Which Structural Rules Admit Cut Elimination?  
— An Algebraic Criterion (Excerpt)

Kazushige Terui  National Institute of Informatics
terui@nii.ac.jp

This is an excerpt of our recent paper [Ter05]. See [Ter05] for the details.

1 Introduction

Gentzen’s original sequent calculus contains three structural rules:

\[
\begin{align*}
\text{Exchange:} & \quad \frac{\Gamma, \alpha, \beta, \Delta \Rightarrow \gamma}{\Gamma, \beta, \alpha, \Delta \Rightarrow \gamma} e \\
\text{Weakening:} & \quad \frac{\Gamma, \Delta \Rightarrow \gamma}{\Gamma, \alpha, \Delta \Rightarrow \gamma} w \\
\text{Contraction:} & \quad \frac{\Gamma, \alpha, \alpha, \Delta \Rightarrow \gamma}{\Gamma, \alpha, \Delta \Rightarrow \gamma} c
\end{align*}
\]

where \( \alpha, \beta \) and \( \gamma \) stand for formulas and \( \Gamma \) and \( \Delta \) stand for sequences of formulas (we only consider intuitionistic sequents in this paper). In addition, one can also consider other non-standard structural rules such as:

\[
\begin{align*}
\text{Expansion (cf. [vB91]):} & \quad \frac{\Gamma, \alpha, \Delta \Rightarrow \gamma}{\Gamma, \alpha, \alpha, \Delta \Rightarrow \gamma} \exp \\
\text{Mingle (cf. [OM64]):} & \quad \frac{\Gamma, \Sigma, \Delta \Rightarrow \gamma}{\Gamma, \Theta, \Delta \Rightarrow \gamma} \min
\end{align*}
\]

(See also [HOS94, Kam02] for a detailed account.) Among them, some are harmless but others cause failure of cut elimination. In fact, the availability of cut elimination is very sensitive to the choice of structural rules:

- In general, sequent calculi with Contraction but without Exchange do not enjoy cut elimination. One way to recover cut elimination is to generalize Contraction to the one for sequences of formulas:

\[
\frac{\Gamma, \Sigma, \Sigma, \Delta \Rightarrow \gamma}{\Gamma, \Sigma, \Delta \Rightarrow \gamma} \text{ seq-c}
\]
• Expansion and Mingle are derivable from each other. However, Mingle admits cut elimination whereas Expansion does not.

In view of these intricacies, it is natural to look for some general criteria for a set of structural rules to admit cut elimination. The aim of this paper is to give such a criterion for cut elimination by using algebraic semantics.

We consider (the 0-free fragment of) full Lambek calculus (FL⁺, [Ono90, Ono94, Ono03]), i.e., intuitionistic logic without any structural rules, as our basic framework. We then introduce structural rules on FL⁺ in a general format. Residuated lattices are the algebraic structures corresponding to FL⁺ (see [JT02, Ono03]). In this setting, we introduce a criterion, called the propagation property, that can be stated both in syntactic and algebraic terminologies. It is a refinement of Girard’s naturality test, which appears in an informal discussion in Appendix C.4 of [Gir99].

We then show that, for any set \( \mathcal{R} \) of structural rules, the cut elimination theorem holds for FL⁺ enriched with \( \mathcal{R} \) if and only if \( \mathcal{R} \) satisfies the propagation property. To show the ‘if’ direction, the phase structures ([Abr90, Tro92, Ono94]) as well as Okada’s cut elimination technique [Oka96, Oka99, Oka02] are essentially used.

As an application, we show that any set \( \mathcal{R} \) of structural rules can be "completed" into another set \( \mathcal{R}^* \), so that the cut elimination theorem holds for FL⁺ enriched with \( \mathcal{R}^* \), while the provability remains the same.

2 Full Lambek Calculus and Structural Rules

The formulas of FL⁺ are built from propositional variables \( a, b, c, \ldots \) and constants \( 1 \) (unit), \( \top \) (true) and \( \bot \) (false) by using binary logical connectives \( \cdot \) (fusion), \( \backslash \) (right implication), \( / \) (left implication), \( \wedge \) (conjunction) and \( \vee \) (disjunction). The set of formulas is denoted by \( \mathcal{F} \). Small Greek letters \( \alpha, \beta, \ldots \) range over \( \mathcal{F} \). For simplicity, we do not consider negation nor \( 0 \) in this paper. We use \( \Rightarrow \) as synonym for \( \backslash \).

A sequent of FL⁺ is of the form \( \alpha_1, \ldots, \alpha_n \Rightarrow \beta \). Here, formulas \( \alpha_1, \ldots, \alpha_n \) are called antecedents and \( \beta \) is called a succedent. In the sequel, \( \Gamma, \Delta, \ldots \) stand for finite sequences of formulas, and \( \emptyset \) stands for the empty sequence.

A sequent \( \Gamma \Rightarrow \alpha \) is said to be provable in FL⁺ if it is derivable by using the inference rules in Figure 1. A formula \( \alpha \) is provable if the sequent \( \Rightarrow \alpha \) is provable. Given a (possibly infinite) set \( \Omega \) of sequents, a sequent \( \Gamma \Rightarrow \gamma \) is said to be derivable from \( \Omega \) if \( \Gamma \Rightarrow \gamma \) is provable in FL⁺ enriched with the additional axioms \( \Omega \) (see [Ono94, Ono03] for more information).
### 1: Inference Rules of FL⁺

When it is necessary to indicate variables \(a_1, \ldots, a_m\) that might possibly occur in a formula \(\alpha\), we shall use the notation \(\alpha[a_1, \ldots, a_m]\), or \(\alpha[\vec{a}]\) for short. The formula obtained from \(\alpha[a_1, \ldots, a_m]\) by substituting \(\beta_i\) for each \(a_i\) is denoted by \(\alpha[\beta_1, \ldots, \beta_m]\), or \(\alpha[\vec{\beta}]\). Similar notation is used for sequences of formulas (and structural rules introduced below).

For \(\Sigma \equiv \alpha_1, \ldots, \alpha_n (n \geq 1)\), we define

\[
\begin{align*}
* \Sigma & \equiv \alpha_1 \ldots \alpha_n, \\
\lor \Sigma & \equiv \alpha_1 \lor \ldots \lor \alpha_n.
\end{align*}
\]

**FL⁺** is entirely free from structural rules. Various systems of so-called substructural logics are obtained by enriching it with a suitable set of structural rules. Formally, a *structural rule* \(R\) is an \(n + 1\) tuple \((\Theta_1; \ldots; \Theta_n \triangleright \Theta_0)\), where \(n \geq 1\) and each \(\Theta_i\) is a finite sequence of variables, that satisfies the following condition:

*(*) any variable occurring in \(\Theta_1, \ldots, \Theta_n\) also occurs in \(\Theta_0\).

The last condition will be referred to as the *non-erasing condition*.

Let \(R[\vec{a}]\) be a structural rule \((\Theta_1[\vec{a}]; \ldots; \Theta_n[\vec{a}] \triangleright \Theta_0[\vec{a}])\), and \(\vec{\beta}\) be a sequence of formulas. Then the result of substitution \(R[\vec{\beta}] = (\Theta_1[\vec{\beta}]; \ldots; \Theta_n[\vec{\beta}] \triangleright \Theta_0[\vec{\beta}]\)
$\Theta_0[\vec{\beta}]$, is called an instance of $R$. When $\Phi$ is a set of formulas and formulas $\vec{\beta}$ belong to $\Phi$, $R[\vec{\beta}]$ is called a $\Phi$-instance. Each instance $R[\vec{\beta}]$ codifies an inference scheme of the form:

$$
\frac{\Gamma, \Theta_1[\vec{\beta}], \Delta \Rightarrow \gamma \cdots \Gamma, \Theta_n[\vec{\beta}], \Delta \Rightarrow \gamma}{\Gamma, \Theta_0[\vec{\beta}], \Delta \Rightarrow \gamma}
$$

with $\Gamma$, $\Delta$ and $\gamma$ arbitrary.

For example, the structural rules mentioned in the introduction can be formally specified as follows:

- **e**: $(a, b \triangleright b, a)$
- **w**: $(\emptyset \triangleright a)$
- **c**: $(a, a \triangleright a)$
- **exp**: $(a \triangleright a, a)$
- **min**: $\{(a_1, \ldots, a_k; b_1, \ldots, b_l \triangleright a_1, \ldots, a_k, b_1, \ldots, b_l) | 1 \leq k, 1 \leq l\}$
- **seq-c**: $\{(a_1, \ldots, a_k, a_1, \ldots, a_k \triangleright a_1, \ldots, a_k) | 1 \leq k\}$

Notice that **min** and **seq-c** are specified by a countable set of structural rules.

Given a set $\mathcal{R}$ of structural rules, the system $\text{FL}^+(\mathcal{R})$ is defined to be $\text{FL}^+$ enriched with all instances of the additional structural rules $\mathcal{R}$. For instance, $\text{FL}^+\{\text{e}\}$ amounts to $\text{FL}^+_{\text{e}}$ (intuitionistic linear logic without modality), while $\text{FL}^+\{\text{e, w, c}\}$ is nothing but intuitionistic logic.

Due to the non-erasing condition, our structural rules satisfy the following property: any formula occurring in the upper sequents of a structural rule also occurs in the lower sequent. It follows that the cut elimination theorem always implies the subformula property.

Given a sequent, the *positive subformulas* and *negative subformulas* are defined as usual. We then have:

**Lemma 2.1** Let $\mathcal{R}$ be a set of structural rules. Suppose that $\text{FL}^+(\mathcal{R})$ enjoys cut elimination. Then it satisfies the (polarized) subformula property: if a sequent $\Gamma \Rightarrow \alpha$ is provable in $\text{FL}^+(\mathcal{R})$, then it has a derivation $\pi$ in which only subformulas of $\Gamma \Rightarrow \alpha$ occur. Moreover, any antecedent (succedent, resp.) formula of a sequent in $\pi$ is a negative (positive, resp.) subformula of $\Gamma \Rightarrow \alpha$. 
To study the properties of structural rules, it is convenient to represent them as formulas. Given a structural rule $R = (\Theta_1, \ldots, \Theta_n \triangleright \Theta_0)$, define its formula representation $\hat{R}$ by

$$\hat{R} \equiv *\Theta_0 \rightarrow (*\Theta_1 \lor \cdots \lor *\Theta_n).$$

For instance, $\check{e} \equiv b \cdot a \rightarrow a \cdot b$ and $\check{w} \equiv a \rightarrow 1$. The formula representation of $\text{min}_1 = (a; b \triangleright a, b)$ is $a \cdot \bar{a} \rightarrow (a \cdot b) \cdot (a \cdot b)$. If $R$ is of the form $R[a_1, \ldots, a_m]$ and $\alpha_1, \ldots, \alpha_m$ belong to a set $\Phi$ of formulas, then $\hat{R}[\alpha_1, \ldots, \alpha_m]$ is called a $\Phi$-instance of $\hat{R}$. When $\mathcal{R}$ is a set of structural rules, $\hat{\mathcal{R}}$ denotes the set $\{\hat{R} \mid R \in \mathcal{R}\}$.

As expected, there is an instance-wise correspondence between structural rules and their formula representations:

**Lemma 2.2** Let $R[\vec{a}]$ be a structural rule. Then an instance $R[\vec{\alpha}]$ is derivable from $\hat{R}[\vec{\alpha}]$ and vice versa.

### 3 Syntactic Propagation

Let us now introduce a syntactic version of the propagation property. To motivate the notion, consider the contrast between $\text{FL}^+ (\{c\})$ and $\text{FL}^+ (\text{seq-c})$. As is mentioned in the introduction, the former does not enjoy cut elimination. For instance, the cut below cannot be eliminated:

$$\frac{\alpha \Rightarrow \alpha \quad \beta \Rightarrow \beta}{\alpha \cdot \beta \Rightarrow \alpha \cdot \beta} \quad \alpha \cdot \beta \Rightarrow (\alpha \cdot \beta) \cdot (\alpha \cdot \beta)$$

On the other hand, if $c$ is generalized to seq-c, the cut can be easily eliminated:

$$\frac{\alpha \Rightarrow \alpha \quad \beta \Rightarrow \beta}{\alpha \cdot \beta \Rightarrow \alpha \cdot \beta} \quad \alpha \cdot \beta \Rightarrow (\alpha \cdot \beta) \cdot (\alpha \cdot \beta)$$

Now our question is this: what is the essential difference between $c$ and seq-c? A distinctive feature of seq-c is that it propagates from variable instances to fusion instances. Namely, a fusion instance $(a \cdot b, a \cdot b \triangleright a \cdot b)$ is derivable from a variable instance $(a, b, a, b \triangleright a, b)$ as follows:
\[
\frac{\Gamma, a \cdot b, a \cdot b, \Delta \Rightarrow \gamma}{\Gamma, a, b, a, b, \Delta \Rightarrow \gamma} \quad \text{seq-c}
\]

(Pedantically speaking, an instance \(R[\vec{a}] = (\Theta_1[\vec{a}]; \ldots ; \Theta_n[\vec{a}] \triangleright \Theta_0[\vec{a}])\) is *derivable* from a set \(\Omega\) of instances of some structural rules if for arbitrary \(\Gamma, \Delta\) and \(\gamma\), the sequent \(\Gamma, \Theta_i[\vec{a}], \Delta \Rightarrow C\) is derivable from the sequents \(\Gamma, \Theta_i[\vec{a}], \Delta \Rightarrow \gamma\) for \(1 \leq i \leq n\) in \(\mathbf{FL}^+\) enriched with the rule instances \(\Omega\).

In contrast, one can observe that \(c\) does not propagate to fusion instances.

Next, consider the contrast between \(\mathbf{FL}^+ (\{\exp\})\) and \(\mathbf{FL}^+ (\min)\). The former does not enjoy cut elimination, as witnessed by:

\[
\begin{align*}
\beta \Rightarrow \beta & \quad & \frac{\alpha \Rightarrow \alpha \vee \beta}{\alpha \Rightarrow \alpha} & \quad & \frac{\alpha \Rightarrow \alpha \vee \beta}{\alpha \Rightarrow \alpha} \\
\alpha \Rightarrow \alpha \vee \beta & \quad & \frac{\alpha \vee \beta \Rightarrow \alpha \vee \beta}{\alpha \vee \beta} & \quad & \frac{\alpha \vee \beta \Rightarrow \alpha \vee \beta}{\alpha \vee \beta}
\end{align*}
\]

\(\text{exp} \quad \text{cut} \quad \text{cut} \)

Notice that one cannot obtain a cut-free proof even if \(\exp\) is generalized to a sequence version as above. On the other hand, when \(\exp\) is replaced with \(\min\), a cut-free proof is obtained:

\[
\begin{align*}
\alpha \Rightarrow \alpha & \quad & \frac{\beta \Rightarrow \beta}{\beta \Rightarrow \beta} \\
\alpha \Rightarrow \alpha \vee \beta & \quad & \frac{\alpha \Rightarrow \alpha \vee \beta}{\alpha} & \quad & \frac{\beta \Rightarrow \alpha \vee \beta}{\beta}
\end{align*}
\]

\(\min\)

Therefore, we may again ask what is the *essential* difference between \(\exp\) and \(\min\). This time, our answer is that \(\min\) *propagates from variable instances to disjunction instances*. Namely, a disjunction instance \((a_1 \vee b_1 ; a_2 \vee b_2 \triangleright a_1 \vee b_1, a_2 \vee b_2)\) is derivable from variable instances \((a_1 ; a_2 \triangleright a_1, a_2)\), \((a_1 ; b_2 \triangleright a_2, b_2)\), \((b_1 ; a_2 \triangleright b_1, a_2)\) and \((b_1 ; b_2 \triangleright b_1, b_2)\) as follows:

\[
\begin{align*}
\Gamma, a_1 \vee b_1, \Delta \Rightarrow \gamma & \quad & \Gamma, a_2 \vee b_2, \Delta \Rightarrow \gamma & \quad & \Gamma, a_1 \vee b_1, \Delta \Rightarrow \gamma & \quad & \Gamma, a_2 \vee b_2, \Delta \Rightarrow \gamma \\
\Gamma, a_1, \Delta \Rightarrow \gamma & \quad & \Gamma, a_2, \Delta \Rightarrow \gamma & \quad & \min & \quad & \min \\
\Gamma, a_1, a_2, \Delta \Rightarrow \gamma & \quad & \Gamma, b_1, \Delta \Rightarrow \gamma & \quad & \Gamma, b_1, \Delta \Rightarrow \gamma & \quad & \Gamma, b_2, \Delta \Rightarrow \gamma \\
\Gamma, a_1 \vee b_1, a_2 \vee b_2, \Delta \Rightarrow \gamma & \quad & \min & \quad & \min & \quad & \min
\end{align*}
\]

In contrast, \(\exp\) does not propagate to disjunction instances.

These observations bring us to the following definition. A set \(\mathcal{R}\) of structural rules satisfies the *syntactic propagation property* if the following holds:

- For every \(R[a_1, \ldots , a_m] \in \mathcal{R}\) and every \(\Sigma_1, \ldots , \Sigma_m\), where each \(\Sigma_i\) is a sequence of variables, both \(R[\Sigma_1, \ldots , \Sigma_m]\) and \(R[\Sigma_1, \ldots , \Sigma_m]\)
are derivable from the \( \Phi \)-instances of the structural rules in \( \mathcal{R} \), where \( \Phi \) is the set of variables occurring in \( \Sigma_1, \ldots, \Sigma_m \).

In view of Lemma 2.2, this is equivalent to say that

- the formulas \( \hat{\mathcal{R}}[\star \Sigma_1, \ldots, \star \Sigma_m] \) and \( \hat{\mathcal{R}}[\vee \Sigma_1, \ldots, \vee \Sigma_m] \) are deducible from the \( \Phi \)-instances of the formulas in \( \mathcal{R} \).

The syntactic propagation property does not explicitly refer to, but is actually closely related to cut elimination. In fact, we have:

**Proposition 3.1** Let \( \mathcal{R} \) be a set of structural rules. If \( \text{FL}^+(\mathcal{R}) \) enjoys cut elimination, then \( \mathcal{R} \) satisfies the syntactic propagation property.

### 4 Residuated lattices and semantic propagation

An algebra \( \mathbf{P} = \langle P, \wedge, \vee, \cdot, \backslash, /, 1 \rangle \) is called a (bounded) residuated lattice if

1. \( \langle P, \wedge, \vee \rangle \) is a lattice with the greatest element \( \top \) and the least element \( \bot \).
2. \( \langle P, \cdot, 1 \rangle \) is a monoid.
3. The operations \( \backslash \) and \( / \) are right and left residuals of \( \cdot \). Namely, for any \( x, y, z \in P \),
   \[
   x \cdot y \leq z \iff x \leq z/y \iff y \leq x \backslash z.
   \]

(See [JT02, Ono03] for general introductions to residuated lattices.)

A *valuation* \( f \) on \( \mathbf{P} \) maps each variable to an element of \( P \). Given a set \( X \subseteq P \), \( f \) is called an \( X \)-valuation if the range is a subset of \( X \). As usual, \( f \) can be extended to a map from the formulas \( \mathcal{F} \) to \( P \) as follows:

\[
\begin{align*}
   f(\top) &= \top & \text{for } \top \in \{ \top, \bot, 1 \}, \\
   f(\alpha \star \beta) &= f(\alpha) \star f(\beta) & \text{for } \star \in \{ \wedge, \vee, \cdot, \backslash, / \}.
\end{align*}
\]

A formula \( \alpha \) is said to be *true* under valuation \( f \) in \( \mathbf{P} \) if \( f(\alpha) \geq 1 \). In particular, \( \alpha \rightarrow \beta \), i.e., \( \alpha \backslash \beta \) is true iff \( f(\alpha) \leq f(\beta) \). A formula \( \alpha \) is *valid* (\( X \)-valid, resp.) in \( \mathbf{P} \) if it is true under all valuations (\( X \)-valuations, resp.) on \( \mathbf{P} \).

The residuated lattices are algebraic models of \( \text{FL}^+ \). In particular, the following strong form of soundness holds for them:
Lemma 4.1 Let $\mathbf{P}$ be a residuated lattice and $f$ be a valuation on it. If $\alpha$ is deducible from $\Phi$ and all formulas in $\Phi$ are true under $f$ in $\mathbf{P}$, then $\alpha$ is also true under $f$.

Given a set $\mathcal{R}$ of structural rules, an $\mathcal{R}$-residuated lattice is a residuated lattice in which all formulas in $\mathcal{R}$ are valid. By the previous lemma, any formula provable in $\mathbf{F} \mathbf{L}^+(\mathcal{R})$ is valid in all $\mathcal{R}$-residuated lattices.

Coming back to the residuated lattices in general, we may observe that the monoid multiplication $\cdot$ is continuous in the following sense:

Lemma 4.2 Let $q_0, \ldots, q_m \in P$ and let

$$\delta(p_1, \ldots, p_m) = q_0 \cdot p_1 \cdot q_1 \cdot p_2 \cdot \cdots \cdot q_m \cdot p_m,$$

for any $p_1, \ldots, p_m \in P$. Let also $\tilde{\delta}(p) = \delta(p, \ldots, p)$. Suppose that $X$ is a subset of $P$ for which $\bigvee X$ exists. We then have:

$$\tilde{\delta}(\bigvee X) = \bigvee_{Y \subseteq_{fin} X} \tilde{\delta}(\bigvee Y),$$

where $Y \subseteq_{fin} X$ holds iff $Y$ is a finite subset of $X$.

Given $X \subseteq P$, the multiplication closure $\prod(X)$, the join closure $\mathbf{J}(X)$ and the finite join closure $\mathbf{I}_{fin}(X)$ are defined by

$$\prod(X) = \{p_1 \cdots p_n \mid n \geq 0, p_1, \ldots, p_n \in X\},$$

$$\mathbf{J}(X) = \{\bigvee Y \mid Y \subseteq X, \bigvee Y \text{ exists}\},$$

$$\mathbf{I}_{fin}(X) = \{\bigvee Y \mid Y \subseteq_{fin} X\}.$$

A set $\mathcal{R}$ of structural rules satisfies the semantic propagation property if for any residuated lattice $\mathbf{P}$ and $X \subseteq P$, the following holds:

- if all formulas in $\mathcal{R}$ are $X$-valid, then they are also $\mathbf{I}_{fin}(\prod(X))$-valid.

We have:

Proposition 4.3 If a set $\mathcal{R}$ of structural rules satisfies the syntactic propagation property, it also satisfies the semantic propagation property.
5 Phase structures and semantic cut elimination

We now introduce a special class of residuated lattices, sometimes called (intuitionistic noncommutative) phase structures (see [Abr90, Tro92, Ono94]). Let $M = \langle M, \cdot, 1 \rangle$ be a monoid. Denote the powerset of $M$ by $\wp(M)$, and define for $X, Y \in \wp(M)$,

$$X \cdot Y = \{x \cdot y \mid x \in X, y \in Y\}.$$ 

A function $C : \wp(M) \to \wp(M)$ is said to be a closure operator on $\wp(M)$ if for all $X, Y \in \wp(M)$,

1. $X \subseteq C(X)$,
2. $C(C(X)) \subseteq C(X)$,
3. $X \subseteq Y$ implies $C(X) \subseteq C(Y)$,
4. $C(X) \cdot C(Y) \subseteq C(X \cdot Y)$.

A set $X \in \wp(M)$ is closed if $X = C(X)$. The set of all closed sets in $\wp(M)$ is denoted by $C_M$. Define for any closed sets $X, Y \in C_M$ and for any family $\mathcal{X}$ of closed sets,

$$X \cup_C Y = C(X \cup Y),$$
$$\bigcup_C \mathcal{X} = C(\bigcup \mathcal{X}),$$
$$X \bullet_C Y = C(X \bullet Y),$$
$$X \parallel Y = \{y \mid \forall x \in X, x \cdot y \in Y\},$$
$$Y /\!/ X = \{y \mid \forall x \in X, y \cdot x \in Y\}.$$

We then have:

**Lemma 5.1** If $M$ is a monoid and $C$ is a closure operator on $\wp(M)$, then the algebra

$$C_M = \langle C_M, \cap, \cup_C, \bullet_C, \parallel, \bigcup_C, \{1\} \rangle,$$

is a complete residuated lattice with infinite join $\bigcup_C$.

In every phase structure, the following hold:

1. $C(\{x \cdot y\}) = C(\{x\}) \bullet_C C(\{y\})$ for any $x, y \in M$,
2. $C(X) = \bigcup_C x \in X C(\{x\})$ for any $X \subseteq M$. 
As a consequence, phase structures satisfy the following remarkable property which plays a key role in connecting the semantic propagation property to cut elimination:

**Lemma 5.2** Suppose that $M$ is finitely generated by a set $A$, i.e., any element $x$ of $M$ can be written as $y_1 \cdots y_n$ for some $y_1, \ldots, y_n \in A$. Let $C'_A = \{C(\{y\}) \mid y \in A\}$. Then we have $C_M = \prod(\prod(C'_A))$.

We now describe a specific construction of a phase structure due to [Oka96, Oka99] (and slightly remedied by [OT99]), which is quite useful for proving the cut elimination theorem. (See also [BOJ01], where Okada’s construction is reformulated as algebraic quasi-completion and quasi-embedding.)

Let $\mathcal{F}^*$ be the free monoid generated by the formulas $\mathcal{F}$ of $\text{FL}^+$; the elements of $\mathcal{F}^*$ are sequences of formulas, the monoid multiplication is concatenation, and the unit element is the empty sequence $\emptyset$.

Let us fix a set $\mathcal{R}$ of structural rules. The operator $C$ is defined on the basis of cut-free provability in $\text{FL}^+ (\mathcal{R})$:

\[ [\Gamma, \Delta \Rightarrow \gamma] = \{ \Sigma \mid \Gamma, \Sigma, \Delta \Rightarrow \gamma \text{ is cut-free provable in } \text{FL}^+ (\mathcal{R}) \}, \]

\[ \mathcal{D} = \{ [\Gamma, \Delta \Rightarrow \gamma] \mid \Gamma, \Delta, \gamma \text{ arbitrary} \}, \]

\[ C(X) = \bigcap_{X \subseteq Y \in \mathcal{D}} Y. \]

Then one can show that $C$ is indeed a closure operator on $\varphi (\mathcal{F}^*)$ (for an arbitrary $\mathcal{R}$). Hence by Lemma 5.1, the algebra

\[ \mathcal{C}_{\mathcal{F}^*} = \langle \mathcal{C}_{\mathcal{F}^*}, \cap, \cup_C, \bullet_C, \setminus, \bigcup, C(\{\emptyset\}) \rangle \]

is a residuated lattice.

Let $f_0$ be a valuation on $\mathcal{C}_{\mathcal{F}^*}$ defined by $f_0(a) = C(\{a\})$. In this setting, we have Okada’s lemma:

**Lemma 5.3** For every formula $\alpha$, $\alpha \in f_0(\alpha) \subseteq [\_ \Rightarrow \alpha]$. In particular, for every sequent $\Gamma \Rightarrow \alpha$, if $(*\Gamma) \rightarrow \alpha$ is true under $f_0$, then $\Gamma \Rightarrow \alpha$ is cut-free provable in $\text{FL}^+ (\mathcal{R})$.

It is worth noting that Okada’s lemma holds independently of which structural rules $\mathcal{R}$ we adopt. It only concerns with the properties of logical inference rules. What depends on the choice of $\mathcal{R}$ is the following:

**Lemma 5.4** If $\mathcal{R}$ satisfies the semantic propagation property, then $\mathcal{C}_{\mathcal{F}^*}$ is an $\mathcal{R}$-residuated lattice.
We have thus arrived at:

**Proposition 5.5** If $\mathcal{R}$ satisfies the semantic propagation property, then $\mathsf{FL}^+(\mathcal{R})$ enjoys cut elimination.

By putting Propositions 3.1, 4.3 and 5.5 together, we obtain our main theorem:

**Theorem 5.6** Let $\mathcal{R}$ be a set of structural rules. Then the following are equivalent:

1. $\mathsf{FL}^+(\mathcal{R})$ enjoys cut elimination.
2. $\mathcal{R}$ satisfies the syntactic propagation property.
3. $\mathcal{R}$ satisfies the semantic propagation property.

## 6 Completion of Structural Rules

Recall that Contraction $c$ can be generalized to its sequence version $\text{seq-c}$ without changing provability so that the cut elimination theorem holds for $\mathsf{FL}^+(\text{seq-c})$. We say that $c$ can be completed into $\text{seq-c}$. Likewise, Expansion $\exp$ can be completed into Mingle $\min$. The completion techniques implicitly used there are by no means specific to $c$ and $\exp$. In fact, we can show that an arbitrary set of structural rules can be completed by using those techniques.

**Theorem 6.1** Given a set $\mathcal{R}$ of structural rules, one can obtain another set $\mathcal{R}^*$ of structural rules such that the following hold.

- $\mathsf{FL}^+(\mathcal{R})$ and $\mathsf{FL}^+(\mathcal{R}^*)$ are equivalent.
- $\mathcal{R}^*$ satisfies the syntactic propagation property. Hence $\mathsf{FL}^+(\mathcal{R}^*)$ enjoys cut-elimination.

To prove this, we use our characterization of cut elimination by the syntactic propagation property.

**Acknowledgements.** We are indebted to Jean-Yves Girard, who suggested to the author a possible linkage between his test and cut elimination in 1999, and thus motivated the current work. Our thanks are also due to Shun’ichi Amano, Nicolas Galatos, Makoto Kanazawa, Hiroakira Ono, Takafumi Saku-rai, Kentaro Sato and Hiroki Takamura for various comments and stimulating discussions.
参考文献


