star}$, is the symmetric core of derivation, *i.e.* $P \longleftrightarrow^{\star} Q = P \longrightarrow^{\star} Q \cap Q \longrightarrow^{\star} P$. *Proofs* are derivations that terminate in the unit, *e.g.* a derivation $P \longrightarrow^{\star} I$ is a proof of P.

The inference rules come in dual pairs. For every rule $P \longrightarrow Q$, there is a dual rule $\overline{Q} \longrightarrow \overline{P}$. The exception is (SWITCH), which is self-dual, up to commutativity. The (COINTERACT), (COLEFT), and (CORIGHT) rules introduce new structures going left-to-right. Normal proofs, which we define below, avoid these *synthetic* rules.

Remark 2.1 In BV, the structural connectives are usually presented as lists, distinguished only by their brackets: $P \otimes Q$ is written as $(P\ Q)$; $P \mathbin{⅋} Q$ is written as $[P\ Q]$; and $P \triangleleft Q$ is written as $\langle P\ Q \rangle$. Inferences, derivations, and proofs are presented vertically, as (7), (8), and (9), respectively.

$$\frac{Q}{P} \ (7) \qquad \begin{matrix} Q \\ \| \\ P \end{matrix} \ (8) \qquad \begin{matrix} \mathbb{I} \\ \| \\ P \end{matrix} \ (9)$$

The relation between the deductive system for BV and rewrite systems is well-known, *e.g.* by Kahramanogullari [23], who implements proof search for several deep inference systems in Maude [9]. Inferences rules are usually named with the combination of a letter and an up or down arrow, *e.g.* (INTERACT) and (COINTERACT) are i↓ and i↑, respectively. The exception are self-dual rules, which are named with a single letter, *e.g.* (SWITCH) is usually named s.

Definition 2.2 A derivation is *normal* when it does not use (CoInteract) nor any of the other (CoX) rules, and its uses of (Interact) are restricted to (AtomInteract), as defined by the following axiom:

$$\overline{\alpha} \,\invamp\, \alpha \qquad\qquad \longrightarrow I \qquad\qquad\qquad \Big| \quad \text{AtomInteract}$$

Normal derivations avoid the use of rules that introduce new structures into proofs, and so can be termed *analytic* in contrast to the need for the synthetic rules to synthesise new structures.

Our main result, Theorem 4.2, is that every structure that is provable in full MAV also has a normal proof. Therefore, the system with only analytic rules is complete for MAV provability. Horne [22] proves this result via a syntactic proof involving rewriting and termination measures. In the following sections, we construct a semantic proof that normal proofs are complete for MAV.

3 Semantic Models for MAV

To prove the normalisation property for all MAV proofs, we use a semantic technique that is akin to Okada's phase space method and to Normalisation by Evaluation (NbE) [6]. We construct a semantics of the whole proof system from the system of normal proofs. This semantics is constructed in such a way that after interpreting a proof, the (existence of) a normal form can be extracted (or *read back* or *reified* in NbE terminology) from the semantic proof.

To our knowledge, the semantics of MAV have not been previously studied in the literature. The rules are an extension of MALL, so an *MAV-algebra* will partly be a ∗-autonomous partial order (Definition 3.3) with meets and joins. The additional structure for ◁ satisfies the conditions of a duoidal category [3, Definition 6.1] (Definition 3.5). We show that MAV is sound for MAV-algebras in Theorem 3.11.

To build MAV-algebras from normal proofs, we define the weaker notion of an *MAV-frame* (Definition 3.13). We show that a combination of certain closed lower sets (Section 3.4.2) and the Chu construction (Section 3.4.3) construct an MAV-algebra from any MAV-frame. Much of these constructions are well-known, but we have new results on lifting the Day construction of monoids on closed lower sets and the preservation of duoidal relationships that are required for MAV.

3.1 Pomonoidal, ∗-autonomous, and Duoidal Structure on Partial Orders

The algebraic semantics of MAV is a collection of interacting monoids on a partial order. We collect here the basic definitions and kinds of interaction we will need.

Definition 3.1 A *partial order monoid (pomonoid)* $(\bullet, i)$ on a poset $(A, \leq)$ comprises a binary operator $\bullet : A \times A \to A$ that is monotone in both arguments and an element $i \in A$ such that the usual monoid laws hold. A *commutative pomonoid* is a pomonoid where additionally $x \bullet y = y \bullet x$.

Definition 3.2 A commutative pomonoid $(\bullet, i)$ on a poset $(A, \leq)$ is *residuated* if there is a function $\multimap : A \times A \to A$ such that $x \bullet y \leq z$ iff $x \leq y \multimap z$.

Linear logic adds a duality, or negation, to a commutative pomonoid structure. Semantically, duality with commutativity is captured in the definition of ∗-autonomous category, due to Barr [5]. For our purposes, we need the partial order analogue, also called a CL algebra by Troelstra [34].

Definition 3.3 A *∗-autonomous partial order* is a structure $(A, \leq, \otimes, I, \neg)$ where $(\otimes, I)$ is a pomonoid on $(A, \leq)$ and $\neg : A^{\mathrm{op}} \to A$ is an anti-monotone and involutive operator on A, together satisfying $x \otimes y \leq \neg z$ iff $x \leq \neg(y \otimes z)$. A ∗-autonomous partial order satisfies *mix* if $\neg I = I$.

Remark 3.4 The structure of a ∗-autonomous partial order has a number of immediate consequences, but we leave description of these until after the definition of MAV-algebra in Definition 3.7.

BV and MAV extend linear logic by adding a non-commutative pomonoid structure that interacts with the existing pomonoid via a kind of interchange law (the (Sequence) rule in the proof system). We

follow [3, Definition 6.1] generalising these to the case of monoids with differing units. Their terminology is of a category having duoidal structure. We find it useful to describe one pomonoid as being *duoidal over* another to emphasise the non-symmetric nature of the relationship, and by analogy with one monoid distributing over another.

Definition 3.5 A pomonoid $(\bullet, i)$ is *duoidal over* another pomonoid $(\triangleleft, j)$ on a partial order $(A, \leq)$ if the following inequalities hold:

(i) $(w \triangleleft x) \bullet (y \triangleleft z) \leq (w \bullet y) \triangleleft (x \bullet z)$

(ii) $j \bullet j \leq j$

(iii) $i \leq i \triangleleft i$

(iv) $i \leq j$

Remark 3.6 In the case when the two pomonoids share a common unit the last three conditions for a duoidal relationship are automatically satisfied. We can also ignore the existence of the units and just describe two binary operators as being duoidal. If $\bullet$ is a join, or $\triangleleft$ is a meet, then all the conditions for a duoidal relationship are automatically met.

3.2 MAV-algebras

We define MAV-algebras as the algebraic semantics of MAV. The definition is a direct translation of the rules of MAV into order-theoretic language, using the definitions we have seen so far.

Definition 3.7 An *MAV-algebra* is a structure $(A, \leq, \otimes, \triangleleft, I, \neg)$ with the following properties:

(i) $(A, \leq, \otimes, I, \neg)$ is $*$-autonomous and satisfies *mix*.

(ii) $(A, \leq, \triangleleft, I)$ is a pomonoid.

(iii) $\triangleleft$ is self dual: $\neg(x \triangleleft y) = (\neg x) \triangleleft (\neg y)$.

(iv) $(\otimes, I)$ is duoidal over $(\triangleleft, I)$.

(v) $(A, \leq)$ has binary meets, which we write as $x \mathbin{\&} y$.

Proposition 3.8 *Let $(A, \leq, \otimes, \triangleleft, I, \neg)$ be a MAV-algebra.*

(i) *There is another commutative pomonoid structure $(\bindnasrepma, I)$ on $(A, \leq)$, defined as $x \bindnasrepma y = \neg(\neg x \otimes \neg y)$.*

(ii) *$(\otimes, I)$ and $(\bindnasrepma, I)$ are linearly distributive [10]: $x \otimes (y \bindnasrepma z) \leq (x \otimes y) \bindnasrepma z$.*

(iii) *$(A, \leq)$ has binary joins, given by $x \oplus y = \neg(\neg x \mathbin{\&} \neg y)$.*

(iv) *$\oplus$ distributes over $\otimes$: $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$.*

(v) *$\mathbin{\&}$ distributes over $\bindnasrepma$: $(x \bindnasrepma z) \mathbin{\&} (y \bindnasrepma z) = (x \mathbin{\&} y) \bindnasrepma z$.*

(vi) *$\triangleleft$ is duoidal over $\bindnasrepma$: $(w \bindnasrepma x) \triangleleft (y \bindnasrepma z) \leq (w \triangleleft y) \bindnasrepma (x \triangleleft z)$.*

(vii) *$\triangleleft$ is duoidal over $\mathbin{\&}$: $(w \mathbin{\&} x) \triangleleft (y \mathbin{\&} z) \leq (w \triangleleft y) \mathbin{\&} (x \triangleleft z)$.*

(viii) *$\oplus$ is duoidal over $\triangleleft$: $(w \triangleleft x) \oplus (y \triangleleft z) \leq (x \oplus y) \triangleleft (x \oplus z)$.*

Definition 3.9 Let At be a set of atomic propositions. Given an MAV-algebra $(A, \leq, \otimes, \triangleleft, I, \neg)$ and valuation $V : At \to A$, define the interpretation of MAV Formulas as follows: $[\![I]\!] = I$, $[\![\alpha]\!] = V(\alpha)$, $[\![\overline{\alpha}]\!] = \neg V(\alpha)$, $[\![P \otimes Q]\!] = [\![P]\!] \otimes [\![Q]\!]$, $[\![P \bindnasrepma Q]\!] = [\![P]\!] \bindnasrepma [\![Q]\!]$, $[\![P \triangleleft Q]\!] = [\![P]\!] \triangleleft [\![Q]\!]$, $[\![P \mathbin{\&} Q]\!] = [\![P]\!] \mathbin{\&} [\![Q]\!]$, and $[\![P \oplus Q]\!] = [\![P]\!] \oplus [\![Q]\!]$.

Lemma 3.10 *For all P, $[\![\overline{P}]\!] = \neg[\![P]\!]$.*

Theorem 3.11 *The interpretation in Definition 3.9 is sound: for all structures P, if $P \longrightarrow^{\star} I$, then $I \leq [\![P]\!]$.*

Proof Each of the required inequalities has been established in Proposition 3.8. □

Remark 3.12 More generally, if $P \longrightarrow^\star Q$ in MAV, then $[\![Q]\!] \leq [\![P]\!]$ in an MAV-algebra. Note that the ordering is reversed! It will be reversed again in the definition of MAV-frame.

3.3 MAV-frames

To prove completeness of the normal proofs of MAV, we will construct a particular MAV-algebra from the structures and normal proofs. Since normal proofs do not *a priori* have all the necessary structure for an MAV-algebra, in the following sections we develop a procedure to construct MAV-algebra from the lighter requirements of an MAV-frame. In Section 4 we will show that the MAV-algebra constructed from the normal proof MAV-frame allows us to prove that all proofs in MAV can be normalised to normal proofs.

Definition 3.13 An *MAV-frame* is a structure $(F, \leq, \mathbin{⅋}, ◁, i, +)$ where $(F, \leq)$ is a partial order, $(F, \leq, \mathbin{⅋}, i)$ is a commutative pomonoid, $(F, \leq, ◁, i)$ is a pomonoid, $+$ is a binary monotone function on $(F, \leq)$, and these data satisfy the following inequalities:

(i) $(w ◁ x) \mathbin{⅋} (y ◁ z) \leq (w \mathbin{⅋} y) ◁ (x \mathbin{⅋} z)$

(ii) $(x + y) \mathbin{⅋} z \leq (x \mathbin{⅋} z) + (y \mathbin{⅋} z)$

(iii) $(w ◁ x) + (y ◁ z) \leq (w + y) ◁ (x + z)$

(iv) $i + i \leq i$

Remark 3.14 An MAV-frame is essentially two duoidal relationships and a distributivity law.

Remark 3.15 MAV-frames have a intuitive reading as CCS-like process algebras (see Milner [25] for an introduction to CCS). If we assume the existence of a collection of "action" elements $a \in F$ and their duals $\bar{a} \in F$, satisfying $a \mathbin{⅋} \bar{a} \leq i$, then we can read the constructs of an MAV-frame as parallel composition, sequential composition, and choice. The ordering is interpreted as a reduction relation. An interesting avenue for future work would be to discover to what extent MAV can be thought of as a logic for processes in this process algebra.

Remark 3.16 MAV-frames (and MAV-algebras) are also very similar to the definition of a *Concurrent Kleene Algebra* (CKA) due to Hoare, Möller, Struth and Wehrman [21, Definition 4.1]. One difference is that we do not assume that $+$ is a join, nor do we assume the existence of infinitary joins. Consequently, we have no analogue of the Kleene Star. Another difference is that the duoidal relationship is reversed in MAV-frames, indicating that MAV-frames capture evolution of processes while MAV-algebras and CKA capture properties of processes.

Proposition 3.17 *The* normal proof MAV-frame NMAV *is the partial order arising as the quotient of the preorder formed from the structures of MAV and $P \leq Q$ if there is a normal derivation $P \longrightarrow^\star Q$, defined as $(\mathcal{S}, \longrightarrow^\star, \mathbin{⅋}, ◁, I, \&)$, where $\mathcal{S}$ is the set of all MAV structures. The required (in)equalities follow directly from the definition of $\longrightarrow^\star$ for normal proofs.*

Remark 3.18 The construction of the MAV-frame NMAV does not use the $\otimes$ and $\oplus$ structure of MAV directly. This structure is recovered by duality from the other connectives by the constructions in the rest of this section. This corresponds to the fact that the Co-X rules in MAV that we wish to show admissible are the ones that mention the $\otimes$ and $\oplus$ connectives, with the exception of (Switch), which has a special role to play in Proposition 4.1 in mediating interaction.

3.4 Constructing MAV-algebras from MAV-frames

We construct MAV-algebras from MAV-frames in a three step process. In Section 3.4.1, we use lower sets and the Day construction to add meets, joins and residuals for pomonoids to a partial order. This construction creates joins freely, so we restrict to $+$-closed lower sets (*i.e.*, order ideals, but not necessarily over a $\vee$-semilattice) in Section 3.4.2 to turn the $+$ operation in MAV-frames into joins. Restricting to $+$-closed lower sets separates the Day construction of pomonoids into two separate cases, depending on how the original pomonoid interacts with $+$. Finally, we create the $*$-autonomous structure using the Chu construction in Section 3.4.3. The necessary duoidal structure is maintained through each construction.

3.4.1 Lower Sets and Day pomonoids

Definition 3.19 Given a partial order $(A, \leq)$, the set of lower sets $\widehat{A}$ consists of subsets $F \subseteq A$ that are down-closed: $x \in F$ and $y \leq x$ implies $y \in F$. Lower sets are ordered by inclusion. Define the embedding $\eta : A \to \widehat{A}$ as $\eta(x) = \{y \mid y \leq x\}$.

Proposition 3.20 *For any $(A, \leq)$, the function η is monotone, and $(\widehat{A}, \subseteq)$ has meets and joins given by intersection and union respectively.*

Proposition 3.21 *If $(\bullet, i)$ is a pomonoid on $(A, \leq)$, then there is a corresponding Day pomonoid $(\widehat{\bullet}, \widehat{i})$ on $\widehat{A}$ defined as $F \mathbin{\widehat{\bullet}} G = \{z \mid z \leq x \bullet y, x \in F, y \in G\}$ and $\widehat{i} = \eta(i)$. Moreover:*

 (i) *If $(\bullet, i)$ is a commutative pomonoid, then so is $(\widehat{\bullet}, \widehat{i})$.*

 (ii) *$(\widehat{\bullet}, \widehat{i})$ has left and right residuals, which coincide when it is commutative. We will only be interested in residuals for commutative pomonoids, which we write as $F \multimap G$.*

 (iii) *The embedding preserves the monoid: $\eta(x \bullet y) = \eta(x) \mathbin{\widehat{\bullet}} \eta(y)$.*

Remark 3.22 Proposition 3.21 is the Day monoidal product on functor categories [12] restricted to the case of partial orders and lower sets.

Remark 3.23 When $(A, \leq)$ is an MAV-frame, Proposition 3.21 gives us two pomonoids $(\widehat{\invamp}, \widehat{I})$ and $(\widehat{\vartriangleleft}, \widehat{I})$ on $\widehat{A}$. Moreover, the next proposition states that the duoidal relationship between these monoids is preserved by the Day construction:

Proposition 3.24 *If $(\bullet, i)$ is duoidal over $(\vartriangleleft, j)$ then $(\widehat{\bullet}, \widehat{i})$ is duoidal over $(\widehat{\vartriangleleft}, \widehat{j})$.*

3.4.2 +-closed Lower Sets

In the Phase Semantics, structures are interpreted as elements that are fixed points of a closure operator defined by a double negation with respect to the monoid on the original frame. This closure operator generates a partial order of "facts" whose meets and joins exactly correspond to the syntactic ones when the original monoid is derived from the proofs. As we mentioned in the introduction, the presence of a self-dual operator in BV means that we cannot use double negation closure, and we have to proceed more deliberately to preserve join-like structure in an MAV-frame when building MAV-algebras. We do this by defining +-closed lower sets as those that are closed under finite +-combinations of their elements. This leads to a closure operator on lower sets that allows us to immediately deduce that +-closed lower sets form a lattice. We also preserve the Day-pomonoids from lower sets, but in two different ways, depending on how the original pomonoid interacts with +. In Proposition 3.32 we handle pomonoids that distribute over +, and in Proposition 3.33 we handle pomonoids that are duoidal under +. We need these constructions to lift the $\invamp$ and $\vartriangleleft$ pomonoids from MAV-frames to +-closed lower sets. Finally in this section, we show that duoidal structure on lower sets from Proposition 3.24 is preserved in +-closed lower sets.

For this section, we assume that $(A, \leq)$ is a partial order with a monotone binary operation $+ : A \times A \to A$ (we do not assume that + is a join or even a pomonoid.)

Definition 3.25 A lower set $F \in \widehat{A}$ is *+-closed* if $x \in F$ and $y \in F$ imply $x + y \in F$. +-closed lower sets are ordered by subset inclusion and form a partial order $(\widehat{A}^+, \subseteq)$.

Proposition 3.26 *Let $U : \widehat{A}^+ \to \widehat{A}$ be the "forgetful" function that forgets the +-closed property. There is a monotone function $\alpha : \widehat{A} \to \widehat{A}^+$ such that for all $F \in \widehat{A}^+$, $\alpha(UF) = F$ and for all $F \in \widehat{A}$, $F \subseteq U(\alpha F)$.*

Proof To define α, we close lower sets under all +-combinations. To this end, for $F \in \widehat{A}$, define $\mathrm{ctxt}(F)$, the set of all +-combinations of F inductively built from constructors $\mathsf{leaf} : F \to \mathrm{ctxt}(F)$ and $\mathsf{node} : \mathrm{ctxt}(F) \times \mathrm{ctxt}(F) \to \mathrm{ctxt}(F)$. We define the *sum* of a context as $sum(\mathsf{leaf}\ x) = x$ and $sum(\mathsf{node}(c, d)) = sum(c) + sum(d)$. Now define: $\alpha(F) = \{x \mid c \in \mathrm{ctxt}(F), x \leq sum(c)\}$. This is +-closed, by taking the node combination of contexts. $\alpha \circ U$ is idempotent because α does not introduce any elements to lower sets that are already closed. For arbitrary lower sets F, $F \subseteq U(\alpha F)$ by the leaf constructor. $\square$

Definition 3.27 Define the embedding $\eta^+ : A \to \widehat{A}^+$ as $\eta^+(x) = \alpha(\eta(x))$.

Remark 3.28 By this proposition, $U \circ \alpha$ is a closure operator on $\widehat{A}$ [11], and the closed elements are those of $\widehat{A}^+$. The next proposition is standard for showing that meets and joins exist on the closed elements for some closure operator.

Proposition 3.29 $(\widehat{A}^+, \subseteq)$ *has all meets and joins. In the binary case, meets are defined by intersection and joins are defined by* $F \vee G = \alpha(UF \cup UG)$.

Proposition 3.30 $\eta^+(x + y) \subseteq \eta^+(x) \vee \eta^+(y)$.

Remark 3.31 Proposition 3.30 is the reason for requiring +-closure. This property will allow us to prove the crucial embedding property for all structures in Section 4.

Proposition 3.32 *For a commutative pomonoid* $(\bullet, i)$ *on* $(A, \leq)$ *that distributes over* + *(i.e.,* $(x+y) \bullet z \leq (x \bullet z) + (y \bullet z)$ *holds), we have that* $F \widehat{\bullet}^+ G = \alpha(UF \widehat{\bullet} UG)$ *and* $\widehat{j}^+ = \alpha(\widehat{j})$ *define a residuated commutative pomonoid on* $\widehat{A}^+$. *Moreover,* $\eta^+(x \bullet y) = \eta^+(x) \widehat{\bullet}^+ \eta^+(y)$.

Proof Define an operation $\bullet^c : \mathrm{ctxt}(F) \times \mathrm{ctxt}(G) \to \mathrm{ctxt}(F \widehat{\bullet} G)$ that "multiplies" two trees, such that $sum(c) \bullet sum(d) \leq sum(c \bullet^c d)$, using the distributivity. This allows us to show that α preserves the monoid operation: $\alpha F \widehat{\bullet}^+ \alpha F = \alpha(F \widehat{\bullet} G)$. With this, we can show that the monotonicity, associativity, unit, and commutativity properties of $\widehat{\bullet}$ transfer over to $\widehat{\bullet}^+$. The definition of the residual from lower sets is already +-closed, by distributivity. $\square$

Proposition 3.33 *For a pomonoid* $(\lhd, j)$ *on* $(A, \leq)$, *if this satisfies* $(w \lhd x) + (y \lhd z) \leq (w + y) \lhd (x + z)$ *then the Day construction* $F \widehat{\lhd} G = \{z \mid z \leq x \lhd y, x \in F, y \in G\}$ *on lower sets is +-closed when* F *and* G *are. We write* $F \widehat{\lhd}^+ G$ *to indicate when we mean this construction as an operation on +-closed lower sets. If* $j + j \leq j$, *then the Day unit* $\widehat{j} = \eta(j)$ *is also closed and we write it as* $\widehat{j}^+ \in \widehat{A}^+$. *Together* $(\widehat{\lhd}^+, \widehat{j}^+)$ *form a pomonoid on* $(\widehat{A}^+, \subseteq)$. *Moreover,* $\eta^+(x \lhd y) \leq \eta^+(x) \widehat{\lhd}^+ \eta^+(y)$.

Proof Since + is duoidal over $(\lhd, j)$, the Day monoid $\widehat{\lhd}$ is automatically +-closed by calculation. The monoid structure directly transfers. Similarly, $\eta(j)$ is automatically +-closed since $j + j \leq j$. $\square$

Remark 3.34 Generalising the situation for the unit j in Proposition 3.33, $\eta(x)$ is closed for any x such that $x + x \leq x$. Note that if + were a join on $(A, \leq)$, then this would automatically be satisfied.

Remark 3.35 We have used the same decoration $\widehat{\bullet}^+$ and $\widehat{\lhd}^+$ for two separate constructions of pomonoids on +-closed lower sets. We will be careful to distinguish which we mean: in our present application, a symmetric operator like $\bullet$ will distribute over + and so $\widehat{\bullet}^+$ will be constructed by Proposition 3.32; and a non-symmetric operator like $\lhd$ will be duoidal under + and so $\widehat{\lhd}^+$ will be constructed by Proposition 3.33.

Remark 3.36 If we have two pomonoids on $(A, \leq)$ that share a unit, then the two constructions of units in Propositions 3.32 and 3.33 will yield the same element of $\widehat{A}^+$.

Proposition 3.37 *If* $(\bullet, i)$ *is duoidal over* $(\lhd, j)$ *on* $(A, \leq)$, *and* $(\bullet, i)$ *distributes over* + *(as in Proposition 3.32) and* + *is duoidal over* $(\lhd, j)$ *(as in Proposition 3.33), then* $(\widehat{\bullet}^+, \widehat{i}^+)$ *is duoidal over* $(\widehat{\lhd}^+, \widehat{j}^+)$ *on* $(\widehat{A}^+, \subseteq)$.

Proof The duoidal relationship established in Proposition 3.24 carries over thanks to the properties of α and U. The fact that $\widehat{i}^+ \subseteq \widehat{j}^+$ relies on the condition $j + j \leq j$ to collapse +-contexts of js. $\square$

3.4.3 Chu Construction
To construct suitable MAV-algebras, we use the partial order version of the Chu construction [5, Appendix by Po-Hsiang Chu]. The Chu construction builds $*$-autonomous categories from symmetric monoidal closed categories with pullbacks. In the partial order case, the requirement for pullbacks simplifies to binary meets. For this section, we let $(A, \leq, \bullet, i, \multimap)$ be a partial order with a residuated pomonoid structure and all binary meets.

Definition 3.38 Let k be an element of A. $\mathrm{Chu}(A, k)$ is the partial order with elements pairs (a^+, a^-) such that $a^+ \bullet a^- \leq k$, with ordering $(a^+, a^-) \sqsubseteq (b^+, b^-)$ when $a^+ \leq b^+$ and $b^- \leq a^-$. There is a monotone embedding function $\eta^c : A \to \mathrm{Chu}(A, k)$ defined as $\eta^c(x) = (x, x \multimap k)$.

Proposition 3.39 $(\mathrm{Chu}(A, k), \sqsubseteq)$ *has a* $*$*-autonomous structure defined as:*

$$(a^+, a^-) \otimes (b^+, b^-) = (a^+ \bullet b^+, (b^+ \multimap a^-) \wedge (a^+ \multimap b^-)) \qquad I = (i, k) \qquad \neg(a^+, a^-) = (a^-, a^+)$$

Moreover, $\eta^c(x \bullet y) = \eta^c(x) \otimes \eta^c(y)$ *and* $\eta^c(i) = I$.

Remark 3.40 If we choose $k = i$, then $(\mathrm{Chu}(A, i), \sqsubseteq)$ has $*$-autonomous structure that satisfies *mix*.

Proposition 3.41 *If* A *has binary joins, then* $(\mathrm{Chu}(A, k), \sqsubseteq)$ *has binary meets, given by* (a^+, a^-) & $(b^+, b^-) = (a^+ \wedge b^+, a^- \vee b^-)$.

Remark 3.42 Since $(\mathrm{Chu}(A, k), \sqsubseteq, \otimes, I, \neg)$ is a $*$-autonomous partial order, then Proposition 3.41 also means that $\mathrm{Chu}(A, k)$ has all binary joins, with $(a^+, a^-) \oplus (b^+, b^-) = (a^+ \vee b^+, a^- \wedge b^-)$.

We now turn to the self-dual duoidal structure required to interpret the $\lhd$ connective. First we transfer pomonoids from $(A, \leq)$ to self-dual pomonoids on $(\mathrm{Chu}(A, k), \sqsubseteq)$ provided they interact well with k:

Proposition 3.43 *Let* $(\lhd, j)$ *be a pomonoid on* $(A, \leq)$ *such that* $(\bullet, i)$ *is duoidal over* $k \lhd k \leq k$ *and* $j \leq k$, *then* $x \lhd y = (x^+ \lhd y^+, x^- \lhd y^-)$ *and* $J = (j, j)$ *form a self-dual pomonoid on* $\mathrm{Chu}(A, k)$.

Proof $x \lhd y$ is well-defined because $(x^+ \lhd y^+) \bullet (x^- \lhd y^-) \leq (x^+ \bullet x^-) \lhd (y^+ \bullet y^-) \leq k \lhd k \leq k$. J is well defined because $j \bullet j \leq j \leq k$. The pomonoid laws all transfer directly. $\square$

Remark 3.44 When $k = j$, the two conditions in the proposition are automatically satisfied. Moreover, if $k = i = j$, then not only does the $*$-autonomous structure satisfy *mix*, but we also have $I = J$.

Finally, we need to show that if $(\bullet, j)$ is duoidal over $(\lhd, j)$, then their Chu counterparts are in the same relationship. Due to the use of residuals in the definition of $\otimes$, we need the following fact about duoidal residuated pomonoids:

Lemma 3.45 *If* $(\bullet, j)$ *is duoidal over* $(\lhd, j)$ *in a partial order* $(A, \leq)$ *and* $(\bullet, j)$ *has a residual* $\multimap$, *then* $(w \multimap x) \lhd (y \multimap z) \leq (w \lhd y) \multimap (x \lhd z)$.

Remark 3.46 Lemma 3.45 is in some sense the "intuitionistic" version of the duoidal relationship for $\parr$ arising as the dual of that for $\otimes$ in a $*$-autonomous partial order, as we saw in Proposition 3.8.

Proposition 3.47 *If* $(\bullet, i)$ *is duoidal over* $(\lhd, j)$ *on* $(A, \leq)$, *and* $(\lhd, j)$ *satisfies the conditions of Proposition 3.43, then* $(\otimes, I)$ *and* $(\lhd, J)$ *are in a duoidal relationship on* $\mathrm{Chu}(A, k)$.

Proof For the positive half of the Chu construction, this is a direct consequence of the duoidal relationship. For the negative half, we use Lemma 3.45 and the fact that meets are always duoidal. $\square$

3.4.4 Construction of MAV-algebras from MAV-frames

The propositions in the preceeding three sections together prove that every MAV-frame yields an MAV-algebra:

Theorem 3.48 *If* $(F, \leq, \parr, \lhd, i, +)$ *is an MAV-frame, then* $(\mathrm{Chu}(\widehat{F^+}, \widehat{i^+}), \sqsubseteq)$ *has the structure of an MAV-algebra.*

With this theorem we can define a notion of validity in MAV in terms of truth in all MAV-frame generated algebras. By Theorem 3.11, MAV is sound for this notion of validity:

Theorem 3.49 *MAV is sound for the MAV-frame semantics: if* $P \longrightarrow^\star I$ *then for all MAV-frames* F, $I \sqsubseteq \llbracket P \rrbracket$ *in* $(\mathrm{Chu}(\widehat{F^+}, \widehat{i^+}), \sqsubseteq)$.

4 Semantic Cut-Elimination and Proof Normalisation

Let $\mathrm{Chu}(\widehat{\mathrm{NMav}}^+, \widehat{I}^+)$ be the MAV-algebra constructed (Theorem 3.48) from the normal proof MAV-frame (Proposition 3.17), where elements are positive/negative pairs of $+$-closed lower sets of structures. We define the valuation of atoms as $V(\alpha) = \eta^c(\eta^+(\overline{\alpha}))$. By Theorem 3.11, we have an interpretation of MAV structures $[\![P]\!]$ such that if $P \longrightarrow^\star I$, then $I \sqsubseteq [\![P]\!]$. We now prove our main proposition about this interpretation in the MAV-algebra derived from the MAV-frame of normal proofs NMav that will allow us to derive the admissibility of all the non-normal proof rules of MAV.

Proposition 4.1 *For all structures P, $[\![P]\!] \sqsubseteq \neg(\eta^c(\eta^+(P)))$.*

Proof By Definition 3.38, this statement comprises two inclusions between pairs of $+$-closed lower sets:

(i) $\eta^+(P) \subseteq [\![P]\!]^-$

(ii) $[\![P]\!]^+ \subseteq \eta^+(P) \multimapdotinv^+ \widehat{I}^+$

We prove the second assuming the first. It suffices to prove that $[\![P]\!]^+ \mathbin{\widehat{\bullet}^+} \eta^+(P) \subseteq \widehat{I}^+$, which follows from the first part and the property of all Chu-elements that $[\![P]\!]^+ \mathbin{\widehat{\bullet}^+} [\![P]\!]^- \subseteq \widehat{I}^+$.

We prove the first part by induction on P. In the cases when $P = I$ or $P = \overline{\alpha}$, we already have $[\![P]\!]^- = \eta^+(P)$. When $P = \alpha$, we have $[\![\alpha]\!]^- = \eta^+(\overline{\alpha}) \multimapdotinv^+ \widehat{I}^+$. It suffices to prove that $\eta^+(\alpha) \mathbin{\widehat{\bullet}^+} \eta^+(\overline{\alpha}) \subseteq \widehat{I}^+$, which follows from the preservation of monoid operations by η^+ and the (ATOMINTERACT) rule.

When $P = Q \mathbin{\rotatebox[origin=c]{180}{\&}} R$, $Q \mathbin{\&} R$, or $Q \mathbin{\lhd} R$, the result follows from preservation of the corresponding monoid structure by η^+. For example, $\eta^+(Q \mathbin{\rotatebox[origin=c]{180}{\&}} R) \subseteq \eta^+(Q) \mathbin{\widehat{\bullet}^+} \eta^+(R) \subseteq [\![Q]\!]^- \mathbin{\widehat{\bullet}^+} [\![R]\!]^- = [\![Q \mathbin{\rotatebox[origin=c]{180}{\&}} R]\!]^-$.

When $P = Q \oplus R$, we have $\eta^+(Q \oplus R) \subseteq \eta^+(Q)$ and $\eta^+(Q \oplus R) \subseteq \eta^+(R)$, by the (LEFT) and (RIGHT) rules. Therefore, $\eta^+(Q \oplus R) \subseteq \eta^+(Q) \vee \eta^+(R) \subseteq [\![Q]\!]^- \vee [\![R]\!]^- = [\![Q \oplus R]\!]^-$.

When $P = Q \otimes R$, we have $[\![Q \otimes R]\!]^- = ([\![Q]\!]^+ \multimapdotinv [\![R]\!]^-) \wedge ([\![R]\!]^+ \multimapdotinv [\![Q]\!]^-)$. We prove inclusion in the left-hand side, the right-hand side is similar. The key property we need to prove is:

$$\eta^+(Q \otimes R) \mathbin{\widehat{\bullet}^+} (\eta^+(Q) \multimapdotinv^+ \widehat{I}^+) \subseteq \eta^+(R) \tag{10}$$

Using the monoidality and monotonicity of α, this inclusion is implied by the following inclusion in $\widehat{\mathrm{NMav}}$:

$$\eta(Q \otimes R) \mathbin{\widehat{\bullet}} (U(\eta^+(Q)) \multimapdotinv \widehat{I}) \subseteq \eta(R)$$

which follows from the (SWITCH) rule of MAV and calculation. Using 10, and inclusion (ii) above, we can prove the inequality we need:

$$\eta^+(Q \otimes R) \mathbin{\widehat{\bullet}^+} [\![Q]\!]^+ \subseteq \eta^+(Q \otimes R) \mathbin{\widehat{\bullet}^+} (\eta^+(Q) \multimapdotinv^+ \widehat{I}^+) \subseteq \eta^+(R) \subseteq [\![R]\!]^-$$

Using the residuation property of $\widehat{\bullet}^+$ we can conclude. □

Theorem 4.2 *If $P \longrightarrow^\star I$ in MAV, then there is a normal proof $P \longrightarrow^\star I$.*

Proof By Theorem 3.11, $P \longrightarrow^\star I$ in MAV implies $I \sqsubseteq [\![P]\!]$. Combined with Proposition 4.1, we have $I \sqsubseteq \neg\eta^c(\eta(P))$. By Definition 3.38 of the ordering of Chu elements, we have $\eta^+(P) \subseteq \widehat{I}^+$. Since $P \in \eta^+(P)$, we have $P \in \widehat{I}^+$, which by definition means that there is a normal proof $P \longrightarrow^\star I$. □

Another consequence of Proposition 4.1 is that MAV is complete for the MAV-frame semantics:

Theorem 4.3 *MAV is complete for the MAV-frame semantics: if, for all MAV-frames F, $[\![I]\!] \sqsubseteq [\![P]\!]$ in $(\mathrm{Chu}(\widehat{F}^+, \widehat{I}^+), \sqsubseteq)$, then $P \longrightarrow^\star I$.*

5 Mechanisation in Agda

We formalised the proofs in the paper in Agda [2]. The source code is available at the following URL:

https://github.com/bobatkey/semantic-cut-elimination

Furthermore, a hyperlinked HTML rendition of the source can be browsed at the following URL:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024

In Appendix A, we provide a guide to the mechanisation relating the Definitions, Propositions, and Theorems in the previous sections to the Agda definitions in the mechanisation. The formalisation uses setoids to represent sets, and reuses definitions from the Agda Standard Library [32] where appropriate.

We did not attempt to formalise Horne's syntactic proof of generalised cut-elimination directly. We suspect that this would likely be quite involved, due to the widespread and implicit use of syntactic equalities when manipulating structures, as well as the construction of the relevant termination measures. We found that the semantic constructions were relatively straightforward to formalise in Agda.

In addition to increasing the confidence in our results, a key benefit of the formalised proof in a proof assistant for constructive proof such as Agda is that the proof normalisation procedure defined by Theorem 4.2 is executable. As an example, we have normalised the one-step proof below:

$$((I \oplus I) \lhd (I \binampersand I)) \bindnasrepma ((I \binampersand I) \lhd (I \oplus I)) \xrightarrow{\text{(Interact)}} I$$

The proof normalises to a 38-step normal proof, of which 9 are inference steps, and the remainder are (sometimes spurious) equalities. The example can be found at the following URL:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024/MAV.Example.html

6 Extensions and Future Work

We have presented a semantic proof of generalised Cut elimination for the Multiplicative-Additive System Virtual (MAV), which reduces Horne's approximately 41 page proof to a 7 page proof.

Our proof technique is modular, and can be adapted with relative ease to a variety of related systems. To evidence this claim, we have adapted our Agda formalisation to prove generalised cut elimination to the following systems:

BV The basic system does not have the additives (*i.e.* $\binampersand$ and $\oplus$). The proof is a straightforward restriction of our proof for MAV, but only relies on lower sets, rather than +-closed lower sets. The source code of the proof is available at the following URL:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024/BV.CutElim.html

MAUV The multiplicative-additive-unital system adds the additive units (*i.e.* $\top$ and $\mathbf{0}$, using Girard's notation). The proof is a straightforward extension of our proof for MAV, and requires the use of lower sets which are 0-closed as well as +-closed. The source code of the proof is available at the following URL:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024/MAUV.CutElim.html

NEL Non-commutative exponential logic [20] extends BV with the exponentials (*i.e.* ! and ?, using Girard's notation). The proof is a straightforward extension of our proof for MAV, and requires that we: (i) add *near-exponentials*, which do not satisfy monoidality, to BV-frames; (ii) construct exponentials on lower sets, which adds monoidality; (iii) construct exponentials on Chu spaces, which adds duality; and (iv) extend the main proposition (Proposition 4.1) to account for the new connectives, which requires lemmas similar to those for $\otimes$ and $\bindnasrepma$. The source code of the proof is available at the following URL:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024/NEL.CutElim.html

The work opens up several paths for future work. The theory, developed here, for lifting Day pomonoids to +-closed lower sets enables alternative Cut-elimination proofs for other substructural logics, such as MALL and Bunched Implications. (Okada's technique has already been applied to Bunched Implications by Frumin [14].) We find the technique of using +-closed lower sets more revealing in how the join structure is preserved than the use of impredicative closure operators in Frumin's proof or double negation closure in Okada's.

We plan to investigate extensions of MAV with exponentials, as in the system NEL, and with a Kleene Star operator, which can be seen as the exponential for the ⊲ connective. Adding a Kleene Star would tighten the connection with Concurrent Kleene Algebras we highlighted in Remark 3.16. It would interesting to see to what extent MAV can be seen as a logic for processes represented as elements of MAV-frames. Furthermore, we plan to investigate fixpoint operators following Baelde [4] and De, Jafarrahmani and Saurin [13]. The latters' use of Okada's technique is not compatible with Agda's type theory, as it relies on impredicativity to construct fixpoints with the double negation closure. We believe that our more direct predicative technique will be able to use Agda's inductive and coinductive types.

Lastly, we plan to extend our semantics of BV and MAV to a categorical semantics that considers equalities between proofs as well as provability. Such a semantics ought to be useful for treating MAV as a session-typed language, as considered by Ciobanu and Horne [8]. The necessary analogue of MAV-algebras has already been investigated by Blute, Panangaden, and Slavnov [7] as BV-categories, which are Aguiar and Mahajan's 2-monoidal (or duoidal) categories [3] extended with duality. The key task will be to categorify the constructions in this paper to show how the categorical analogue of MAV-frames induces MAV-categories.

Acknowledgement

References

[1] Abrusci, V. M., *Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic*, The Journal of Symbolic Logic **56**, pages 1403–1451 (1991), ISSN 00224812.
http://www.jstor.org/stable/2275485

[2] Agda Developers, *Agda*.
https://agda.readthedocs.io/en/v2.6.4/

[3] Aguiar, M. and S. Mahajan, *Monoidal Functors, Species and Hopf Algebras*, American Mathematical Society (2010), ISBN 9781470417680.
https://doi.org/10.1090/crmm/029

[4] Baelde, D., *Least and greatest fixed points in linear logic*, ACM Trans. Comput. Log. **13**, pages 2:1–2:44 (2012).
https://doi.org/10.1145/2071368.2071370

[5] Barr, M., **-Autonomous Categories*, Springer Berlin Heidelberg (1979), ISBN 9783540348504.
https://doi.org/10.1007/bfb0064579

[6] Berger, U., M. Eberl and H. Schwichtenberg, *Normalisation by Evaluation*, in: B. Möller and J. V. Tucker, editors, *Prospects for Hardware Foundations, ESPRIT Working Group 8533, NADA - New Hardware Design Methods, Survey Chapters*, volume 1546 of *Lecture Notes in Computer Science*, pages 117–137, Springer (1998).
https://doi.org/10.1007/3-540-49254-2_4

[7] Blute, R., P. Panangaden and S. Slavnov, *Deep inference and probabilistic coherence spaces*, Applied Categorical Structures **20**, page 209–228 (2010), ISSN 1572-9095.
https://doi.org/10.1007/s10485-010-9241-0

[8] Ciobanu, G. and R. Horne, *Behavioural Analysis of Sessions Using the Calculus of Structures*, page 91–106, Springer International Publishing (2016), ISBN 9783319415796.
https://doi.org/10.1007/978-3-319-41579-6_8

[9] Clavel, M., F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer and J. Quesada, *Maude: specification and programming in rewriting logic*, Theoretical Computer Science **285**, pages 187–243 (2002), ISSN 0304-3975. Rewriting Logic and its Applications.
https://doi.org/https://doi.org/10.1016/S0304-3975(01)00359-0

[10] Cockett, J. and R. Seely, *Linearly distributive functors*, Journal of Pure and Applied Algebra **143**, page 155–203 (1999), ISSN 0022-4049.
https://doi.org/10.1016/s0022-4049(98)00110-8

[11] Davey, B. A. and H. A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press (2002), ISBN 9780511809088.
https://doi.org/10.1017/cbo9780511809088

[12] Day, B., *On closed categories of functors*, page 1–38, Springer Berlin Heidelberg (1970), ISBN 9783540362920.
https://doi.org/10.1007/bfb0060438

[13] De, A., F. Jafarrahmani and A. Saurin, *Phase Semantics for Linear Logic with Least and Greatest Fixed Points*, in: A. Dawar and V. Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*, volume 250 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:23, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022), ISBN 978-3-95977-261-7, ISSN 1868-8969.
https://doi.org/10.4230/LIPIcs.FSTTCS.2022.35

[14] Frumin, D., *Semantic cut elimination for the logic of bunched implications, formalized in Coq*, in: *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2022, page 291–306, Association for Computing Machinery, New York, NY, USA (2022), ISBN 9781450391825.
https://doi.org/10.1145/3497775.3503690

[15] Girard, J., *Linear logic*, Theor. Comput. Sci. **50**, pages 1–102 (1987).
https://doi.org/10.1016/0304-3975(87)90045-4

[16] Girard, J.-Y., P. Taylor and Y. Lafont, *Proofs and Types*, Cambridge University Press, USA (1989), ISBN 0521371813.

[17] Guglielmi, A., *A calculus of order and interaction*, Technical report, Knowledge Representation and Reasoning Group, Department of Computer Science, Dresden University of Technology (1999). The technical report is no longer available. The author states it was "thoroughly rewritten" as A System of Interaction and Structure [18].

[18] Guglielmi, A., *A system of interaction and structure*, ACM Trans. Comput. Logic **8**, page 1–es (2007), ISSN 1529-3785.
https://doi.org/10.1145/1182613.1182614

[19] Guglielmi, A., *Deep inference*, in: B. W. Paleo and D. Delahaye, editors, *All about Proofs, Proofs for All*, College Publications (2015), ISBN 978-1-84890-166-7.

[20] Guglielmi, A. and L. Straßburger, *A system of interaction and structure V: the exponentials and splitting*, Math. Struct. Comput. Sci. **21**, pages 563–584 (2011).
https://doi.org/10.1017/S096012951100003X

[21] Hoare, T., B. Möller, G. Struth and I. Wehrman, *Concurrent kleene algebra and its foundations*, The Journal of Logic and Algebraic Programming **80**, page 266–296 (2011), ISSN 1567-8326.
https://doi.org/10.1016/j.jlap.2011.04.005

[22] Horne, R., *The consistency and complexity of multiplicative additive system virtual*, Scientific Annals of Computer Science **25**, pages 245–316 (2015).
https://doi.org/10.7561/SACS.2015.2.245

[23] Kahramanoğulları, O., *Maude as a platform for designing and implementing deep inference systems*, Electron. Notes Theor. Comput. Sci. **219**, page 35–50 (2008), ISSN 1571-0661.
https://doi.org/10.1016/j.entcs.2008.10.033

[24] Milner, R., *A Calculus of Communicating Systems*, Lecture Notes in Computer Science, Springer Berlin Heidelberg (1980), ISBN 978-3-540-10235-9.
https://doi.org/10.1007/3-540-10235-3

[25] Milner, R., *Communication & Concurrency*, Prentice Hall, Philadelphia, PA (1989).

[26] Nguyễn, L. T. D. and L. Straßburger, *A System of Interaction and Structure III: The Complexity of BV and Pomset Logic*, Logical Methods in Computer Science **Volume 19, Issue 4** (2023).
https://doi.org/10.46298/lmcs-19(4:25)2023

[27] Nguyễn, L. T. D. u. T. and L. Straßburger, *BV and Pomset Logic Are Not the Same*, in: F. Manea and A. Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*, volume 216 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:17, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022), ISBN 978-3-95977-218-1, ISSN 1868-8969.
https://doi.org/10.4230/LIPIcs.CSL.2022.32

[28] Okada, M., *Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic*, Theoretical Computer Science **227**, pages 333–396 (1999), ISSN 0304-3975.
https://doi.org/https://doi.org/10.1016/S0304-3975(99)00058-4

[29] Retoré, C., *Pomset logic: A non-commutative extension of classical linear logic*, in: *Lecture Notes in Computer Science*, pages 300–318, Springer Berlin Heidelberg (1997).
https://doi.org/10.1007/3-540-62688-3_43

[30] Slavnov, S., *On noncommutative extensions of linear logic*, Logical Methods in Computer Science **Volume 15, Issue 3** (2019).
https://doi.org/10.23638/LMCS-15(3:30)2019

[31] StraßBurger, L. and A. Guglielmi, *A system of interaction and structure IV: The exponentials and decomposition*, ACM Transactions on Computational Logic **12**, page 1–39 (2011), ISSN 1557-945X.
https://doi.org/10.1145/1970398.1970399

[32] The Agda Community, *Agda Standard Library* (2023).
https://github.com/agda/agda-stdlib

[33] Tiu, A., *A System of Interaction and Structure II: The Need for Deep Inference*, Logical Methods in Computer Science **Volume 2, Issue 2** (2006).
https://doi.org/10.2168/LMCS-2(2:4)2006

[34] Troelstra, A. S., *Lectures on Linear Logic*, Center for the Study of Language and Information, Stanford, CA (1992).

A   Statements and corresponding Agda definitions

The table below lists the main definitions, propositions, and theorems in the paper together with the qualified name of the corresponding Agda definition. The names of the Agda definitions are hyperlinked to the HTML rendition of the source code, which is available at the following URL:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024

If the version of the paper you are reading does not support hyperlinks, you can reconstruct the URL to any definition by taking the URL above, adding a **/**, the module path followed by **.html**, and the definition name prefixed with a **#**. For instance, the URL to the definition of `` `¬_ `` is:

https://bobatkey.github.io/semantic-cut-elimination/MFPS/2024/MAV.Structure.html#`¬_

The table has three columns. The first gives an intuitive name to the definition, proposition, or theorem statement. The second links to its statement in the paper, if applicable. The third gives the name of the corresponding Agda definition, and links to its statement online.

Table A.1: Statements and corresponding Agda definitions.

Section 2		
Structures	P, Q, R, S	MAV.Structure.Structure
Duality	$\overline{P}$	MAV.Structure.`¬_
Symmetric MAV		
Equality	$\simeq$	MAV.Symmetric._≃_
Inference	$\longrightarrow$	MAV.Symmetric._⟶_
Derivation	$\longrightarrow^{\star}$	MAV.Symmetric._⟶★_

Invertible derivation	$\longleftrightarrow^\star$	`MAV.Symmetric._`$\longleftrightarrow\star$`_`
Normal MAV	Definition 2.2	
Equality	$\simeq$	`MAV.Base._`$\simeq$`_`
Inference	$\longrightarrow$	`MAV.Base._`$\longrightarrow$`_`
Derivation	$\longrightarrow^\star$	`MAV.Base._`$\longrightarrow\star$`_`
Invertible derivation	$\longleftrightarrow^\star$	`MAV.Base._`$\longleftrightarrow\star$`_`

Section 3.1

Pomonoids	Definition 3.1	`IsPomonoid`
if Commutative	Definition 3.1	`IsCommutativePomonoid`
if Residuated	Definition 3.2	`IsResiduatedCommutativePomonoid`
if Duoidal	Definition 3.5	`IsDuoidal`
∗-autonomous partial order	Definition 3.3	`IsStarAutonomous`

(The preceding definitions can be found under `Algebra.Orderd.Structures`.)

Section 3.2

MAV-algebra	Definition 3.7	`MAV.Model.Model`
Interpretation	Definition 3.9	`MAV.Interpretation.`⟦`_`⟧
Soundness	Theorem 3.11	
for Equality		`MAV.Interpretation.`⟦`_`⟧`eq`
for Inference		`MAV.Interpretation.`⟦`_`⟧`step`
for Derivation		`MAV.Interpretation.`⟦`_`⟧`steps`

Section 3.3

MAV-frame	Definition 3.13	`MAV.Frame.Frame`
Normal MAV-frame	Proposition 3.17	`MAV.Base.frame`

Section 3.4.1

Lower sets	Definition 3.19	`LowerSet`
Day pomonoid	Proposition 3.21	module `Day`
if Commutative	Proposition 3.21	module `DayCommutative`
if Duoidal	Proposition 3.24	module `DayDuoidal`

(The preceding definitions can be found under `Algebra.Ordered.Construction.LowerSet`.)

Section 3.4.2

+-closed lower sets	Definition 3.25	`Ideal`
+-closed Day pomonoid		
if Commutative	Proposition 3.32	module `DayCommutative`
if Duoidal	Proposition 3.33	module `DayDuoidal`

(The preceding definitions can be found under `Algebra.Ordered.Construction.Ideal`.)

Section 3.4.3		
Chu construction	Definition 3.38	`Construction.Chu`
for duoidal pomonoids	Proposition 3.47	module `Construction.SelfDual`

(The preceding definitions can be found under `Algebra.Ordered.Construction.Chu`.)

Section 3.4.4		
MAV-algebras from MAV-frames	Theorem 3.48	`MAV.Frame.FrameModel.model`

Section 4		
Main lemma	Proposition 4.1	`MAV.CutElim.main-lemma`
Generalised cut-elimination	Theorem 4.2	`MAV.CutElim.cut-elim`