Normal Form Theorem Proving for the Lambek Calculus[1]

Mark Hepple
Centre for Cognitive Science, University of Edinburgh,
2 Buccleuch Place, Edinburgh, Scotland.
e-mail: mark@uk.ac.ed.epistemi

Abstract

The possibility of ***multiple equivalent proofs*** presents a problem for efficient parsing of a number of flexible categorial grammar (CG) frameworks. In this paper I outline a normal form system for a sequent formulation of the product-free associative Lambek Calculus. This leads to a simple parsing approach that yields only normal form proofs. This approach is both ***safe*** in that all distinct readings for a sentence will be returned, and ***optimal*** in that there is only one normal form proof yielding each distinct meaning.

1 The Lambek Calculus

The (product-free) Lambek Calculus (Lambek, 1958) is a highly flexible CG framework that can be given a Gentzen-style sequent formulation with the following rules:[2][3]

(1) Axiom: $\mathrm{x}{:}f \Rightarrow \mathrm{x}{:}f$

Right rules:

$$\frac{\Gamma,\ \mathrm{y}{:}i \Rightarrow \mathrm{x}{:}f}{\Gamma \Rightarrow \mathrm{x/y}{:}\lambda i.f}\,[/\mathrm{R}] \qquad \frac{\mathrm{y}{:}i,\ \Gamma \Rightarrow \mathrm{x}{:}f}{\Gamma \Rightarrow \mathrm{x\backslash y}{:}\lambda i.f}\,[\backslash\mathrm{R}]$$

Left rules:

$$\frac{\Delta \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{x}{:}fg,\ \Lambda \Rightarrow \mathrm{z}{:}h}{\Gamma,\ \mathrm{x/y}{:}f,\ \Delta,\ \Lambda \Rightarrow \mathrm{z}{:}h}\,[/\mathrm{L}]$$

$$\frac{\Delta \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{x}{:}fg,\ \Lambda \Rightarrow \mathrm{z}{:}h}{\Gamma,\ \Delta,\ \mathrm{x\backslash y}{:}f,\ \Lambda \Rightarrow \mathrm{z}{:}h}\,[\backslash\mathrm{L}]$$

Cut rule:

$$\frac{\Delta \Rightarrow \mathrm{x}{:}f \qquad \Gamma,\ \mathrm{x}{:}f,\ \Lambda \Rightarrow \mathrm{y}{:}g}{\Gamma,\ \Delta,\ \Lambda \Rightarrow \mathrm{y}{:}g}\,[\mathrm{cut}]$$

In this formulation, which we will call **L**, each type is associated with a lambda expression, corresponding to its meaning, and the rules specify how these are contructed. The left rules correspond semantically to functional application, the right rules to functional abstraction. The lambda expression for the succedent of the root sequent corresponds to its meaning as a combination of the meanings of the antecedent types. This will be loosely refered to as the 'meaning (or reading) assigned by the proof'. We assume that each antecedent type of the initial sequent is assigned a distinct variable for its semantics.[4] The semantics of types will sometimes be omitted to save space.

Lambek (1958) demonstrates that the cut rule is eliminable in the sense that all theorems of the calculus can be proven using just the remaining rules. These remaining rules provide a decision procedure for theoremhood (since each inference removes a connective, so that search for proofs is finite), and we consider only these in the rest of the paper.[5]

2 The Problem of Multiple Equivalent Proofs

The calculus **L** typically allows more than one proof assigning each possible reading for a sequent. We shall consider some illustrative examples. The following two proofs assign the same reading:

$$\frac{\dfrac{\mathrm{z}{:}h \Rightarrow \mathrm{z}{:}h \qquad \mathrm{y}{:}gh \Rightarrow \mathrm{y}{:}gh}{\mathrm{y/z}{:}g,\ \mathrm{z}{:}h \Rightarrow \mathrm{y}{:}gh}\,[/\mathrm{L}] \qquad \mathrm{x}{:}fgh \Rightarrow \mathrm{x}{:}fgh}{\mathrm{x/y}{:}f,\ \mathrm{y/z}{:}g,\ \mathrm{z}{:}h \Rightarrow \mathrm{x}{:}fgh}\,[/\mathrm{L}]$$

$$\frac{\mathrm{z}{:}h \Rightarrow \mathrm{z}{:}h \qquad \dfrac{\mathrm{y}{:}gh \Rightarrow \mathrm{y}{:}gh \qquad \mathrm{x}{:}fgh \Rightarrow \mathrm{x}{:}fgh}{\mathrm{x/y}{:}f,\ \mathrm{y}{:}gh \Rightarrow \mathrm{x}{:}fgh}\,[/\mathrm{L}]}{\mathrm{x/y}{:}f,\ \mathrm{y/z}{:}g,\ \mathrm{z}{:}h \Rightarrow \mathrm{x}{:}fgh}\,[/\mathrm{L}]$$

Here we see that the operations involved in 'building' some argument type (here y) may be performed either before or after the left inference on the functor requiring that argument (here x/y). Another example of distinct proofs assigning the same meaning is the following (in both of which the subproof for the premise x/y, y ⇒ w/(w\x) is omitted):

$$\cfrac{\cfrac{\mathrm{z}{:}g \Rightarrow \mathrm{z}{:}g \qquad \cfrac{\ldots}{\mathrm{x/y}{:}fg,\ \mathrm{y}{:}i \Rightarrow \mathrm{w/(w\backslash x)}{:}\lambda j.(j(fgi))}}{\mathrm{x/y/z}{:}f,\ \mathrm{z}{:}g,\ \mathrm{y}{:}i \Rightarrow \mathrm{w/(w\backslash x)}{:}\lambda j.(j(fgi))}\,[/\mathrm{L}]}{\mathrm{x/y/z}{:}f,\ \mathrm{z}{:}g \Rightarrow \mathrm{w/(w\backslash x)/y}{:}\lambda i.\lambda j.(j(fgi))}\,[/\mathrm{R}]$$

$$\cfrac{\mathrm{z}{:}g \Rightarrow \mathrm{z}{:}g \qquad \cfrac{\cfrac{\ldots}{\mathrm{x/y}{:}fg,\ \mathrm{y}{:}i \Rightarrow \mathrm{w/(w\backslash x)}{:}\lambda j.(j(fgi))}}{\mathrm{x/y}{:}fg,\ \mathrm{y}{:}i \Rightarrow \mathrm{w/(w\backslash x)}{:}\lambda i.\lambda j.(j(fgi))}\,[/\mathrm{R}]}{\mathrm{x/y/z}{:}f,\ \mathrm{z}{:}g \Rightarrow \mathrm{w/(w\backslash x)/y}{:}\lambda i.\lambda j.(j(fgi))}\,[/\mathrm{L}]$$

These show a case where a right inference can equally well be made either before or after some unrelated left

[1] I am grateful to Guy Barry and Glyn Morrill for discussion of the ideas in this paper. Thanks also to Inge Bethke and Neil Leslie for comments on a draft. All errors are my own. This work was carried out under ESRC Postgraduate Award C00428722003.

[2] In the notation used here, types x/y and x\y both denote functions from y (the argument type) into x (the value type).

[3] A sequent is an object of the form Γ ⇒ x, where ⇒ , the derivability relation, indicates that x can be derived from Γ. Specific to the Lambek Calculus, we require that each sequent has a non-empty antecedent sequence and precisely one succedent type. We call the bottom-most sequent of any proof the *root*, and the inference that has the root as its conclusion the *root inference*. We call to the right hand premise of a left rule its *major* premise, its other the *minor* premise. In addition, we call the *main branch* of any proof that (unique) path $s_1, s_2,..,s_n$ through the proof such that s_1 is the root sequent, s_n corresponds to an axiom instance, and non of $s_2,..,s_n$ is the minor premise of a left inference.

[4] This is because we are interested in the equivalence or not of proofs aside from contingent equivalences that may stem from particular semantic assignments to antecedents.

[5] Moortgat (1990) demonstrates that cut elimination preserves the *strong recognising capacity* of the calculus in the sense that the systems with and without cut will yield precisely the same readings for any theorem modulo logical equivalence.

inference. A final example:

$$\mathrm{x/y}{:}f \Rightarrow \mathrm{x/y}{:}f \qquad \cfrac{\cfrac{\mathrm{y}{:}g \Rightarrow \mathrm{y}{:}g \quad \mathrm{x}{:}fg \Rightarrow \mathrm{x}{:}fg}{\mathrm{x/y}{:}f,\ \mathrm{y}{:}g \Rightarrow \mathrm{x}{:}fg}\,\text{[/L]}}{\mathrm{x/y}{:}f \Rightarrow \mathrm{x/y}{:}\lambda g.fg}\,\text{[/R]}$$

Here we see that where the type instantiating an axiom is functional, it is also possible to 'unpack' the type and 'equate' subformulas of this in further axiom instances. The lambda expressions assigned by these two proofs are equivalent under η-reduction.

The existence of multiple equivalent proofs presents a problem for efficient theorem proving based on the calculus **L**. Search for proofs must be exhaustive to ensure that all different 'readings' for a given sequent are found, and a naive theorem prover will expend considerable effort constructing proofs that assign the same meaning. This radically reduces the efficiency of Lambek Calculus theorem proving. Hepple and Morrill (1989), working with a somewhat different CG framework that also admits multiple equivalent proofs, suggest that this problem be dealt with by deriving a notion of normal form (NF) for proofs, and then adapting the parsing method such that this only returns NF proofs. König (1989) takes this kind approach to handling the problem of multiple equivalent proofs for Lambek sequent theorem proving, generating a notion of NF for such proofs, and deriving from this a parsing strategy intended to compute only NF proofs. However, König's parsing algorithm fails to exclude all non-NF proofs when used with a standard propositional Lambek Calculus.

In this paper I define a NF system for the sequent formulation of the (product-free) Lambek Calculus, which gives rise to a parsing approach that yields only normal proofs.

3 A New Approach

3.1 Headed Proofs

We begin by introducing the notion of *head type*, as this is crucial to the approach. We use symbols **P**, **Q**, **R**, etc to designate proofs, and the notation **P**(S) to represent a proof **P** of the sequent S. Intuitively, the head type of a sequent S under some proof **P**(S) is the antecedent type of S that has widest scope semantically. The meaning assigned by a proof is always a lambda expression of the form:[6]

$$\text{(2)} \qquad \lambda v_1..v_n.(hU_1..U_m) \qquad (n, m \geq 0)$$

where h is some semantic variable. The head type of the sequent under the given proof is that antecedent whose meaning is the variable h. Consider (3):

$$\text{(3)} \qquad \cfrac{\cfrac{\mathrm{z}{:}i \Rightarrow \mathrm{z}{:}i \qquad \cfrac{\mathrm{y}{:}f \Rightarrow \mathrm{y}{:}f \quad \mathrm{x}{:}gif \Rightarrow \mathrm{x}{:}gif}{\mathrm{y}{:}f,\ \mathrm{x\backslash y}{:}gi \Rightarrow \mathrm{x}{:}gif}\,\text{[\L]}}{\mathrm{y}{:}f,\ \mathrm{x\backslash y/z}{:}g,\ \mathrm{z}{:}i \Rightarrow \mathrm{x}{:}gif}\,\text{[/L]}}{\mathrm{y}{:}f,\ \mathrm{x\backslash y/z}{:}g \Rightarrow \mathrm{x/z}{:}\lambda i.(gif)}\,\text{[/R]}$$

The head type here is the antecedent x\y/z since it is the semantic variable g of this that has widest scope in the meaning assigned by the proof. Note the characteristic of this proof that a value subpart of the head type is 'equated' with a value subpart of the succedent type in the sense that these together instantiate opposite sides of an axiom instance, in particular that axiom instance at the top of the main branch (cf. footnote 3). This fact is the syntactic counterpart of the head type having the widest scope semantically.[7] More generally, for the head H of a sequent under some proof and the sole antecedent A of the proof's main branch axiom, it is always the case that A is a value subformula of H (or is H itself). Not all proofs have a head, as in (4), where the variable that has widest scope in the meaning assigned by the proof does not originate with an antecedent of the initial sequent, but rather with a type that is 'introduced' in a right inference:

$$\text{(4)} \qquad \cfrac{\cfrac{\cfrac{\mathrm{y}{:}g \Rightarrow \mathrm{y}{:}g \quad \mathrm{x}{:}fg \Rightarrow \mathrm{x}{:}fg}{\mathrm{x/y}{:}f,\ \mathrm{y}{:}g \Rightarrow \mathrm{x}{:}fg}\,\text{[/L]} \qquad \mathrm{z}{:}i(fg) \Rightarrow \mathrm{z}{:}i(fg)}{\mathrm{x/y}{:}f,\ \mathrm{y}{:}g,\ \mathrm{z\backslash x}{:}i \Rightarrow \mathrm{z}{:}i(fg)}\,\text{[\L]}}{\mathrm{x/y}{:}f,\ \mathrm{y}{:}g \Rightarrow \mathrm{z/(z\backslash x)}{:}\lambda i.(i(fg))}\,\text{[/R]}$$

We use numerically subscripted proof symbols (e.g. $\mathbf{P}_3$, $\mathbf{P}_\mathrm{n}$, $\mathbf{Q}_\mathrm{m}$, etc) to refer to the headedness of proofs. In particular a proof $\mathbf{P}_n$, $n \geq 1$, is a headed proof, with the head being the nth member of the antecedent sequence of the root sequent. If $n = 0$, then the proof is unheaded. Given the relation mentioned above between the head of a proof and the main branch axiom of this, we can give a recursive characterization of the form of subscripted proofs as in (5) (where superscripted Greek letters are used to indicate sequences of given numbers of types, i.e. π^n corresponds to a sequence of n types):

(5) Each proof $\mathbf{P}_n$ is of one of the forms:

a. $\dfrac{\text{axiom}}{\mathrm{x} \Rightarrow \mathrm{x}}$ where $n = 1$

b. $\dfrac{\mathbf{Q}_m(\mathrm{y}, \pi \Rightarrow \mathrm{x})}{\pi \Rightarrow \mathrm{x\backslash y}}\,\text{[\R]}$ where $((m > 0)\ \&\ (n = m - 1))$ or $((m = 0)\ \&\ (n = 0))$

c. $\dfrac{\mathbf{Q}_m(\pi^k, \mathrm{y} \Rightarrow \mathrm{x})}{\pi^k \Rightarrow \mathrm{x/y}}\,\text{[/R]}$ where $((m = k + 1)\ \&\ (n = 0))$ or $((m \leq k)\ \&\ (n = m))$

[6] Note that the meaning assigned by any (cut-free) proof is always be a lambda expression in β-normal form (i.e. containing no subterms of the form $(\lambda v.\mathrm{P})\mathrm{Q})$ — provided that the meanings assigned to the root sequent antecedents are in β-normal (which is the case here, since we assign these distinct variables) as may be shown by an inductive proof.

[7] Note the semantic significance of the main branch axiom in this. The semantics assigned by a proof **P** always corresponds to that of the succedent of the main branch axiom of **P** having undergone zero or more lambda abstractions. This follows from the semantics of the rules of **L**.

d. $$\frac{\mathbf{R}(\pi^k \Rightarrow \mathrm{y}) \qquad \mathbf{Q}_j(\phi^m, \mathrm{x}, \psi \Rightarrow \mathrm{z})}{\phi^m, \pi^k, \mathrm{x}\backslash\mathrm{y}, \psi \Rightarrow \mathrm{z}}[\backslash\mathrm{L}]$$

where $((j \leq m)\ \&\ (n = j))$
or $((j > m)\ \&\ (n = j + k))$

e. $$\frac{\mathbf{R}(\pi^k \Rightarrow \mathrm{y}) \qquad \mathbf{Q}_j(\phi^m, \mathrm{x}, \psi \Rightarrow \mathrm{z})}{\phi^m, \mathrm{x}/\mathrm{y}, \pi^k, \psi \Rightarrow \mathrm{z}}[/\mathrm{L}]$$

where $((j \leq m + 1)\ \&\ (n = j))$
or $((j > m + 1)\ \&\ (n = j + k))$

The base case for the definition is where a subproof consists only of an axiom inference, in which case the head of the proof is the single antecedent member (and hence, $n = 1$). From there, the position of the head can be kept track of by counting the number of antecedents added in beside the head as subproofs are combined. Note that *every* cut-free proof in **L** is a proof $\mathbf{P}_n$ for *some* value of n.

3.2 A Constructive Notion of Normal Form

In (6), a recursive definition is given for a *constructive* notion of NF (CNF). For this definition, it is useful to distinguish two subtypes of proof: $\mathcal{T}_1$ and $\mathcal{T}_2$. The set of $\mathcal{T}_1$ proofs is precisely the set of CNF proofs. The $\mathcal{T}_2$ proofs are a subset of the $\mathcal{T}_1$ proofs—those whose main branch includes no right inferences.

(6) The set of CNF ($\mathcal{T}_1$) proofs is the smallest set such that:

a. *if* x is an atomic type, *then* $\dfrac{\text{axiom}}{\mathrm{x} \Rightarrow \mathrm{x}} \in \mathcal{T}_2$

b. *if* $\mathbf{P}_{n+1}(\pi^n, \mathrm{x}, \psi \Rightarrow \mathrm{z}) \in \mathcal{T}_2$ *and* $\mathbf{Q}(\phi \Rightarrow \mathrm{q}) \in \mathcal{T}_1$

then $$\frac{\mathbf{Q}(\phi \Rightarrow \mathrm{q}) \quad \mathbf{P}_{n+1}(\pi^n, \mathrm{x}, \psi \Rightarrow \mathrm{z})}{\pi^n, \mathrm{x}/\mathrm{q}, \phi, \psi \Rightarrow \mathrm{z}}[/\mathrm{L}] \in \mathcal{T}_2$$

and $$\frac{\mathbf{Q}(\phi \Rightarrow \mathrm{q}) \quad \mathbf{P}_{n+1}(\pi^n, \mathrm{x}, \psi \Rightarrow \mathrm{z})}{\pi^n, \phi, \mathrm{x}\backslash\mathrm{q}, \psi \Rightarrow \mathrm{z}}[\backslash\mathrm{L}] \in \mathcal{T}_2$$

c. *if* $\mathbf{P} \in \mathcal{T}_2$ *then* $\mathbf{P} \in \mathcal{T}_1$

d. *if* $\mathbf{P}(\pi, \mathrm{x} \Rightarrow \mathrm{y}) \in \mathcal{T}_1$ *then*

$$\frac{\mathbf{P}(\pi, \mathrm{x} \Rightarrow \mathrm{y})}{\pi \Rightarrow \mathrm{y}/\mathrm{x}}[/\mathrm{R}] \in \mathcal{T}_1, \textit{ and } \frac{\mathbf{P}(\mathrm{x}, \pi \Rightarrow \mathrm{y})}{\pi \Rightarrow \mathrm{y}\backslash\mathrm{x}}[/\mathrm{R}] \in \mathcal{T}_1$$

Given (6a), CNF proofs only contain axiom leaves that are instantiated with atomic types. (6b) allows for the occurrence of left inferences in CNF proofs, and requires that these must be with respect to the head of the major premise's subproof ('major subproof'). Given (6c), every $\mathcal{T}_2$ proof is a $\mathcal{T}_1$ proof also. Given (6d), only $\mathcal{T}_1$ proofs may have a right rule as the root inference—no $\mathcal{T}_2$ proof will be of this form. Since the major subproof of a left inference *must* be a $\mathcal{T}_2$ proof, a right inference may never occur above a left inference on the main branch of a CNF proof. Thus, the main branch of a CNF proof is always of the following form (starting at the root): zero or more right inferences, followed by zero or more left inferences, terminating with an axiom inference. The minor subproofs of left inferences are $\mathcal{T}_1$, and so are also of this general form.

We will see later how we can ensure that the theorem prover generates all and only the CNF proofs of a sequent. However, for such a move to be useful, the notion of CNF must satisfy certain requirements. Most importantly, generating only CNF proofs should be *safe* in the sense that the theorem prover returns every distinct reading of a sequent that can be assigned. This will be the case if for every proof of a sequent, there exists a semantically equivalent CNF proof. To demonstrate this, a second notion of normal form will be defined, using the method of proof reduction, which is equivalent to the constructive notion of NF, and provably complete.

3.3 A Reductive Notion of Normal Form

3.3.1 Reduction and Normal Forms

A standard approach to defining NFs involves defining a *contraction* relation ($\rhd_1$) between terms. The contraction relation is stated as a number of contraction rules, each of the form X $\rhd_1$ Y (in which the form on the left is termed a *redex* and the form on the right its *contractum*). Each contraction rule allows that any term containing an occurrence of a redex may be transformed into a term in which that occurrence is replaced by its contractum. A term is said to be in NF if and only if it contains no occurrences of any redex. The contraction relation generates a *reduction* relation ($\rhd$) which is such that X *reduces* to Y (X $\rhd$ Y) if and only if Y is obtained from X by a finite series (possibly zero) of contractions. A term Y is a NF of X if and only if Y is a NF and X $\rhd$ Y.

3.3.2 Proof Reduction and the Lambek Calculus

We shall next consider a set of contraction rules stated on proofs in **L**.[8] These together define a reductive notion of NF. A total of eighteen contraction rules are required, which fall into four groups.

An instance of the first group of contraction rules is shown in (7).

(7) $$\mathrm{x}/\mathrm{y} \Rightarrow \mathrm{x}/\mathrm{y} \quad \rhd_1 \quad \cfrac{\cfrac{\mathrm{y} \Rightarrow \mathrm{y} \quad \mathrm{x} \Rightarrow \mathrm{x}}{\mathrm{x}/\mathrm{y}, \mathrm{y} \Rightarrow \mathrm{x}}[/\mathrm{L}]}{\mathrm{x}/\mathrm{y} \Rightarrow \mathrm{x}/\mathrm{y}}[/\mathrm{R}]$$

This contraction rule expands an axiom leaf instantiated with a functional type to give a subproof containing two axiom leaves, each of which is instantiated with a 'simpler' type than the original axiom (under some metric of simplicity). There is a second rule in this group for which the functional type in the redex is leftward directional.

An instance of the second group of contraction rules is shown in (8). In the redex, a right inference is applied to the major premise of left inference. In the contractum, the subproof has been restructured so

[8] Proof reduction originates with Prawitz (1965).

that the right inference is applied to the root sequent, and the left inference to its sole premise.

(8)
$$\dfrac{\mathbf{P}(\pi \Rightarrow y) \qquad \dfrac{\mathbf{Q}(\phi, x, \psi, w \Rightarrow z)}{\phi, x, \psi \Rightarrow z/w}[/R]}{\phi, x/y, \pi, \psi \Rightarrow z/w}[/L]$$

$$\rhd_1 \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow y) \quad \mathbf{Q}(\phi, x, \psi, w \Rightarrow z)}{\phi, x/y, \pi, \psi, w \Rightarrow z}[/L]}{\phi, x/y, \pi, \psi \Rightarrow z/w}[/R]$$

There are four contraction rules of this group which arise with the directionality of the connectives for the two inferences.

An instance of the third group of contraction rules is shown in (9). Note that this makes use of the subscripted proof notation, allowing us to recognise where a left inference is with respect to the head of the stated subproof. In the subproof $\mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)$, the head is the type x. It follows that the type x/y is the head of the entire redex proof, and that v/w is not. Thus we can see that in the redex, a head left inference (i.e a left inference with respect to the head of the given subproof) is applied to the major premise of non-head left inference. In the contractum, the subproof has been restructured so that the head inference is applied to the root sequent, and the non-head inference to its major premise.

(9)
$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \;\; \mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)}{\phi^n, x/y, \pi, \psi, v, \sigma \Rightarrow z}[/L]}{\phi^n, x/y, \pi, \psi, v/w, \gamma, \sigma \Rightarrow z}[/L]$$

$$\rhd_1 \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \;\; \mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)}{\phi^n, x, \psi, v/w, \gamma, \sigma \Rightarrow z}[/L]}{\phi^n, x/y, \pi, \psi, v/w, \gamma, \sigma \Rightarrow z}[/L]$$

There are eight contraction rules in this group, which arise with the directionality of the connectives involved, and whether the head functor occurs to the right or left of the non-head functor.

An instance of the fourth group of contraction rules is shown in (10). In the redex, a head-left inference is applied to the major premise of a non-head left inference, where the latter can be seen to in-part serve the purpose of 'building' the argument required by the head functor. In the contractum, the inferences have been reordered so that the head inference applies to the root sequent, and the non-head inference applies to the minor premise of this.

(10)
$$\dfrac{\mathbf{R}(\psi \Rightarrow w) \qquad \dfrac{\mathbf{Q}(\pi, v, \gamma \Rightarrow y) \;\; \mathbf{P}_{n+1}(\phi^n, x, \sigma \Rightarrow z)}{\phi^n, x/y, \pi, v, \gamma, \sigma \Rightarrow z}[/L]}{\phi^n, x/y, \pi, v/w, \psi, \gamma, \sigma \Rightarrow z}[/L]$$

$$\rhd_1 \quad \dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow w) \;\; \mathbf{Q}(\pi, v, \gamma \Rightarrow y)}{\pi, v/w, \psi, \gamma \Rightarrow y}[/L] \qquad \mathbf{P}_{n+1}(\phi^n, x, \sigma \Rightarrow z)}{\phi^n, x/y, \pi, v/w, \psi, \gamma, \sigma \Rightarrow z}[/L]$$

We require four contraction rules of this pattern, which arise with the directionality of two functors (the left-right order of the two functors being determined by the head functor).

We call any proof that contains no redexes (which is consequently a NF under the reduction system) an *irreducible normal form* (INF). It is easily verified that for each contraction rule, the contractum subproof assigns a lambda expression for the root sequent combination equivalent to that assigned by the redex. Thus, meaning is preserved under contraction, and also in turn under reduction. Hence, an INF of any proof **P** always assigns the same reading as **P**.

We will next demonstrate that *normalisation* holds for the reductive NF system; that is, that every proof has an INF. This property follows provided that the following (stronger) property holds:

(11) Strong Normalisation: Every reduction is finite.

Thus, for every proof, a finite series of contractions always leads to an irreducible proof.

To prove that every reduction is finite, it is sufficient to give a metric that assigns to each proof a finite non-negative integer score, and under which it can be shown that every application of a contraction rule reduces the score of a proof by a positive integer amount. The scoring system given in (12) satisfies these requirements.[9]

(12) The score for any proof **P** (written sc(**P**)) is as follows:

a. *if* **P** is an axiom leaf instantiated with type x *then* sc(**P**) = *factorial*(3 atc(x))

b. *if* **P** has a right inference at its root, with premise subproof **Q** *then* sc(**P**) = sc(**Q**) + 1

c. *if* the root inference of **P** is a head left inference, with major subproof **Q** and minor subproof **R** *then* sc(**P**) = (sc(**R**) + 1)sc(**Q**) + 1

d. *if* the root inference of **P** is a non-head left inference, with major subproof **Q** and minor subproof **R** *then* sc(**P**) = (sc(**R**) + 1)sc(**Q**)

3.4 Equivalence of the Two Systems

We write CNF(**P**) and INF(**P**) to indicate that a proof is in NF under the respective systems. The two NF systems can be shown to be equivalent in that they designate precisely the same sets of proofs to be in NF, i.e:

(13) Lemma 1: for all **P**, CNF(**P**) ↔ INF(**P**)

This follows given Lemmas 2 and 3.

(14) Lemma 2: for all **P**, ¬INF(**P**) → ¬CNF(**P**)

It is easily checked that no redex subproof is in CNF. Hence, any reducible proof is not a CNF.

(15) Lemma 3: for all **P**, ¬CNF(**P**) → ¬INF(**P**)

[9] The atomic type count (atc) of a type is defined as follows: (i) atc(x) = 1 if x is an atomic type; (ii) atc(x/y) = atc(x\y) = atc(x) + atc(y).

This may be proven by induction on **P**

Given the equivalence of the two NF systems, and the normalisation result for the reductive NF system, it follows that for every proof in **L**, there exists a CNF proof that assigns the same reading. Hence, generating all and only the CNF proofs of a given sequent is *safe* in the sense that we can be sure all possible readings for the combination will be returned.

3.5 The Uniqueness of Normal Forms

The approach will be *optimal* if there is only a single CNF assigning each distinct reading. As we have already noted (footnote 6), the meaning assigned by any (cut-free) proof will be a lambda expression that is in β-normal form. Extensional equivalence for the Lambda Calculus is given by the $\beta\eta$-reduction system. Since this system exhibits the Church-Rosser property, any two terms in β-normal form will be equivalent just in case applying some sequence of η-contractions (i.e. $\lambda v.fv \rhd_1 f$) to each yields the same term (modulo α-conversion).

Let us consider two distinct CNF proofs **P** and **Q** of some sequent S. Assume that these differ in some regard along their main branches. The bottom part of each main branch will consist of a sequence of zero or more right inferences. The length and nature of this sequence of inferences is fully determined by the type of the succedent in the root sequent S, and so **P** and **Q** may not differ in this regard. Thus, the meanings assigned by **P** and **Q** will be lambda expressions of the following forms (in which the lambda abstractions $\lambda v_1..v_n$. correspond to the initial sequence of right inferences):

$$\mathbf{P} : \lambda v_1..v_n.(fU_1..U_m)$$
$$\mathbf{Q} : \lambda v_1..v_n.(gU_1{}'..U_k{}')$$

The remainder of each main branch will consist of some sequence of head left inferences, terminating in an axiom instance. Consequently, the two CNF proofs may differ along their main branch only in two regards: (i) a different antecedent type is the functor for the first left inference (this determining the functor for the remaining left inferences on the main branch), (ii) even if the same type is chosen for the functor of the first left inference, at some point a different subsequence of antecedents is chosen to 'construct' the argument of the functor. In either case, the semantics assigned to the two proofs will be non-equivalent. Thus, for case (i) f and g will be distinct variables. In both cases, it may be that $m \neq k$, and even if $m = k$, at least some of the U_i and $U_i{}'$ will differ in the lambda variables that are put together to form these (i.e. the lambda expressions corresponding to antecedent semantics). It is evident that in any of these eventualities, no sequence of η-contractions can yield a common term from two such terms and so the two proofs must assign distinct readings. If the two NF proofs are similar along their main branches, they must differ in the minor subproof for some left inference. The same arguments apply in turn that if these subproofs differ in form, then they must be non-equivalent. Hence, distinct CNF proofs always differ semantically, and so the uniqueness property holds. Since reduction preserves meaning, and distinct NFs are non-equivalent, it follows that every proof has a *unique* NF.[10]

3.6 Normal Form Theorem Proving

For the purpose of theorem proving, we specify an amended version of the calculus, which we term the 'parsing calculus' (PC), which is such as to only allow the construction of proofs that correspond to CNF proofs. To this end, we move from a system which uses a single notion of sequent, to one which distinguishes two different types of sequent Type 1 and Type 2—this type being marked as a subscript on the derivability arrow $\Rightarrow$. The point of this distinction is to allow us to limit the form of the proof that may be given for some sequent merely by its type. The Type 1/Type 2 distinction essentially corresponds to the r_1/r_2 distinction in the definition of CNF. For Type 2 sequents we distinguish the antecedent element that *must* be the head of any proof of the sequent by dividing the antecedent sequence into three subparts, separated by +'s, as for example in $\phi+\mathrm{x}+\pi \Rightarrow_2 z$, of which the first and last parts are (possibly empty) sequences of types, and the middle part a single type that is required to be the head of any proof of this sequent.

The following set of inference rules allow only proofs of the required form to be constructed:

(16) Axiom: $\quad e+\mathrm{x}{:}f+e \Rightarrow_2 \mathrm{x}{:}f$ $\quad$ where x is an atomic type (and e denotes the empty sequence)

Right rules: $$\frac{\Gamma,\ \mathrm{y}{:}i \Rightarrow_1 \mathrm{x}{:}f}{\Gamma \Rightarrow_1 \mathrm{x/y}{:}\lambda i.f}[/\mathrm{R}] \qquad \frac{\mathrm{y}{:}i,\ \Gamma \Rightarrow_1 \mathrm{x}{:}f}{\Gamma \Rightarrow_1 \mathrm{x\backslash y}{:}\lambda i.f}[\backslash\mathrm{R}]$$

Left rules: $$\frac{\Delta \Rightarrow_1 \mathrm{y}{:}g \qquad \Gamma+\mathrm{x}{:}fg+\Lambda \Rightarrow_2 \mathrm{z}{:}h}{\Gamma+\mathrm{x/y}{:}f+(\Delta,\ \Lambda) \Rightarrow_2 \mathrm{z}{:}h}[/\mathrm{L}]$$

$$\frac{\Delta \Rightarrow_1 \mathrm{y}{:}g \qquad \Gamma+\mathrm{x}{:}fg+\Lambda \Rightarrow_2 \mathrm{z}{:}h}{(\Gamma,\ \Delta)+\mathrm{x\backslash y}{:}f+\Lambda \Rightarrow_2 \mathrm{z}{:}h}[\backslash\mathrm{L}]$$

$[2 \mapsto 1]$ rule: $$\frac{\Delta+\mathrm{x{:}f}+\Gamma \Rightarrow_2 \mathrm{y{:}g}}{\Delta,\ \mathrm{x{:}f},\ \Gamma \Rightarrow_1 \mathrm{y{:}g}}[2 \mapsto 1]$$ where y is an atomic type, and x = y or x a function onto y

Observe that the axiom inference specifies a Type 2 sequent, with the single antecedent type being the designated head of that sequent. This corresponds to clause (a) of the definition of CNF in (6). The left inference rules have Type 2 sequents for their conclusion and major premise and a Type 1 sequent for the minor premise. Note that the active type for the inference is the designated head of the conclusion and its value subformula is the designated head of the major premise. This corresponds to clause (b) of the CNF definition. The right rules have a Type 1

[10] From this it follows that the reductive NF system exhibits the Church-Rosser property.

premise and conclusion, in line with clause (d) of the CNF definition. In addition, there is a rule [2↦1], corresponding to the clause (c) of the CNF definition. Note there is a condition on the rule which does not correspond to anything in clause (c). It is easily shown that this condition is always fulfilled by $\mathcal{T}_2$ proofs, so its addition does not alter the set of proofs admitted by the PC. However, the condition will considerably limit the application of the [2↦1] rule in top-down search for proofs and so increase the efficiency of theorem proving. It is easily shown that PC and CNF proofs exist in 1-to-1 correspondence.

4 Discussion

It can be seen that the form of any CNF proof is strongly tied to the form of the lambda expression it assigns as its meaning. As we have seen, the lambda term corresponding to the meaning of any (cut free) proof in **L** is always a β-NF term of the form:

$$\lambda v_1..v_n.(hU_1..U_m) \quad (n,m \geq 0)$$

where h is a variable, and the main branch of a CNF proof is always of the following form (starting at the root): zero or more right inferences, followed by zero or more left inferences, terminating with an axiom inference. The correspondence between the two is as follows: the initial sequence of right inferences corresponds to the lambda abstractions of the variables $v_1..v_n$, and the ensuing left inferences are just those required to apply the variable h (the semantics of the head) to each of its arguments $U_1..U_m$ in turn, with each of the subterms U_i being 'constructed' in the subproof for a minor premise.

These observations provide the basis for relating this approach to that of König (1989), mentioned earlier. König uses a non-standard method for arriving at a notion of NF proof which involves firstly mapping proofs into objects called 'syntax trees', where proofs that yield the same syntax tree form an equivalence class, and then mapping from each syntax tree to a single NF proof. From the form of such NF proofs, König derives a set of 'nesting constraints' which are used to limit the operation of a top-down theorem prover, and which are such that they will never prevent the construction of any NF proof. As König points out, however, the 'nesting constraints' do not exclude the construction all non-NF proofs when used with a standard propositional formulation of the Lambek Calculus (though better results are obtained with a unification-based version of the Lambek Calculus that König describes). König's syntax trees can be seen to bear a strong correspondence, in terms of their structure, to the lambda term for the meaning assigned by a proof (although the former include sufficient information, of types etc, to allow (re)construction of a complete proof for the initial sequent), and the relation of König's NFs to the syntax trees used to define them closely parallels the relation between CNF proofs in the present approach and their corresponding lambda terms.

A further topic worthy of comment is the relation between the current approach and natural deduction approaches such as that of Prawitz (1965). As Prawitz observes, sequent calculi can be understood as meta-calculi for corresponding natural deduction systems. Introduction rules correspond to right rules and elimination rules to left rules. In Prawitz's NFs, an introduction rule may never apply to the major premise of an elimination rule (such a subproof being a redex) so that eliminations always appear above introductions on the main branch of a NF proof,[11] which seems to parallel the form of CNF sequent proofs. However, the relation is not so straightforward. For a natural deduction formulation of the (product-free) Lambek Calculus,[12] the occurrence of a relevant redex in a natural deduction proof (i.e. where an introduction rule applies to the major premise of an elimination) corresponds to the occurrence of a β-redex in the corresponding proof term. For sequent proofs, however, the occurrence of a β-redex corresponds to a use of the cut rule in the proof—the lambda terms for *cut-free* proofs are always in β-NF. Unfortunately, limitations of space prevent due discussion of this topic here.

References

Hepple, M. 1990. Grammatical Relations and the Lambek Calculus. In *Proceedings of the Symposium on Discontinuous Constituency.* Institute for Language Technology and Artificial Intelligence, Tilburg University, The Netherlands.

Hepple, M.R. and Morrill, G.V. (1989). Parsing and Derivational Equivalence. In: Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics. Manchester, UK. 1989.

König, E. (1989). Parsing as Natural Deduction. In: Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics. USA. 1989.

Lambek, J. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, **65**, 154-170.

Morrill, G. 1990. Grammar as Logic. To appear in: *Prodeedings of the Seventh Amsterdam Colloquium.* University of Amsterdam.

Moortgat, M. 1990. Cut Elimination and the Elimination of Spurious Ambiguity. To appear in: *Prodeedings of the Seventh Amsterdam Colloquium.* University of Amsterdam.

Prawitz, D. 1965. *Natural Deduction: A Proof-Theoretical Study.* Almqvist and Wiksell, Uppsala.

[11] The terms *main branch, major premise*, etc have been borrowed from Prawitz, and are defined analogously for his system.

[12] Note that a natural deduction formulation of the Lambek Calculus differs from standard natural deduction systems in that the linear order of assumptions within a proof is important, (roughly) corresponding to the linear order of words combined. See, for example, the 'ordered' natural deduction formulations outlined in Hepple (1990) and Morrill (1990).