Gaggles, Gentzen and Galois: How to display your favourite substructural logic.

Rajeev Goré, *Automated Reasoning Project*[1] *and Department of Computer Science, Australian National University, Canberra, ACT, 0200, Australia,* rpg@arp.anu.edu.au

Abstract

We show how to obtain cut-free Display Calculi for algebraic logics characterised by the Gaggle Theory of Dunn. These Display Calculi automatically inherit the Kripke-style relational semantics associated with gaggles thereby completing a unified, proof-theoretic, algebraic and model-theoretic picture for these logics.

Keywords: display logic, gaggle theory, proof theory, substructural logics

1 Introduction

Everyone is familiar with the method of solving algebraic equations like $2x^2 - 4 \leq 0$ by manipulating the equation to "make x the subject" as follows:

the equation	$(2 \times x^2) - 4 \leq 0$	
becomes	$(2 \times x^2) \leq 0 + 4$	by adding 4 to both sides
becomes	$x^2 \leq (0+4)/2$	by dividing both sides by 2
becomes	$x \leq \sqrt{(0+4)/2}$	by taking (positive) square root of both sides
becomes	$x \leq \sqrt{2}$	by simplification (or rewriting).

Most of these manipulations are only possible because the operations come in "opposite" (residuated or Galois connected) pairs that undo the effects of each other; namely $(+,-)$, $(\times,/)$, $(x^2,\sqrt{x})$. The ability to "display the x" by "making x the subject" allows us to unravel the context surrounding the x, thereby shedding some light on the meaning of x in the given context. Once we have a handle on x, we can replace all occurrences of x in other equations by the right hand side, thereby reducing the number of variables in the problem at hand.

In this paper we show how to construct proof systems with such a "display property" for a plethora of logics with numerous applications in theoretical computer science, theoretical linguistics, and computational linguistics. In most of these applications, the essence of the task is to decide whether some given statement A follows from a collection of some statements Γ using the rules of one of these logics. An important challenge for computer scientists is to automate this proof search procedure, particularly because in many of these logics, this task is known to be undecidable.

As has been argued elsewhere [27, 39] a particularly good logical formalism for

[1]Work partially supported by a visiting fellowship from the Swiss National Science Foundation to the Institute for Applied Mathematics and Computer Science, University of Bern, Switzerland, and the Australian Research Council via a Queen Elizabeth II Fellowship.

proof search is the method of Gentzen [37]. But Gentzen systems typically contain (some variation of) the cut rule, which says that "if Γ entails A, and A entails Δ, then Γ entails Δ", thereby allowing us to compose two proofs to build a third. In backward proof search, however, this rule is notoriously bad since it forces us to guess some arbitrary cut-formula A which may have no syntactic similarity with the given Γ or Δ, introducing non-determinism and thereby frustrating the search for a proof.

Consequently, only Gentzen systems in which the cut-rule is redundant, or in which all uses of the cut rule are "analytic" [7], are suited for backward proof search. We concentrate on the former, and such Gentzen systems obey the "cut-elimination" principle that "if there is a proof of $\Gamma \vdash \Delta$, then there is a proof of $\Gamma \vdash \Delta$ in which the cut-rule is not used". Much of the literature on proof theory is devoted to proving "cut-elimination" for specific Gentzen systems, and these proofs are typically extremely complicated.

In contrast, Belnap's Display Logic [4] is a general Gentzen-style framework which specifies eight easily checked conditions on rules. The beauty of Display Logic is a general cut-elimination theorem which states that any collection of rules obeying these conditions automatically enjoys cut-elimination. Furthermore, in keeping with the algebraic connections with logic, the ability to "make x the subject" is a fundamental principle of Display Logic. Indeed, the name comes from this very principle.

In his original paper, Belnap [4] showed how to capture many different logics in Display Logic purely by the addition or deletion of only structural rules. But subsequent research in display logic has shown that Belnap's original framework could be greatly improved [40, 25, 19, 36, 22]. Here we generalise all these calculi.

Certain rules in Gentzen's original calculus involved no logical connectives; they merely swapped the order of formulae in Γ, or affected the number of copies of the same formula in Γ, or allowed the addition of extra formulae to Γ. Since these rules affected only the structure of the sequent they were called structural rules. Over the last sixty years, Gentzen's original calculus has been refined to build in the effect of these structural rules, thereby making them implicit. But logicians have also investigated the effects of adding these structural rules piecemeal; one at a time, or in certain combinations. By doing so they discovered a plethora of "substructural logics" [10], each characterised by its own collection of structural rules.

The modern study of substructural logics has taken on a computational flavour since these logics can also be viewed from many perspectives relevant to computer science. For example:

- "resource sensitive" substructural logics can be used to model situations where, in a proof of $\Gamma \vdash A$, each resource from Γ is used at least once [1], at most once [31], and even exactly once [16].
- the Lambek Calculus [26] and its extensions are used in computational linguistics to model the mathematical structure of (English) sentences [29].
- some substructural logics have been reinvented in the study of fuzzy set theory [30, 23, 33].
- Action Logic [34] and Channel Theory [3], which are attempts to formalise a "theory of actions", can also be seen as substructural logics.

It is well known that most of these substructural logics have algebraic, relational, categorial and topological semantics [10]. In a series of papers, J Michael Dunn has given a general theory, called Gaggle Theory, showing how to obtain uniform relational semantics for substructural logics by generalising the algebraic notions of residuation and Galois connections [12, 13, 14]. Since residuation and Galois connections are at the heart of the algebraic manipulations which allowed us to "make x the subject", an open question for many years has been to find the exact connections between Dunn's Gaggle Theory and Belnap's Display Logic.

Here we show how to obtain a *cut-free* display calculus from some given partial gaggle. By choosing the gaggle-theoretic description that is appropriate for your favourite substructural logic, this allows you to automatically obtain a cut-free display calculus for that logic; see also [22]. Furthermore, this display calculus is *guaranteed* to be sound and complete with respect to the relational semantics given by gaggle theory. Since the "intended semantics" are often the main reason for inventing a logic, this is a powerful way to invent Gentzen systems from desired semantic criteria.

From an automated theorem proving perspective, this work paves the way to extend the "connection method" to substructural logics [6, 2, 5, 39]. There is still much work to be done, for though Display Calculi obey a "subformula property", they do not obey a "substructure property". Thus Display Calculi are not immediately suitable for *naive* backward proof search. But Dawson [8] has recently shown that an existing Display Calculus for relation algebras [20] is amenable to mechanisation using Paulson's generic *proof assistant* Isabelle [32].

The rest of this paper is organised as follows. In Section 2 we explain the basic algebraic notions that are used later. In Section 3 we explain, informally, all of the steps involved in the methodology by concentrating on a particular example. In Section 4 we present the method for obtaining sequent rules which engender the display property and show how to obtain a display calculus from a gaggle-theoretic description. In Section 5 we mention further work and in Section 6 we conclude. Belnap's conditions can be found in Section 7.

Acknowledgements: I am grateful to Nuel Belnap, J Michael Dunn, Hiroakira Ono and Giovanni Sambin for encouragement. Hiroakira Ono also informed me of the connections between fuzzy set theory and substructural logics. Thanks also to Greg Restall (whose paper [35] also broached some of these issues) for many useful discussions.

2 Basic Notions

The following definitions are a slight modification of those from Dunn [14]; Dunn's definitions do not quite work out since he sometimes forgets to multiply the signs of a "trace" by the output sign in order to obtain the "tonicity", explained shortly. These definitions have appeared already in [22] but we repeat them for completeness.

Let $(A, \leq, \perp, \top)$ be a lattice with least and greatest elements $\top$ and $\perp$, respectively. Let f be an n-ary function on this lattice. The function f is **isotonic in the j-th position** if for all $a, b \in A$,

$$(a \leq b \Rightarrow f(a_1, \cdots, a_{j-1}, a, a_{j+1}, \cdots, a_n) \leq f(a_1, \cdots, a_{j-1}, b, a_{j+1}, \cdots, a_n))$$

and is **antitonic in the j-th position** if for all $a, b \in A$,

$$(a \leq b \Rightarrow f(a_1, \cdots, a_{j-1}, b, a_{j+1}, \cdots, a_n) \leq f(a_1, \cdots, a_{j-1}, a, a_{j+1}, \cdots, a_n)).$$

function	tonicity	trace
$\otimes$	$(+,+)$	$(-,-)\mapsto -$
$\rightarrow$	$(-,+)$	$(-,+)\mapsto +$
$\leftarrow$	$(+,-)$	$(+,-)\mapsto +$
$\oplus$	$(+,+)$	$(+,+)\mapsto +$
$\succ$	$(-,+)$	$(+,-)\mapsto -$
$\prec$	$(+,-)$	$(-,+)\mapsto -$

FIG. 1. Tonicities and Traces of Various "functions"

Let $\pm$ stand for one of $+$ or $-$, let $+- = -+ = -$, let $++ = -- = +$, and let $\mathrm{tn}(f,j,+)$ and $\mathrm{tn}(f,j,-)$ stand for the fact that the function f is isotonic or antitonic in the j-th position respectively.

Then

$$f : (\pm_1, \pm_2, \ldots, \pm_j, \ldots, \pm_n) \mapsto \pm_{n+1}$$

is a **trace** for the function f iff $(\pm_j, \pm_{n+1})$ is:

$$\begin{cases} (+,+) & \text{if} \quad \mathrm{tn}(f,j,+) \quad \text{and} \quad (a_j = \top \Rightarrow f(a_1,\ldots,a_j,\ldots,a_n) = \top); \\ (-,-) & \text{if} \quad \mathrm{tn}(f,j,+) \quad \text{and} \quad (a_j = \bot \Rightarrow f(a_1,\ldots,a_j,\ldots,a_n) = \bot); \\ (-,+) & \text{if} \quad \mathrm{tn}(f,j,-) \quad \text{and} \quad (a_j = \bot \Rightarrow f(a_1,\ldots,a_j,\ldots,a_n) = \top); \\ (+,-) & \text{if} \quad \mathrm{tn}(f,j,-) \quad \text{and} \quad (a_j = \top \Rightarrow f(a_1,\ldots,a_j,\ldots,a_n) = \bot). \end{cases}$$

The reader should check that the function f has **tonicity** $\mathrm{tn}(f,j,\pm_{n+1}\pm_j)$. For example, the "functions" that interest us have the traces and tonicities shown in Figure 1.

A **tonoid** [14] is a structure $\mathcal{T} := (A, \leq, OP)$ where

1. $(A, \leq)$ is a non-empty partially ordered set; and
2. OP is a collection of n-ary functions, each having a trace (so that each function respects the bounds in at least one way).

Following Dunn [14]

1. The n-ary function g is a **contrapositive** (with respect to the j-th position) of another n-ary function f when:

$$\begin{array}{rl} \text{if} & f : (\pm_1, \pm_2, \ldots, \pm_j, \ldots, \pm_n) \mapsto \pm_{n+1} \\ \text{then} & g : (\pm_1, \pm_2, \ldots, -\pm_{n+1}, \ldots, \pm_n) \mapsto -\pm_j\,. \end{array}$$

2. If $\pm_{n+1} = -$, we write $S(f, a_1, a_2, \ldots, a_n, b)$ for $f(a_1,\ldots,a_n) \leq b$.
   If $\pm_{n+1} = +$, we write $S(f, a_1, a_2, \ldots, a_n, b)$ for $b \leq f(a_1,\ldots,a_n)$.
3. Two functions f and g satisfy the **Abstract Law Of Residuation** (in their j-th place) when:
   (i) f and g are contrapositives (with respect to their j-th place); and

(ii) $S(f, a_1, a_2, \ldots, a_j, \ldots, a_n, b)$ iff $S(g, a_1, a_2, \ldots, b, \ldots, a_n, a_j)$.

4. Two functions $f, g \in OP$ are **relatives** when they satisfy the Abstract Law Of Residuation in some position.
5. The family of operations OP is **founded** when there is a distinguished operation $f \in OP$, the **head**, such that any other operation $g \in OP$ is a relative of f.

Finally, a **partial gaggle** is a tonoid $\mathcal{T} := (A, \leq, OP)$ in which OP is a founded family.

Example 2.1 *For example, the structure $\mathcal{T} := (A, \leq, \{\otimes, \leftarrow, \rightarrow\})$ where all operations are binary and have traces*

$$\otimes : (-,-) \mapsto (-) \qquad \leftarrow : (+,-) \mapsto (+) \qquad \rightarrow : (-,+) \mapsto (+)$$

is a tonoid. Furthermore, if these operations satisfy

$$a \otimes b \leq c \quad \textit{iff} \quad a \leq c \leftarrow b \quad \textit{iff} \quad b \leq a \rightarrow c$$

for every $a, b, c \in A$, then they satisfy the Abstract Law Of Residuation and this structure is a partial gaggle with head $\otimes$.

The above example is the traditional picture of operations familiar from Lambek's original "sentential calculus" [26]. Dunn's Gaggle Theory [14] is an abstraction of these notions to the n-ary case. The power of Gaggle Theory is a general soundness and completeness result for logics characterised by gaggles in terms of Kripke-style relational semantics for these logical operations, as stated next.

Given a tonoid $\mathcal{T} := (A, \leq, OP)$, a **frame** for $\mathcal{T}$ [14] is a structure $(U, \sqsubseteq, \langle R_i \rangle_{i \in I})$ where U is a non-empty set (of points, worlds or setups), $\sqsubseteq$ is a partial order over U, and each R_i is a relation such that there is a 1-1 correspondence between OP and I, and for each $f \in OP$, if the degree of f is n, then the corresponding relation $R_i \subseteq U^{n+1}$. Thus each n-ary operation is associated with an $n+1$-ary relation [24].

Theorem 2.2 (Dunn [14]) *Every partial gaggle $\mathcal{T} := (A, \leq, OP)$ can be represented using a frame $(U, \sqsubseteq, \langle R_i \rangle_{i \in I})$ for $\mathcal{T}$.*

Our aim is to show that partial gaggles can be captured by a cut-free display calculus.

2.1 Display Logic

Display Logic [4] is a generalisation of Gentzen's sequent calculi which incorporates a multitude of complex structural connectives instead of Gentzen's single structural comma connective ",". The basic idea is that instead of building sequents from comma-separated sets, multisets or list of formulae (data), we should consider sequents built from formulae using arbitrary structural connectives. This idea of using extra structural connectives to capture different ways of "bunching data" now appears in many places, but its origins lie in the independent work of Mints [28] and Dunn [11]. As we shall see, the beauty of Display Logic is a general cut-elimination theorem. It is impossible to summarise Display Logic here so we refer the reader to the following extant works where Belnap's original ideas have been extended and simplified in various ways [40, 25, 19, 36, 22]. We now show how to obtain a Display Calculus for an arbitrary (fully-founded) partial gaggle.

3 Displaying Gaggle Theory: A Concrete Example

In this section, most of which is taken directly from [17], we show the interactions between all the concepts with which we deal, but in the context of a concrete example. The treatment is totally informal, so we abuse notation considerably. In particular, we use a mixture of algebraic and sequent notation, sometimes interchangeably.

3.1 Tonicity

Using the notation from [22], suppose we are given a binary function r_1 with a tonicity vector $\mathrm{tn}(r_1, -, +)$. The tonicity of r_1 can be specified in two *equivalent* ways using "rules". The first is to use the explicit pair of rules shown below in the top row. The second is to use one tonicity rule as shown below in the second row.

$$(\mathrm{tn}(r_1, 2, -)) \ \frac{a \leq b}{r_1(c, b) \leq r_1(c, a)} \qquad\qquad (\mathrm{tn}(r_1, 1, +)) \ \frac{a \leq b}{r_1(a, c) \leq r_1(b, c)}$$

$$(\mathrm{tn}(r_1, +, -)) \ \frac{a \leq c \quad d \leq b}{r_1(a, b) \leq r_1(c, d)}$$

Equivalent because we can derive each from the others as shown below:

$$\dfrac{\dfrac{a \leq c}{r_1(a, b) \leq r_1(c, b)}\,\mathrm{tn}(r_1, 1, +) \qquad \dfrac{d \leq b}{r_1(c, b) \leq r_1(c, d)}\,\mathrm{tn}(r_1, 2, -)}{r_1(a, b) \leq r_1(c, d)}\,(\mathrm{cut})$$

$\mathrm{tn}(r_1, +, -)$ from $(\mathrm{tn}(r_1, 1, +))$ and $(\mathrm{tn}(r_1, 2, -))$

$$\frac{a \leq b \qquad c \leq c}{r_1(a, c) \leq r_1(b, c)}\,\mathrm{tn}(r_1, +, -) \qquad\qquad \frac{c \leq c \qquad b \leq a}{r_1(c, a) \leq r_1(c, b)}\,\mathrm{tn}(r_1, +, -)$$

$(\mathrm{tn}(r_1, 1, +))$ from $(\mathrm{tn}(r_1, +, -))$ $\qquad$ $(\mathrm{tn}(r_1, 2, -))$ from $(\mathrm{tn}(r_1, +, -))$

Notice that the derivations in the second row rely on identity only, whereas the derivations in the first row depend on cut only. Thus, although equivalent, the method using $\mathrm{tn}(r_1, +, -)$ gives us one degree of freedom whilst the method using $\mathrm{tn}(r_1, 1, +)$ and $\mathrm{tn}(r_1, 2, -)$ gives us another. Which is to be preferred depends on our particular biases, and in our case, we are biased against the method requiring cut. That is, we will always prefer the approach using $\mathrm{tn}(r_1, +, -)$ in preference to the approach using $\mathrm{tn}(r_1, 1, +)$ and $\mathrm{tn}(r_1, 2, -)$.

Similar arguments can be made for analogous rules for a second connective r_2 with tonicity vector $\mathrm{tn}(r_2, -, +)$ and a third connective s with tonicity vector $\mathrm{tn}(s, +, +)$.

3.2 Residuation

The collection of "rules" shown below left involve three binary functions s, r_1 and r_2. The objects at the right are tonicity conditions for the three functions together with pairs of axiomatic rules that tell us something about the nesting of these functions.

$$\dfrac{\dfrac{x \leq r_1(z,y)}{\overline{\overline{s(x,y) \leq z}}}}{\overline{\overline{y \leq r_2(x,z)}}}$$

$$\begin{array}{cc} a \leq r_1(s(a,b),b) & s(r_1(a,b),b) \leq a \\ a \leq r_2(b,s(b,a)) & s(b,r_2(b,a)) \leq a \\ a \leq r_2(r_1(b,a),b) & a \leq r_1(b,r_2(a,b)) \end{array}$$

$$\mathrm{tn}(r_1,+,-) \quad \mathrm{tn}(s,+,+) \quad \mathrm{tn}(r_2,-,+)$$

The functions s and r_1 form a *dual residuated pair in the first position*; the functions s and r_2 form a *residuated pair in the second position*; and the functions r_1 and r_2 form a *Galois connection in the first and second positions*. The rules, or the tonicity vectors together with the axioms, are equivalent definitions of these concepts [12].

3.3 Partial Gaggles

Consider the partial gaggle $\mathcal{T} := (A, \leq, OP)$ where $OP = \{\otimes, \leftarrow, \rightarrow\}$, with head $\otimes$, and traces shown as below left:

$$\begin{array}{lll} \otimes: & (-,-) \mapsto & - \\ \leftarrow: & (+,-) \mapsto & + \\ \rightarrow: & (-,+) \mapsto & + \end{array}$$

$$\dfrac{\dfrac{a \leq c \leftarrow b}{\overline{\overline{a \otimes b \leq c}}}}{\overline{\overline{b \leq a \rightarrow c}}}$$

By definition of a partial gaggle, these operations satisfy the Abstract Law of Residuation [14, 22] in the appropriate places as shown above right.

3.4 Logical Connectives and Gentzen

Suppose that we now wish to Gentzenize[2] the "logic" obtained from the partial gaggle $\mathcal{T} := (A, \leq, OP)$ shown above by positing a "logical" connective for each operation from OP.

One way is to use our knowledge of "rules" to capture the necessary tonicities as outlined in Section 3.1 and Section 3.2 respectively. That is, we could posit introduction rules like those shown below left, and posit further axioms like those shown below right, where we introduce sequent notation by using $\vdash$ instead of $\leq$:

Method 1

$$(\mathrm{tn}(\leftarrow,+,-)) \quad \frac{A \vdash C \quad D \vdash B}{A \leftarrow B \vdash C \leftarrow D} \qquad\qquad A \vdash (A \otimes B) \leftarrow B \quad (A \leftarrow B) \otimes B \vdash A$$

But there are two reasons why this is unsatisfactory for Gentzenization. The first is that the rule shown above left "introduces" the connective $\leftarrow$ into both the left hand and right hand sides of the sequent at once, breaking one of the desiderata for good Gentzen systems [40]. The second is that the extra axioms, shown above right, go against the Gentzen philosophy since, to be useful, they require the use of cut.

A second way is to take the elements common to Section 3.2 and Section 3.3, by using the alternative method of capturing the necessary residuation conditions, namely via a set of rules exactly mimicking the residuation conditions:

[2] Even though some would say that this is a sin.

Method 2

$$\frac{A \vdash C \leftarrow A}{\overline{A \otimes B \vdash C}} \qquad\qquad \frac{A \otimes B \vdash C}{\overline{B \vdash A \rightarrow B}}$$

Such rules are undesirable since they break the subformula property in a drastic way.

A third way is to use parts of both of these methods by separating the syntax into a structural part (using structure variables X, Y, Z) that obeys explicit residuation, and a logical part (using formula variables A, B, C) that obeys the required tonicity rules. The separate parts form a hierarchy since every formula is also a structure. The structural part consists of three binary structural connectives s, r_1, and r_2, say, which are forced to obey the "display postulates" shown below:

Method 3

$$\frac{\dfrac{X \vdash r_1(Z, Y)}{\overline{s(X, Y) \vdash Z}}}{\overline{Y \vdash r_2(X, Z)}}$$

We immediately know that the connectives s, r_1 and r_2 have the properties we desire for $\otimes$, $\leftarrow$ and $\rightarrow$ respectively. The task now is to bequeath these properties onto $\otimes$, $\leftarrow$ and $\rightarrow$ in an acceptable way.

One way is to make s, r_1 and r_2 into *exact* proxies for $\otimes$, $\leftarrow$ and $\rightarrow$ respectively. One half of this map can be obtained via the following "rewriting" rules where the logical connective is introduced directly from a structural connective:

$$(\otimes\vdash)\ \frac{s(A, B) \vdash Z}{A \otimes B \vdash Z} \qquad (\vdash\rightarrow)\ \frac{Y \vdash r_2(A, C)}{Y \vdash A \rightarrow C} \qquad (\vdash\leftarrow)\ \frac{X \vdash r_1(C, B)}{X \vdash C \leftarrow B}$$

The remaining half of these mappings will come from the inverted version of these new rules. But if we simply postulate these rules as invertible rules, then we are back in the same situation as with Method 2, so this is not desirable. Is there another way to ensure that these rules are invertible ?

If we simply try to derive the inverted forms by using the above introduction rules, we obtain the invertibility conditions that we will need, namely:

$$\frac{\dfrac{A \vdash A \qquad B \vdash B}{s(A, B) \vdash A \otimes B}\,(\mathrm{tn}(s/\otimes, +, +)) \qquad A \otimes B \vdash Z}{s(A, B) \vdash Z}\,(\mathrm{cut})$$

$$\frac{Y \vdash A \rightarrow C \qquad \dfrac{A \vdash A \qquad C \vdash C}{A \rightarrow C \vdash r_2(A, C)}\,(\mathrm{tn}(r_2/\rightarrow, -, +))}{Y \vdash r_2(A, C)}\,(\mathrm{cut})$$

$$\frac{X \vdash C \leftarrow B \qquad \dfrac{B \vdash B \qquad C \vdash C}{C \leftarrow B \vdash r_1(C, B)}\,(\mathrm{tn}(r_1/\leftarrow, +, -))}{X \vdash r_1(C, B)}\,(\mathrm{cut})$$

The notation "$(\mathrm{tn}(s/\otimes, +, +))$" in the above is a deliberate attempt to show that these "invertibility conditions" are not rigorous tonicity conditions since on one side we have a structural connective, and on the other a logical connective. But we know from the "rewriting" rules that the given structural connective is already a proxy for the given logical connective on the side of turnstile in question.

Thus, the "invertibility conditions" *explicitly* complete the mapping between s, r_1, r_2 and $\otimes$, $\leftarrow$ and $\rightarrow$. That is, s, r_1, r_2 would be exact proxies for $\otimes$, $\leftarrow$ and $\rightarrow$ if we could push these derivations through.

Recall that the original tonicity rules were couched in terms of the functions r_1, s and r_2. But because of the "rewriting" rules, we can use the tonicity rules shown below instead, which when read upwards, "decode" for the structural connectives:

$$(\vdash \otimes)\ \frac{X \vdash C \quad Y \vdash D}{s(X,Y) \vdash C \otimes D} \qquad (\leftarrow\vdash)\ \frac{A \vdash Z \quad W \vdash B}{A \leftarrow B \vdash r_1(Z,W)} \qquad (\rightarrow\vdash)\ \frac{Z \vdash A \quad B \vdash W}{A \rightarrow B \vdash r_2(Z,W)}$$

That is, these "decoding" rules (or invertibility conditions) merely (re)state the tonicity requirements for $\otimes$, $\leftarrow$ and $\rightarrow$ as $(+,+)$, $(+,-)$, $(-,+)$ respectively using the mixed syntax of structural and logical connectives, but always expressed in terms of $\mathrm{tn}(f, \pm_1, \pm_2)$ rather than $\mathrm{tn}(f, i, +)$ or $\mathrm{tn}(f, i, -)$ from Section 3.1.

Not only do the invertibility condition derivations shown above now become proofs, but notice that for each logical connective, we have managed to derive one introduction rule from the other introduction rule using invertibility considerations and tonicity arguments.

3.5 Overloading a la Gentzen

The Gentzen system obtained so far works only if the structural connectives in question fall into their "natural" side. For example, a backward proof attempt of the endsequent

$$r_2((A \otimes B), (B \otimes A)) \vdash X$$

will flounder immediately since our display postulates from Method 3 of Section 3.4 do not allow us to expose the arguments of r_i [respectively s] when r_i [s] is the outermost connective on the left [right] hand side. Although this may block proofs of arbitrary structures, it does not block proofs of formulae, as we show next.

Suppose we attempt a backward proof of the endsequent $B \vdash C$. Read it upwards, starting from the end-sequent up to the leaves. The endsequent contains only formulae, so our current display postulates of Method 3 from Section 3.4 are not applicable. The only way for a structure to appear in the proof, read upwards, is via one of the "rewrite" introduction rules for the logical connectives (read upwards). If the connective is $\otimes$ [respectively $\rightarrow$, $\leftarrow$] then it turns into an s [r_i] on the left [right] hand side. The display postulates allow us to convert this occurrence of s [r_i] into an occurrence of r_1 or r_2 [s or $r_{j \neq i}$] on the right [left or right] hand side. But in every such move, all outermost occurrences of s [r_i] are always on the left [right] hand side.

That is, if the endsequent $B \vdash C$ is built from **formula**, using the language of $\otimes$, $\leftarrow$ and $\rightarrow$, then in any proof of this endsequent, the structure s [respectively $r_i, i = 1, 2$] appears as the *outermost connective*, only on the left [respectively right] hand side of the turnstile.

Thus, there is room in this setup for a set of "dual" display postulates and introduction rules using the same (or different) structural connectives, but introducing a different set of logical connectives. Adding these dual rules has no effect on proofs of endsequents like $B \vdash C$ where B and C are from the language of $\otimes$, $\leftarrow$ and $\rightarrow$.

However, if we do "overload" the structural connectives s, r_1 and r_2, in this way, we have to be careful about accepting constructs like $s(A, B) \vdash s(A, B)$ as axioms, since now, the occurrences of s are proxies for potentially *different* logical connectives. The "axiom" $s(X, Y) \vdash s(X, Y)$ is even more dangerous because, here, almost everything could be overloaded.

The solution is simply to ban such "ambiguous" axioms and restrict all axioms to be of the form $p \vdash p$, where p is a primitive proposition. We have to leave room for extra axiomatic rules like $I \vdash \top$ which we may wish to incorporate at a later stage (see the completeness argument in the proof of Theorem 4.13). The idea is to exclude all instances of $s(X, Y) \vdash s(X, Y)$, $r_1(X, Y) \vdash r_1(X, Y)$, and $r_2(X, Y) \vdash r_2(X, Y)$.

3.6 Duality Rules OK!

When we "dualise" the $\{s, r_1, r_2\}$ display postulates of Method 3 from Section 3.4, we obtain the display postulates shown below:

$$\text{(dp-dual)}\quad \begin{array}{c} r_1(Z, Y) \vdash X \\ \hline\hline Z \vdash s(X, Y) \\ \hline\hline r_2(X, Z) \vdash Y \end{array}$$

The "functions" s, r_1, r_2 have the same tonicity as their duals, hence there is no confusion, which is why we can reuse the same function names.

We can now follow a similar strategy and invent three new logical connectives $\oplus$, $\succ$ and $\prec$. These last two connectives are supposed to be reminiscent of "not-but" and "but-not". The "rewrite" introduction rules for these connectives are immediate from the previous discussions. And invertibility considerations then give the other introduction rules for free.

That is, the two halves share the structural connectives s, r_1 and r_2 by overloading them to mean different things in different contexts.

3.7 Display Logic

We can now unravel the previously problematic endsequent as shown below, although the fragment shown below is still not a proof since the leaf is not $p \vdash p$:

$$\cfrac{\cfrac{\cfrac{r_1(s(B, A), X) \vdash A \otimes B}{s(B, A) \vdash s(A \otimes B, X)}\,\text{(dp-dual)}}{B \otimes A) \vdash s(A \otimes B, X)}\,(\otimes \vdash)}{r_2((A \otimes B), (B \otimes A)) \vdash X}\,\text{(dp-dual)}$$

That is, the "twin" display set-up enjoys a general display property as well as a limited display property [22], allowing us to seek proofs for arbitrary sequents of the form $X \vdash Y$, as well as formula-sequents of the form $B \vdash C$.

We now generalise all these observations.

4 The General Case

For our purposes we need to extend Dunn's definition slightly. A partial gaggle $\mathcal{T} := (A, \leq, OP)$ with an n-ary function f_0 as head is **fully-founded** iff the j-th contrapositive f_j of f_0 is in OP for every $1 \leq j \leq n$. Any partial gaggle with head f_0 can be extended into a fully-founded one simply by adding the required functions and contraposition conditions.

Example 4.1 *For example, if $(A, \leq)$ is a partially ordered set, then the structure $\mathcal{T} := (A, \leq, \{\otimes, \leftarrow, \rightarrow\})$ where all operations are binary and have traces*

$$\otimes : (-,-) \mapsto (-) \qquad \leftarrow : (+,-) \mapsto (+) \qquad \rightarrow : (-,+) \mapsto (+)$$

is a tonoid [14, 22]. If these operations satisfy

$$a \otimes b \leq c \quad \textit{iff} \quad a \leq c \leftarrow b \quad \textit{iff} \quad b \leq a \rightarrow c$$

for every $a, b, c \in A$, then $\{\otimes, \leftarrow\}$ satisfy the Abstract Law Of Residuation in the 1-st place, $\{\otimes, \rightarrow\}$ satisfy the Abstract Law Of Residuation in the 2-nd place, and this structure is a fully-founded partial gaggle with head $\otimes$ [14, 22].

Example 4.2 *A fully founded partial gaggle of n-ary functions need not consist of $n+1$ functions. For example, if $(B, \leq)$ is a partially ordered set, then the structure $\mathcal{S} := (B, \leq, \{\rightarrow, \otimes\})$ where all operations are binary and have traces*

$$\rightarrow : (-,+) \mapsto (+) \qquad \otimes : (-,-) \mapsto (-)$$

is a tonoid. If these operations satisfy

$$c \leq a \rightarrow b \quad \textit{iff} \quad a \leq c \rightarrow b \quad \textit{iff} \quad a \otimes c \leq b$$

for every $a, b, c \in A$, then $\rightarrow$ satisfies the Abstract Law Of Residuation in the 1-st place with itself, and $\rightarrow$ satisfies the Abstract Law Of Residuation in the 2-nd place with $\otimes$. This structure is a partial gaggle with head $\rightarrow$. Furthermore, $\otimes$ is commutative viz:

$$a \otimes c \leq b \quad \textit{iff} \quad a \leq c \rightarrow b \quad \textit{iff} \quad c \leq a \rightarrow b \quad \textit{iff} \quad c \otimes a \leq b$$

A display logic for such a partial gaggle can be found in [18].

Assume we are given a fully-founded partial gaggle $\mathcal{T} := (A, \leq, OP)$ based on $n+1$, distinct, n-ary functions $OP = \{f_0, \ldots, f_n\}$, with head f_0 and where all functions have *distinct* traces[3]. Let $\mathcal{T}^\Delta := (A, \leq, OP^\Delta)$ be the structure consisting of the functions $OP^\Delta = \{f_0^\Delta, \ldots, f_n^\Delta\}$, each of arity n, such that the trace for f_i^Δ is the trace obtained by multiplying each component of the trace of f_i by $-$. Thus $\mathcal{T}^\Delta := (A, \leq, OP^\Delta)$ is unique. Furthermore, $\mathcal{T}^\Delta := (A, \leq, OP^\Delta)$ satisfies $S(f_0^\Delta, a_1, \ldots, a_j, \ldots, a_n, b)$ iff $S(f_j^\Delta, a_1, \ldots, b, \ldots, a_n, a_j)$, for every $1 \leq j \leq n$.

That is, for $1 \leq j \leq n$, the functions f_0^Δ and f_j^Δ are also contrapositives in the j-th place, they also satisfy the Abstract Law of Residuation in the j-th place, and $\mathcal{T}^\Delta := (A, \leq, OP^\Delta)$ is also a fully-founded partial gaggle, but with head f_0^Δ.

[3] This is the most general case.

Example 4.3 *If* $\mathcal{T} := (A, \leq, \{\otimes, \leftarrow, \rightarrow\})$ *is the fully-founded partial gaggle of Example 4.1 then* $\mathcal{T}^{\Delta} := (A, \leq, \{\otimes^{\Delta}, \rightarrow^{\Delta}, \leftarrow^{\Delta}\})$ *with traces*

$$\otimes^{\Delta} : (+,+) \mapsto (+) \qquad \leftarrow^{\Delta} : (-,+) \mapsto (-) \qquad \rightarrow^{\Delta} : (+,-) \mapsto (-)$$

is also a tonoid. If these operations satisfy

$$c \leq a \otimes^{\Delta} b \quad \textit{iff} \quad c \leftarrow^{\Delta} b \leq a \quad \textit{iff} \quad a \rightarrow^{\Delta} c \leq b$$

for every $a, b, c \in A$*, then they satisfy the Abstract Law Of Residuation in all the appropriate places and this structure is a fully-founded partial gaggle with head* $\otimes^{\Delta}$*.*

In order to obtain a display calculus we now invent new structural connectives. For every function f_i, $0 \leq i \leq n$, let F_i be a new structural connective with the same trace as f_i. For every function f_i^{Δ}, $0 \leq i \leq n$, let F_i^{Δ} be a new structural connective with the same trace as f_i^{Δ}. The functions f will be our logical connectives and the functions F our structural connectives.

To capture rules we define $\mathcal{S}$, an analogue of Dunn's S [14] except that $\mathcal{S}$ is couched in terms of a logical, structural or identity (ι) operation op, upper case variables, $\vdash$, and an extra argument for greater versatility viz:

$$\mathcal{S}(\pm, op, X_1, \ldots, X_n, Y) := \begin{cases} op(X_1, \ldots, X_n) \vdash Y & \text{if} \quad \pm = - \\ Y \vdash op(X_1, \ldots, X_n) & \text{if} \quad \pm = + \end{cases}$$

We regain the essence of Dunn's S if $\pm$ is the output $\pm_{n+1}$ of the trace of op:

$$\mathcal{S}(op, X_1, \ldots, X_n, Y) := \mathcal{S}(\pm_{n+1}, op, X_1, \ldots, X_n, Y)$$

Using this notation, posit the following **display postulates** for F_0 and F_j, $1 \leq j \leq n$ where the (upper case) X_j and Y are arbitrary but distinct structure variables:

$$(\text{dp}) \ \frac{\mathcal{S}(F_j, X_1, \ldots, X_{j-1}, Y, X_{j+1}, \ldots, X_n, X_j)}{\overline{\mathcal{S}(F_0, X_1, \ldots, X_{j-1}, X_j, X_{j+1}, \ldots, X_n, Y)}}$$

This gives a collection of display postulates where, for $1 \leq j \leq n$:

$$\frac{\mathcal{S}(F_j, X_1, \ldots, X_{j-1}, Y, X_{j+1}, \ldots, X_n, X_j)}{\overline{\mathcal{S}(F_{j+1}, X_1, \ldots, X_{j-1}, X_j, Y, X_{j+2}, \ldots, X_n, X_{j+1})}}$$

Posit another set of (dual) display postulates by uniformly replacing F_0 by F_0^{Δ}, and replacing F_j by F_j^{Δ}. Leave $\mathcal{S}$ alone since it changes its behaviour automatically to cater for the dualised traces of f_i^{Δ} and F_i^{Δ}.

Example 4.4 *For the fully-founded partial gaggles of Example 4.3, we obtain the following display postulates:*

$$(dp) \ \begin{array}{c} X_1 \vdash F_{\leftarrow}(Y, X_2) \\ \hline\hline F_{\otimes}(X_1, X_2) \vdash Y \\ \hline\hline X_2 \vdash F_{\rightarrow}(X_1, Y) \end{array} \qquad (dp^{\Delta}) \ \begin{array}{c} F^{\Delta}_{\leftarrow^{\Delta}}(Y, X_2) \vdash X_1 \\ \hline\hline Y \vdash F^{\Delta}_{\otimes^{\Delta}}(X_1, X_2) \\ \hline\hline F^{\Delta}_{\rightarrow^{\Delta}}(X_1, Y) \vdash X_2 \end{array}$$

To construct a display calculus, however, we must overload the structural connectives, analogous to the way that Gentzen's "," is overloaded to mean logical $\wedge/\vee$ when it appears on the left/right of the turnstile.

Lemma 4.5 *The collection of display postulates (dp) has the display property when we put $F_i = F_i^\Delta$ for $0 \leq i \leq n$.*

PROOF. By inspection of the display postulates, there is at least one entry which has every structural connective as the outermost connective, on either side of the turnstile, and all structure variables are arbitrary. ∎

So uniformly put $F_i = F_i^\Delta = Ⓢ_i$ for $0 \leq i \leq n$. Then, for each function f, define the following introduction rules where the A_i are arbitrary distinct formula variables, the X_i are arbitrary distinct structure variables, and ι is the identity function:
If the trace of f is $f : (\pm_1, \ldots, \pm_n) \mapsto (\pm_{n+1})$ then

$$(\mathrm{dec})\ \frac{\mathcal{S}(\pm_1, \iota, X_1, A_1) \cdots \mathcal{S}(\pm_n, \iota, X_n, A_n)}{\mathcal{S}(\pm_{n+1}, Ⓢ, X_1, \ldots, X_n, f(A_1, \ldots, A_n))} \quad (\mathrm{rew})\ \frac{\mathcal{S}(\pm_{n+1}, Ⓢ, A_1, \ldots, A_n, Y)}{\mathcal{S}(\pm_{n+1}, f, A_1, \ldots, A_n, Y)}$$

where "dec" stands for "decoding" and "rew" stands for "rewriting"; see Section 3.

We now have the ingredients to construct a Gentzen system since we have rules to introduce every connective into the left and right hand side of a sequent. Let $\delta\mathbf{OP}$ be the single (display) calculus containing all the display postulates and introduction rules obtained from the fully-founded partial gaggles OP and OP^Δ using the steps outlined above. We make minor additions in the proof of Theorem 4.13.

If we collapse F_i and F_i^Δ their traces no longer make sense since we cannot tell them apart, so in the sequel we often keep the distinction, even though we need not.

Example 4.6 *Continuing on from Example 4.4 we obtain the display postulates and introduction rules shown in Figure 2. To make the rules easier to read we have replaced $Ⓢ_i$ by more meaningful* infix *structural symbols (connectives), and these are overloaded as shown below:*

value of i		*0*	*1*	*2*
functions from $\mathcal{T}$	f_i	$\otimes$	$\leftarrow$	$\rightarrow$
structural proxy	$Ⓢ_i$	;	$<$	$>$
functions from $\mathcal{T}^\Delta$	f_i^Δ	$\oplus$	$\prec$	$\succ$

The logical connectives come in (dual) pairs, with each component of a pair captured by the same structural connective, but in different (antecedent or succedent) positions. For example, the structural connective " ; " in an antecedent position represents $\otimes$, but in a succedent position represents $\oplus$, whereas the structural connective " > " in an antecedent position represents $\succ$ but in a succedent position represents $\rightarrow$. Thus, just as in Gentzen's original system, each structural connective is overloaded. In [22] we show that the rules from Figure 2 form the basis of most substructural logics.

We have now overloaded $Ⓢ_i$ to mean F_i and F_i^Δ. The definition, and hence appearance, of $\mathcal{S}(F, X_1, \ldots, X_n, f(A_1, \ldots, A_n))$ depends upon the trace of F, which is

$$
(\mathrm{dp})\ \begin{array}{c} X \vdash Z < Y \\ \hline\hline X \,;\, Y \vdash Z \\ \hline\hline Y \vdash X > Z \end{array}
\qquad
(\mathrm{dp})\ \begin{array}{c} Z < Y \vdash X \\ \hline\hline Z \vdash X \,;\, Y \\ \hline\hline X > Z \vdash Y \end{array}
$$

$$
(\otimes \vdash)\ \frac{A \,;\, B \vdash Z}{A \otimes B \vdash Z}
\qquad
(\vdash \otimes)\ \frac{X \vdash A \quad Y \vdash B}{X \,;\, Y \vdash A \otimes B}
$$

$$
(\oplus \vdash)\ \frac{A \vdash X \quad B \vdash Y}{A \oplus B \vdash X \,;\, Y}
\qquad
(\vdash \oplus)\ \frac{Z \vdash A \,;\, B}{Z \vdash A \oplus B}
$$

$$
(\leftarrow\vdash)\ \frac{A \vdash X \quad Y \vdash B}{A \leftarrow B \vdash X < Y}
\qquad
(\vdash\leftarrow)\ \frac{Z \vdash A < B}{Z \vdash A \leftarrow B}
$$

$$
(\prec\ \vdash)\ \frac{A < B \vdash Z}{A \prec B \vdash Z}
\qquad
(\vdash\ \prec)\ \frac{X \vdash A \quad B \vdash Y}{X < Y \vdash A \prec B}
$$

$$
(\rightarrow\vdash)\ \frac{X \vdash A \quad B \vdash Y}{A \rightarrow B \vdash X > Y}
\qquad
(\vdash\rightarrow)\ \frac{Z \vdash A > B}{Z \vdash A \rightarrow B}
$$

$$
(\succ\ \vdash)\ \frac{A > B \vdash Z}{A \succ B \vdash Z}
\qquad
(\vdash\ \succ)\ \frac{A \vdash X \quad Y \vdash B}{X > Y \vdash A \succ B}
$$

FIG. 2. Binary Display Postulates and Intensional Introduction Rules

fine since F has the same trace as f. But when we uniformly replace F by $\circledS$, then $\mathcal{S}(\circledS, X_1, \ldots, X_n, f(A_1, \ldots, A_n))$ is meaningless since $\circledS$ does not have a trace; it is overloaded to stand for both F and F^{Δ}. This is why the more general rules require the extra argument for $\mathcal{S}$. But note that the functions f_i and f_i^{Δ} have the same tonicity vector since the trace of f_i^{Δ} is obtained from the trace of f_i. Thus, each $\circledS_i$ does have a unique tonicity $\mathrm{tn}(\circledS_i, j, \pm_j)$, for every $1 \leq j \leq n$.

In a sequent $V \vdash W$, the structure V is an **antecedent part** and the structure W is a **succedent part**. Furthermore, if the structure $\circledS_i(X_1, \ldots, , X_n)$ with outermost structural connective $\circledS_i$ is an antecedent/succedent part, the substructure X_j is an **antecedent/succedent** part if $\mathrm{tn}(\circledS_i, j, +)$, and a **succedent/antecedent** part if $\mathrm{tn}(\circledS_i, j, -)$. Intuitively, a substructure sitting in an argument position marked $+/-$ inherits the same/opposite "polarity" as its superstructure.

Two sequents σ and σ' are **structurally equivalent** if we can pass from one to the other (and back) using only the display postulates shown above. The name Display Logic comes from the following theorem (which does not hold for subformulae of a formula!).

Theorem 4.7 (Belnap) *For every sequent σ and every antecedent [respectively succedent] part X of σ, there is a structurally equivalent sequent σ' that has X (alone) as* *its antecedent [succedent]. X is said to be "displayed" in σ'.*

PROOF. The display postulates for OP guarantee that the j-th argument for F_0 is displayed, in one (dp) move, by its j-th contrapositive F_j, and vice-versa, for all $1 \leq j \leq n$. If we want to display the i-th argument X_i, $i \geq 1$, of F_j, $j \neq 0$, $j \neq i$, then we can first move to F_0, which does not perturb the position of X_i, and then move to F_i, which displays X_i. A similar argument holds for OP^{Δ}, F_0^{Δ} and F_j^{Δ}. ∎

Example 4.8 *The display postulates shown in Figure 2 enjoy the display property by inspection because there is at least one entry with each structural connective as the main connective in each set of display postulates, and these entries are couched in terms of completely arbitrary structural variables.*

The logical connective introduced by the logical rules is always displayed as the whole of the antecedent or as the whole of the succedent, so these rules do not perturb the display property. The two rules shown below which respectively allow us to terminate backward proof search and compose proofs are common to all Display Logics. Note that we can cut on formulae only.

$$\text{(id)}\ \ p \vdash p \qquad\qquad \text{(cut)}\ \ \frac{X \vdash A \quad A \vdash Y}{X \vdash Y}$$

The restriction of identity to propositional variables only is sufficient because

Lemma 4.9 *For any formula A, the sequent $A \vdash A$ is provable.*

PROOF. By induction on the formation of A we show that for every function f, with trace $f : (\pm_1, \cdots, \pm_n) \mapsto (\pm_{n+1})$, the sequent $f(A_1, \ldots, A_n) \vdash f(A_1, \ldots, A_n)$ is provable. For example, if the output $\pm_{n+1}$ of the trace for f is $-$ then the proof sequence shown below suffices since each $\mathcal{S}(\pm_i, \iota, A_i, A_i)$, $1 \leq i \leq n$, is itself an instance of the induction hypothesis $A_i \vdash A_i$:

$$\cfrac{\cfrac{\mathcal{S}(\pm_1, \iota, A_1, A_1) \quad \ldots \quad \mathcal{S}(\pm_n, \iota, A_n, A_n)}{\circledS(A_1, \ldots, A_n) \vdash f(A_1, \ldots, A_n)}\,(\vdash f)}{f(A_1, \ldots, A_n) \vdash f(A_1, \ldots, A_n)}\,(f \vdash)$$

A dual argument suffices when $\pm_{n+1}$ is $+$. In fact, all these proofs are instances of:

$$\cfrac{\cfrac{\mathcal{S}(\pm_1, \iota, A_1, A_1) \quad \ldots \quad \mathcal{S}(\pm_n, \iota, A_n, A_n)}{\mathcal{S}(\pm_{n+1}, \circledS, A_1, \ldots, A_n, f(A_1, \ldots, A_n))}\,(\text{dec})}{\mathcal{S}(\pm_{n+1}, f, A_1, \ldots, A_n, f(A_1, \ldots, A_n))}\,(\text{rew})$$

∎

In the introduction we argued that the cut-rule is bad for backward proof search. We now show the redundancy of this rule.

Lemma 4.10 *The calculus* δ**OP** *satisfies Belnap's conditions C1-C8 (which are given in the Appendix).*

PROOF. That the calculus satisfies C1-C7 is by inspection of the rules since all rules are couched in terms of different structural variables $X_1, \ldots, X_n$ and all rules respect the tonicities of the associated functions. For example, to see this for the generalised rule (dec) shown above Example 4.6, plug in a few values for $\pm_{n+1}$ and $\pm_j$, $1 \le j \le n$, noting that the conclusion of the rule is couched in terms of Ⓢ rather than f, and that the tonicity for f and Ⓢ in the j-th position is given by the "product" $\pm_{n+1}\pm_j$, for $1 \le j \le n$ [22, 14].

The main problem is to prove that C8 holds generally. This we do next [21].

The hypothesis is that we are given an instance of the cut rule with cut formula A, and that both the displayed occurrences of A in the premisses of this instance of cut are principal occurrences. The C8 condition is that, in this case, we must be able to derive the conclusion of this cut rule instance using only the display postulates, the original premisses of the principal occurrences of A and cuts on formulae strictly smaller than A. We proceed by considering the main connective f_i of A, some $0 \le i \le n$ with trace $f_i : (\pm_1, \cdots, \pm_n) \mapsto (\pm_{n+1})$.

Suppose the trace output $\pm_{n+1}$ of f_i is $-$, and $i \neq 0$. If $i = 0$ we simply start the procedure at a later point in this proof; see below. Then, with $i \neq 0$, the instance of cut we need to reduce is as shown below:

$$\dfrac{\dfrac{\mathcal{S}(\pm_1, \iota, X_1, A_1) \cdots \mathcal{S}(\pm_n, \iota, X_n, A_n)}{\mathcal{S}(-, F_i, X_1, \ldots, X_n, f_i(A_1, \ldots, A_n))}\,(\vdash f_i) \quad \dfrac{\mathcal{S}(-, F_i, A_1 \ldots, A_i, \ldots, A_n, Y)}{\mathcal{S}(-, f_i, A_1 \ldots, A_i, \ldots, A_n, Y)}\,(f_i \vdash)}{\mathcal{S}(-, F_i, X_1, \ldots, X_n, Y)}\,(\text{cut})$$

If the trace output $\pm_{n+1}$ of f_i is $+$ then the rules labelling the premisses are interchanged with each other, and all explicitly shown occurrences of $-$ are uniformly replaced with $+$.

Similar arguments work for $f_i^{\blacktriangle}$.

Generalising, if the trace of f is $f : (\pm_1, \cdots, \pm_n) \mapsto (\pm_{n+1})$, all cases of "principal cuts" can be captured by:

$$\dfrac{\dfrac{\mathcal{S}(\pm_1, \iota, X_1, A_1) \cdots \mathcal{S}(\pm_n, \iota, X_n, A_n)}{\mathcal{S}(\pm_{n+1}, Ⓢ, X_1, \ldots, X_n, f(A_1, \ldots, A_n))}\,(\text{dec}) \quad \dfrac{\mathcal{S}(\pm_{n+1}, Ⓢ, A_1, \ldots, A_n, Y)}{\mathcal{S}(\pm_{n+1}, f, A_1, \ldots, A_n, Y)}\,(\text{rew})}{\mathcal{S}(\pm_{n+1}, Ⓢ, X_1, \ldots, X_n, Y)}\,(\text{cut})$$

Rather than give full generality we continue with the instance where the trace output of f_i is assumed to be $-$, and return to the use of F_i as a concrete instance of Ⓢ$_i$. Recall that the trace of f_i is $f_i : (\pm_1, \cdots, \pm_n) \mapsto (-)$, and that $i \neq 0$.

Starting from $\mathcal{S}(F_i, A_1, \ldots, A_i, \ldots, A_n, Y)$ we first move to F_0, thereby displaying A_i, then cut on A_i, thereby replacing it with X_i:

$$\dfrac{\mathcal{S}(\pm_i, \iota, X_i, A_i) \quad \dfrac{\mathcal{S}(F_i, A_1, \ldots, A_{i-1}, A_i, A_{i+1}, \ldots, A_n, Y)}{\mathcal{S}(F_0, A_1, \ldots, A_{i-1}, Y, A_{i+1}, \ldots, A_n, A_i)}\,(\text{dp})}{\mathcal{S}(F_0, A_1, \ldots, A_{i-1}, Y, A_{i+1}, \ldots, A_n, X_i)}\,(\text{cut})$$

The sequent $\mathcal{S}(\pm_i, \iota, X_i, A_i)$ is taken from the premiss of the given instance of (dec), ensuring that the two displayed occurrences of A_i in the premisses of the new instance of (cut) are on opposite sides as required (recall that $i \neq 0$ by supposition).

Next we show how to move from F_0 to an arbitrary F_j, $j < i$, thereby displaying A_j, and then show how to replace it with X_j, using a cut, and move back to F_0:

$$\dfrac{\mathcal{S}(\pm_j, \iota, X_j, A_j) \quad \dfrac{\mathcal{S}(F_0, A_1, \ldots, A_{i-1}, Y, A_{i+1}, \ldots, A_n, X_i)}{\mathcal{S}(F_j, A_1, \ldots, A_{j-1}, X_i, A_{j+1}, \ldots, A_{i-1}, Y, A_{i+1}, \ldots, A_n, A_j)}\,(\mathrm{dp})}{\dfrac{\mathcal{S}(F_j, A_1, \ldots, A_{j-1}, X_i, A_{j+1}, \ldots, A_{i-1}, Y, A_{i+1}, \ldots, A_n, X_j)}{\mathcal{S}(F_0, A_1, \ldots, A_{j-1}, X_j, A_{j+1}, \ldots, A_{i-1}, Y, A_{i+1}, \ldots, A_n, X_i)}\,(\mathrm{dp})}\,(\mathrm{cut})$$

The proof fragment above illustrates how the pure cut rule of Display Logic simulates more complex "surgical" or "substitutional" cut rules by using the display postulates to display the cut-formula, perform the cut, and "undisplay" its replacement. In this proof fragment, for example, the only difference between the top sequent and the bottom sequent is the substitution of A_j by X_j.

Contiguous applications for all $j < i$ produces

$$\mathcal{S}(F_0, X_1, \ldots, X_{i-1}, Y, A_{i+1}, \ldots, A_n, X_i)$$

(If $i = 0$ then we would start here.) A similar procedure for $j > i$ then gives

$$\mathcal{S}(F_0, X_1, \ldots, X_{i-1}, Y, X_{i+1}, \ldots, X_n, X_i)$$

The desired sequent, $\mathcal{S}(F_i, X_1, \ldots, X_n, Y)$, which was the conclusion of the initial cut, is obtained by a final application of (dp):

$$\dfrac{\mathcal{S}(F_0, X_1, \ldots, X_{i-1}, Y, X_{i+1}, \ldots, X_n, X_i)}{\mathcal{S}(F_i, X_1, \ldots, X_{i-1}, X_i, X_{i+1}, \ldots, X_n, Y)}\,(\mathrm{dp})$$

All cuts in this procedure are on formulae of the form A_j, $1 \leq j \leq n$, which are strict subformulae of the initial cut-formula $f_i(A_1, \ldots, A_n)$. Furthermore, the only ancillary sequents, $\mathcal{S}(\pm_j, \iota, X_j, A_j)$, $1 \leq j \leq n$, are (all of) the original premisses of the given instance of (dec), as required.

The same argument caters for operations from OP^{Δ}.

Thus C8 is satisfied. ∎

Since the rules obey Belnap's conditions C1-C8 we obtain cut-elimination for free:

Theorem 4.11 (cut-elimination) *If there is a proof of the sequent $X \vdash Y$ in $\delta\mathbf{OP}$, then there is a cut-free proof of $X \vdash Y$ in $\delta\mathbf{OP}$ [4].*

As stated previously, one introduction rule, when rules are read backwards, is always a simple "rewrite" rule where a logical connective is replaced by a structural connective. The other rule (when read backwards) then "decodes" for that connective on the other side; see Figure 2 for concrete examples. In Section 3 we showed that this is not an accident and that the "decoding" rules can be *derived* from the "rewrite" rules. Such rule pairs enjoy the following

Theorem 4.12 *Every rewrite rule is invertible: if the conclusion of the rule is provable then so are each of the premisses.*

PROOF. If the output $\pm_{n+1}$ of the trace for f is $-$ [respectively $+$] then the "rewrite" rule shown below is $(f \vdash)$ [$(\vdash f)$]:

$$\text{(rew)}\ \frac{\mathcal{S}(\pm_{n+1}, \circledS, A_1, \ldots, A_n, Y)}{\mathcal{S}(\pm_{n+1}, f, A_1, \ldots, A_n, Y)}$$

Every proof of invertibility is an instance of the following where the trace of f is $f : (\pm_1, \cdots, \pm_n) \mapsto (\pm_{n+1})$, and label (dec) is $(\vdash f)$ [respectively $(f \vdash)$] if the output $\pm_{n+1}$ of the trace for f is $-$ [$+$]:

$$\dfrac{\dfrac{\mathcal{S}(\pm_1, \iota, A_1, A_1) \quad \ldots \quad \mathcal{S}(\pm_n, \iota, A_n, A_n)}{\mathcal{S}(\pm_{n+1}, \circledS, A_1, \ldots, A_n, f(A_1, \ldots, A_n))}\text{(dec)} \qquad \mathcal{S}(\pm_{n+1}, f, A_1, \ldots, A_n, Y)}{\mathcal{S}(\pm_{n+1}, \circledS, A_1, \ldots, A_n, Y)}\text{(cut)}$$

In this proof, each $\mathcal{S}(\pm_i, \iota, A_i, A_i)$, $1 \le i \le n$, is an instance of Lemma 4.9. ∎

In proof search, we can therefore apply the (rew) rules at any time, preferably as early in the proof search as possible. Consequently, the order of application of the "decoding" rules is the crux of proof search.

Using Dunn's representation results for partial gaggles (Theorem 2.2) we obtain

Theorem 4.13 *If $\mathcal{T} := (A, \le, OP)$ is a fully-founded partial gaggle then $\delta\mathbf{OP}$ is a* cut-free *display calculus which is automatically* sound and complete *with respect to the gaggle-theoretic relational semantics associated with the partial gaggle operations.*

PROOF. For the soundness proof we translate a sequent $X \vdash Y$ into a quasi-equation $\tau(X \vdash Y)$ using a generic translation τ [40]:

$$\tau(X \vdash Y) := \tau_1(X) \leq \tau_2(Y)$$

$$\tau_1(A) := \tau_2(A) := A$$

$$\tau_1(Ⓢ_i(X_1, \ldots, X_n)) := \begin{cases} f_i(\tau'(X_1), \ldots, \tau'(X_n)) & \text{if} \quad \text{trace of } f_i \text{ has output } - \\ f_i^{\Delta}(\tau'(X_1), \ldots, \tau'(X_n)) & \text{if} \quad \text{trace of } f_i \text{ has output } + \end{cases}$$

$$\text{where } \tau'(X_j) \begin{cases} \tau_1(X_j) & \text{if} \quad \mathrm{tn}(f_i, j, +) \\ \tau_2(X_j) & \text{if} \quad \mathrm{tn}(f_i, j, -) \end{cases}$$

$$\tau_2(Ⓢ_i(X_1, \ldots, X_n)) := \begin{cases} f_i^{\Delta}(\tau''(X_1), \ldots, \tau''(X_n)) & \text{if} \quad \text{trace of } f_i \text{ has output } - \\ f_i(\tau''(X_1), \ldots, \tau''(X_n)) & \text{if} \quad \text{trace of } f_i \text{ has output } + \end{cases}$$

$$\text{where } \tau''(X_j) \begin{cases} \tau_2(X_j) & \text{if} \quad \mathrm{tn}(f_i, j, +) \\ \tau_1(X_j) & \text{if} \quad \mathrm{tn}(f_i, j, -) \end{cases}$$

We have to prove that if the sequent $X \vdash Y$ is provable in $\delta\mathbf{OP}$ then the quasi-equation $\tau(X \vdash Y)$ is "valid" in the original partial gaggle $\mathcal{T} := (A, \leq, OP)$. The proof proceeds by induction on the given proof of $X \vdash Y$ and requires that for every introduction rule and every display postulate: if the τ-translation of the premisses is "valid" then so is the τ-translation of the conclusion.

The identity rule is the base case and simply τ-translates to the quasi-equation $p \leq p$ which is alway "valid". Each of the "rewrite" rules is trivial since the premiss and conclusion τ-translate to the same quasi-equation. The display postulates τ-translate to the various contraposition conditions that make up the definition of $\mathcal{T} := (A, \leq, OP)$. The only remaining rules are the "decoding" introduction rules which are of the form

$$(\mathrm{dec}) \quad \frac{\mathcal{S}(\pm_1, \iota, X_1, A_1) \cdots \mathcal{S}(\pm_n, \iota, X_n, A_n)}{\mathcal{S}(\pm_{n+1}, Ⓢ, X_1, \ldots, X_n, f(A_1, \ldots, A_n))}$$

and translate into

$$\frac{S(\pm_1, \iota, \tau'(X_1), A_1) \cdots S(\pm_n, \iota, \tau'(X_n), A_n)}{f_i(\tau'(X_1), \ldots, \tau'(X_n)) \leq f_i(A_1, \ldots, A_n)} \text{ if trace } f_i : (\pm_1, \cdots, \pm_n) \mapsto (-)$$

$$\frac{S(\pm_1, \iota, \tau''(X_1), A_1) \cdots S(\pm_n, \iota, \tau''(X_n), A_n)}{f_i(A_1, \ldots, A_n) \leq f_i(\tau''(X_1), \ldots, \tau''(X_n))} \text{ if trace } f_i : (\pm_1, \cdots, \pm_n) \mapsto (+)$$

where S is Dunn's gaggle-theoretic notation for writing quasi-equations extended appropriately to be the notational equivalent for our $\mathcal{S}$. Not surprisingly, the translated "rules" just reiterate that the function f has tonicity vector $\mathrm{tn}(f, j, \pm_j)$, for $1 \leq j \leq n$; see Section 3.1.

For completeness, we show that the Lindenbaum algebra obtained from the provably equivalent *formulae* of $\delta\mathbf{OP}$ is a partial gaggle with the same properties as the original partial gaggle $\mathcal{T} := (A, \leq, OP)$ which gave rise to $\delta\mathbf{OP}$ [20].

Specifically, let Fml be the set of all formulae and let $[OP] := \{[f_0], \ldots, [f_n]\}$ be a set of $n+1$, distinct n-ary function names. Now construct the Lindenbaum algebra:

$$[\mathcal{T}] := \langle Fml/\!\equiv, [\leq], [OP]\rangle$$

$$\begin{array}{lll} A \equiv B & \text{if} & A \vdash B \text{ and } B \vdash A \\ [A] & := & \{B \in Fml \mid A \equiv B\} \\ Fml/\!\equiv & := & \{[A] : A \in Fml\} \\ [A][\leq][B] & \text{if} & A \vdash B \\ [f]([A_1], \ldots, [A_n]) & := & f(A_1, \ldots, A_n) \end{array}$$

It can be shown that $\equiv$ is not only an equivalence relation, but is also a congruence.

We show that in this construction, the function $[f]$ has the same trace as the function f from $\mathcal{T} := (A, \leq, OP)$. So suppose that f has trace $f : (\pm_1, \cdots, \pm_n) \mapsto (\pm_{n+1})$. We consider only one of the four conditions which ensure that f respects the bounds [22, 14].

For example, suppose $\pm_{n+1} = +$ and that for some j, $1 \leq j \leq n$:

$$(\pm_j, \pm_{n+1}) = (+,+) \text{ if } \mathrm{tn}(f, j, +) \text{ and } (A_j = \top \Rightarrow f(A_1, \ldots, A_n) = \top)$$

To show that the trace of $[f]$ has identical trace output $\pm_{n+1} = +$ and identical j-th component $\pm_j = +$,we must show for this j that:

$$\mathrm{tn}([f], j, +) \text{ and } ([A_j] = [\top] \Rightarrow [f]([A_1], \ldots, [A_n]) = [\top])$$

which immediately reduces to showing that:

$$\mathrm{tn}([f], j, +) \text{ and } (A_j \dashv\vdash \top \Rightarrow f(A_1, \ldots, A_n) \dashv\vdash \top)$$

The first conjunct, $\mathrm{tn}([f], j, +)$, reduces to showing that:

$$\text{if } A_j \vdash B_j \text{ then } f(A_1, \ldots, A_j, \ldots, A_n) \vdash f(A_1, \ldots, A_{j-1}, B_j, A_{j+1}, \ldots, A_n)$$

which is almost an instance of Lemma 4.9 (thus covering the other three cases)

$$\cfrac{\cfrac{\mathcal{S}(\pm_1, \iota, A_1, A_1) \ldots \mathcal{S}(\pm_j, \iota, B_j, A_j) \ldots \mathcal{S}(\pm_n, \iota, A_n, A_n)}{\mathcal{S}(+, \circledS, A_1, \ldots, B_j, \ldots, A_n, f(A_1, \ldots, A_j, \ldots, A_n)}\,(\mathrm{dec})}{\mathcal{S}(+, f, A_1, \ldots, B_j, \ldots, A_n, f(A_1, \ldots, A_j, \ldots, A_n))}\,(\mathrm{rew})$$

except that $\mathcal{S}(\pm_j, \iota, A_j, B_j)$ is $A_j \vdash B_j$ since the pair $(\pm_j, \pm_{n+1})$ is $(+,+)$.

To handle the second conjunct we need to augment $\delta\mathbf{OP}$ with a structural connective I (say) to capture $\top$ and $\bot$, and add a couple of extra rules to ensure $X \vdash \top$ and $\bot \vdash X$ for any structure X. We omit the details, which can be found in [22].

Then, the second conjunct reduces to showing that:

$$(\top \vdash A_j \Rightarrow \top \vdash f(A_1, \dots, A_n))$$

Since we know that the pair $(\pm_j, \pm_{n+1})$ is $(+, +)$, the following proof suffices:

$$\dfrac{\dfrac{\dfrac{\overset{\text{instance of } X \vdash \top}{Ⓢ^j(A_1, \dots, A_{j-1}, \top, A_{j+1}, \dots, A_n) \vdash \top} \quad \top \vdash A_j}{Ⓢ^j(A_1, \dots, A_{j-1}, \top, A_{j+1}, \dots, A_n) \vdash A_j}\,(\text{cut})}{\top \vdash Ⓢ(A_1, \dots, A_n)}\,(\text{dp})}{\top \vdash f(A_1, \dots, A_n)}\,(\vdash f)$$

where $Ⓢ^j$ is the j-th contrapositive of Ⓢ. It is actually an instance of:

$$\dfrac{\dfrac{\dfrac{\overset{\substack{\text{instance of} \\ \top \vdash X \text{ or } X \vdash \bot}}{\mathcal{S}(\pm_j, \iota, A_j, \top/\bot)} \quad \overset{\substack{\text{now axiomatic instance of} \\ X \vdash \top \text{ or } \bot \vdash X}}{\mathcal{S}(-\pm_j, Ⓢ^j, A_1, \dots, A_{j-1}, \top/\bot, A_{j+1}, \dots, A_n, \top/\bot)}}{\mathcal{S}(-\pm_j, Ⓢ^j, A_1, \dots, A_{j-1}, \top/\bot, A_{j+1}, \dots, A_n, A_j)}\,(\text{cut})}{\mathcal{S}(\pm_{n+1}, Ⓢ, A_1, \dots, A_n, \top/\bot)}\,(\text{dp})}{\mathcal{S}(\pm_{n+1}, f, A_1, \dots, A_n, \top/\bot)}\,(\text{rew})$$

with appropriate but *not necessarily uniform* choices of $\top$ and $\bot$ for $\top/\bot$, thus generalising the other cases. The complication arise because we cannot cut on the *structure* I. A rigorous generalisation would use appropriate choices of $\tau_1(\mathrm{I})$, $\tau_2(\mathrm{I})$, $\tau'(\mathrm{I})$, $\tau''(\mathrm{I})$.

Thus the traces of f and $[f]$ are identical.

The remaining property is to ensure that the functions $[OP]$ obey the Abstract Law of Residuation in the appropriate places. That is we must show that:

$$\dfrac{S([f_j], [A_1], \dots, [A_{j-1}], [B], [A_{j+1}], \dots, [A_n], [A_j])}{S([f_{j+1}], [A_1], \dots, [A_{j-1}], [A_j], [B], [A_{j+2}], \dots, [A_n], [A_{j+1}])}$$

which reduces to

$$\dfrac{\mathcal{S}(f_j, A_1, \dots, A_{j-1}, B, A_{j+1}, \dots, A_n, A_j)}{\mathcal{S}(f_{j+1}, A_1, \dots, A_{j-1}, A_j, B, A_{j+2}, \dots, A_n, A_{j+1})}$$

The connectives f_j and f_{j+1} are logical connectives, not structural connectives, so we cannot immediately use the display postulates. But the proof shown below suffices, where "inv-rew" means the inverted version of the usual rule (rew), since all (rew) and (inv-rew) rules are invertible by Theorem 4.12:

$$\dfrac{\dfrac{\dfrac{\mathcal{S}(f_j, A_1, \dots, A_{j-1}, B, A_{j+1}, \dots, A_n, A_j)}{\mathcal{S}(F_j, A_1, \dots, A_{j-1}, B, A_{j+1}, \dots, A_n, A_j)}\,(\text{inv-rew})}{\mathcal{S}(F_{j+1}, A_1, \dots, A_{j-1}, A_j, B, A_{j+2}, \dots, A_n, A_{j+1})}\,(\text{dp})}{\mathcal{S}(f_{j+1}, A_1, \dots, A_{j-1}, A_j, B, A_{j+2}, \dots, A_n, A_{j+1})}\,(\text{rew})$$

Thus $[\mathcal{T}]$ is indeed a fully-founded partial gaggle with head $[f_0]$ and with traces identical to the original fully-founded partial gaggle $\mathcal{T}$.

A similar correspondence holds between the (dual) partial gaggle $\mathcal{T}^\Delta$ and $[\mathcal{T}^\Delta]$. ∎

5 Further Work

We now outline avenues for further work.

5.1 Conservative Extensions and the Fully Founded Restriction

Suppose a logic **L** has an an algebraic semantics in terms of **L**-algebras which are (not necessarily fully founded) partial gaggles. Any such partial gaggle can be extended to a fully-founded partial gaggle of operations OP, with OP possibly containing more operations (logical connectives) than originally specified in **L**. Does **L** have a cut-free display calculus?

The point here is that the Display Calculus δ**OP** which we would construct using the fully-founded partial gaggle has all dual connectives present, even though **L** itself may not be so expressive.

Now consider any proof of the endsequent $\mathrm{I} \vdash A_L$ where A_L is a formula of **L**. By Theorem 4.11, there is a cut-free proof of $\mathrm{I} \vdash A_L$. By C1, this proof involves the introduction rules for connectives from **L** only. Furthermore, by the limited display property [22], the required display postulates are the ones for the structural proxies of members of OP, not OP^Δ. Thus the operations from OP^Δ are harmless. However, some display moves will invariably involve the structural proxies for connectives from OP which are not in **L**. Is the logic of δ**OP** a conservative extension of **L** ?

That is, can we simply leave out the introduction rules for the extra connectives from δ**OP** and still obtain a display calculus for **L** ? Can we also display the "Residuals Without Residuation" of Dunn [14, Section 5] ?

5.2 Extra Axioms As Extra Structural Rules

The Display Calculus obtained so far contains only $p \vdash p$ as an axiom. Dunn [14] shows that we can add further logical axioms to a gaggle-theoretic set up. For example, he shows that the two logical axioms shown below on the left hand side are equivalent to each other [14] when residuation is present.

It can be shown that the rule at the right captures exactly these axioms where s is the structural analogue of $\otimes$. That is, if $f = \otimes$ then $F = s$.

$$a \to b \vdash (c \to a) \to (c \to b) \qquad\qquad \frac{s(s(X,Y),Z) \vdash W}{s(X,s(Y,Z)) \vdash W}$$
$$a \otimes (b \otimes c) \vdash (a \otimes b) \otimes c$$

So, even though the first axiom uses a connective $\to$ which is not expressible on the antecedent side as a structural connective, there is nevertheless an equivalent axiom which is so expressible.

This rule will have many other equivalents expressible in terms of r_1 and r_2, but we refrain from calculating them all and adding them to the system since they are derivable using the display postulates for the corresponding display set-up.

In this way we can capture most of the familiar structural rules like contraction, weakening, and associativity for any structural connective Ⓢ. The introduction rules for f then guarantee that f inherits all these properties just as $\otimes$ did above. We have to be careful to ensure that the new structural rules obey Belnap's conditions C1-C8; but see [25] and [20] for further details.

5.3 Display Elimination and Explicit Substitution

The conditions C1-C8 guarantee that the cut rule (cut) is eliminable from our display calculi. The only condition that really requires work is C8. But it is easy to see that the proof of C8 essentially involves displaying the j-th argument A_j, and then using a smaller cut to replace A_j by X_j. That is, the proof of C8 involves substituting an antecedent A_j with an X_j that satisfies $X_j \vdash A_j$, and substituting a succedent A_j with an X_j that satisfies $A_j \vdash X_j$. The display property gives us a disciplined way to do this substitution.

Therefore an alternative calculus is possible where we simply couch the introduction rules for $(f \vdash)$ and $(\vdash f)$ directly in terms of substitution into antecedent or succedent contexts. We give the rules in backward format; that is, we give them as moves from the conclusion to the premisses. We assume that f is n-ary.

If the trace output of f is $+$ then

$(\vdash f)$ In the conclusion sequent σ we can replace a succedent part occurrence of a formula $f(A_1, \ldots, A_n)$ by $F(A_1, \ldots, A_n)$, thereby giving a premiss sequent σ'.

$(f \vdash)$ If $f(A_1, \ldots, A_n)$ and $F(X_1, \ldots, X_n)$ occupy "corresponding" antecedent and succedent positions respectively, in the conclusion σ, then create n premisses from σ, where premiss σ_j is exactly like σ except that it

contains X_j instead of $F(X_1, \ldots, X_n)$, and A_j instead of $f(A_1, \ldots, A_n)$, if $\mathrm{tn}(f, j, +)$;

contains A_j instead of $F(X_1, \ldots, X_n)$, and X_j instead of $f(A_1, \ldots, A_n)$, if $\mathrm{tn}(f, j, -)$.

And dually if the trace output of f is $-$.

These conditions essentially build in the procedure of displaying and undisplaying involved in proving C8. The notion of "corresponding" needs to be formalised since it is the basis of a "connection" in the "connection method" [39], but the sequent systems of Došen [9] do this already to some extent. Thus they should also enjoy a general cut-elimination theorem based on the intuition that we can replace a succedent part occurrence of a formula by anything that is "greater than" that formula, and conversely, replace an antecedent part occurrence of a formula by anything that is "smaller than" that formula. Both "greater than" and "smaller than" are with respect to the underlying pre-order $\vdash$.

5.4 Applications

In a companion paper [22] we show how these results lead directly to a single, modular, cut-free Display Calculus which captures the intuitionistic and classical versions of all the well-known substructural logics like Lambek's (Sentential) Calculus, Linear Logic, Relevant Logic, BCK logic and Intuitionistic Logic purely by the addition or deletion of structural rules. Display calculi for fully-founded partial gaggles of unary functions like modalities [40], negations [15] and the converse operator [20] are also incorporated there. Moortgat [29] surveys various ways in which such operations can form interesting hybrid logics. Our methodology is applicable to all these logics.

5.5 Miscellaneous

It is clear that the sequent formulation using explicit substitution is very close to term-rewriting systems where such explicit substitution is done by specifying a set of rewrite rules which can be used in any context. It would be interesting to see where notions like the Church-Rosser property show themselves in the explicit substitution sequent framework.

Category theory underlies all of the previous considerations since the operations in a fully-founded partial gaggle naturally obey the n-ary versions of residuated pairs, dual residuated-pairs, Galois-connections, and dual Galois-connections. Further work is needed to clarify these connections. Since we have already given quite general proofs using only $\mathcal{S}$ and the trace of the connective in question, this should not be too difficult.

The direct connection between display calculi and gaggle theory should also allow us to obtain results on ternary correspondence theory [38]

Conjecture 5.1 *If an algebraic logic* **L** *has a display calculus then the class of* **L**-*algebras is quasi-equationally definable.*

Proof. Show that the Lindenbaum algebra formed from the provably equivalent formulae forms a quasi-variety. ∎

6 Conclusion

"What does it mean for a Gentzen system not to enjoy cut-elimination ?"

The work outlined here seems to give an answer. Failure of cut-elimination means that the equivalence relation $\dashv\vdash$ which usually leads to a Lindenbaum construction based upon a partial order $[A][\leq][B]$ is not a congruence because the set of provable sequents is not closed under substitution of antecedent part formulae by "smaller" formulae and substitution of succedent part formulae by "greater" formulae.

This is usually a symptom of inadequate cut rules.

7 Appendix

Appendix: Belnap's Conditions.

For every sequent rule Belnap [4, page 388] first defines the following notions: in an application *Inf* of a sequent rule (ρ), "constituents occurring as part of occurrences of structures assigned to structure-variables are defined to be **parameters** of *Inf* ; all other constituents are defined as **nonparametric**, including those assigned to formula-variables. Constituents occupying similar positions in occurrences of structures assigned to the same structure-variable are defined as **congruent** in *Inf*". The eight (actually seven) conditions shown below are from [25]:

(C1) Each formula which is a constituent of some premiss of a rule ρ is a subformula of some formula in the conclusion of ρ.

(C2) Congruent parameters are occurrences of the same structure.

(C3) Each parameter is congruent to at most one constituent in the conclusion. Equivalently, no two constituents of the conclusion are congruent to each other.

(C4) Congruent parameters are either all antecedent parts or all succedent parts of their respective sequent.

(C5) If a formula is non-parametric in the conclusion of a rule ρ, it is either the entire antecedent, or the entire succedent. Such a formula is called a **principal** formula.

(C6/7) Each rule is closed under simultaneous substitution of arbitrary structures for congruent parameters.

(C8) If there are inference rules ρ_1 and ρ_2 with respective conclusions $X \vdash P$ and $P \vdash Y$ with P principal in both inferences (in the sense of C5), and if (cut) is applied to yield $X \vdash Y$ then, either $X \vdash Y$ is identical to $X \vdash P$ or to $P \vdash Y$; or it is possible to pass from the premisses of ρ_1 and ρ_2 to $X \vdash Y$ by means of inferences falling under (cut) where the cut-formula is always a proper subformula of P. If P satisfies the "if" part of this condition it is known as a "matching principal constituent".

References

[1] A R Anderson and N D Belnap. *Entailment: The Logic of Relevance and Necessity*, volume 1. Princeton University Press, Princeton, USA, 1975.

[2] P B Andrews. Theorem proving via general matings. *JACM*, 28(2):193–214, 1981.

[3] J Barwise, D Gabbay, and C Hartonas. On the logic of information flow. *Bulletin of the IGPL*, 3(1):7–49, 1995.

[4] N D Belnap. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

[5] W Bibel. On matrices with connections. *JACM*, 28(4):633–645, 1981.

[6] R Binkley and R Clark. A cancellation algorithm for elementary logic. *Theoria*, 33:79–97, 1967.

[7] M D'Agostino and M Mondadori. The taming of the cut. classical refutations with analytic cut. *Journal of Logic and Computation*, 4:285–319, 1994.

[8] J Dawson and R Goré. A mechanised proof system for relation algebra using display logic. (submitted), Automated Reasoning Project, 1997.

[9] K Došen. Sequent systems and groupoid models, I. *Studia Logica*, 47:353–389, 1988.

[10] K Došen and P Schroeder-Heister, editors. *Substructural Logics*. Studies in Logic and Computation. Oxford University Press, 1993.

[11] J M Dunn. A "Gentzen system" for positive relevant implication. *Journal of Symbolic Logic*, 38:356–357, 1973.

[12] J M Dunn. Gaggle theory: An abstraction of Galois connections and residuation with applications to negation and various logical operations. In *JELIA 1990: Proceedings of the European Workshop on Logics in Artificial Intelligence*, volume LNCS 478. Springer, 1991.

[13] J M Dunn. Gaggle theory applied to modal, intuitionistic, and relevance logics. In I Max and W Stelzner, editors, *Logik und Mathematik: Frege-Kolloquium Jena*, pages 335–368. de Gruyter, 1993.

[14] J M Dunn. Partial gaggles applied to logics with restricted structural rules. In K Došen and P Schroeder-Heister, editors, *Substructural Logics*, Studies in Logic and Computation, pages 63–108. Oxford University Press, 1993.

[15] J M Dunn. Perp and star: Two treatments of negation. In J Tomberlin, editor, *Philosophy of Language and Logic*, volume 7 of *Philosophical Perspectives*, pages 331–357. Ridgeview Publishing Company, Atascadero, California, USA, 1993.

[16] J-Y Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[17] R Goré. Gaggles, Gentzen and Galois: A proof theory for non-classical logics. Technical Report TR-SRS-1-95, Automated Reasoning Project, Australian National University, Australia, 1995. http://arp.anu.edu.au/~rpg.

[18] R Goré. A uniform display system for intuitionistic and dual intuitionistic logic. Technical Report TR-ARP-6-95, Automated Reasoning Project, Austalian National University, 1995.

[19] R Goré. On the completeness of classical modal display logic. In H Wansing, editor, *Proof Theory of Modal Logic*, volume 2 of *Applied Logic*, pages 137–140. Kluwer, 1996.

[20] R Goré. Cut-free display calculi for relation algebras. In D van Dalen and M Bezem, editors, *CSL96: Selected Papers of the Annual Conference of the European Association for Computer Science Logic*, volume LNCS 1258, pages 198–210. Springer, 1997.

[21] R Goré. Gaggles, Gentzen and Galois: Cut-free display calculi and relational semantics for algebraizable logics. Technical Report TR-ARP-07-97, Automated Reasoning Project, Australian National University, Australia, 1997.

[22] R Goré. Substructural logics on display. *Logic Journal of the Interest Group in Pure and Applied Logic*, to appear, 1998.

[23] U Hoehle. Commutative, residuated l-monoids. In U Hoehle and E P Klement, editors, *Non-classical logics and their applications to fuzzy subsets*, pages 53–106. Dordrecht, Kluwer, 1995.

[24] B Jónsson and A Tarski. Boolean algebras with operators. *American Journal of Mathematics*, 73-74(891-939):127–162, 1951-52.

[25] M Kracht. Power and weakness of the modal display calculus. In H Wansing, editor, *Proof Theory of Modal Logics*, pages 92–121. Kluwer, 1996.

[26] J Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

[27] P D Lincoln and N Shankar. Proof search in first-order linear logic and other cut-free sequent calculi. In *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science*, 1994.

[28] G Mints. Cut-elimination theorem in relevant logics. *Journal of Soviet Mathematics*, 6:422–428, 1976. Original in Russian 1972.

[29] M Moortgat. Categorial type logics. In van Benthem and A ter Meulen, editors, *Handbook of Logic and Language*, chapter 2. Elsevier, To Appear.

[30] H Ono. Personal communication, 1997.

[31] H Ono and Y Komori. Logics without the contraction rule. *Journal of Symbolic Logic*, 50:169–201, 1985.

[32] L Paulson. *Isabelle: A Generic Theorem Prover*, volume LNCS 828. Springer-Verlag, 1994.

[33] J Pavelka. On fuzzy logic II. *Zeitschrift fuer Mathematische Logik*, 25:119–134, 1979.

[34] V Pratt. Action logic and pure induction. In *JELIA90: Proc. European Workshop on Logics in Artificial Intelligence*, volume LNCS 478, pages 97–120. Springer-Verlag, 1991.

[35] G Restall. Display logic and gaggle theory. *Reports on Mathematical Logic*, 29:133–146, 1995, published in 1996.

[36] G Restall. Displaying and deciding substructural logics I: logics with contraposition. *Journal of Philosophical Logic*, 1998 (to appear).

[37] M. E. Szabo, editor. *The collected papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.

[38] J van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 167–247. D. Reidel, 1984.

[39] L A Wallen. *Automated Deduction in Nonclassical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics*. MIT Press, 1989.

[40] H Wansing. Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation*, 4:125–142, 1994.

Received March 10, 1998