Lambek Grammars as Combinatory Categorial Grammars

GERHARD JÄGER, *Zentrum für Allgemeine Sprachwissenschaft (ZAS), Jägerstr. 10/11, 10117 Berlin, Germany.*
E-mail: jaeger@zas.gwz-berlin.de

Abstract

We propose a combinatory reformulation of the product free version of the categorial calculus **LL**, i.e. the associative Lambek calculus that admits empty premises. We prove equivalence of the combinatory with the standard Natural Deduction presentation of **LL**. The result offers a new perspective on the relation between the type logical and the combinatory branch of the Categorial Grammar research program.

Keywords: Categorial Grammars, Lambek Calculus, CCG

1 The Lambek calculus

1.1 Sequent presentation

In his seminal paper [6], Joachim Lambek introduced the (associative) Lambek calculus, a type logical extension of Bar Hillel's [3] Basic Categorial Grammars. Lambek gives two equivalent proof theories for his calculus, an axiomatic one and a Gentzen style sequent calculus. We will restrict attention to the product-free fragment of this calculus. There the categorial slashes "/" and "\" are the only logical constants. The sequent calculus consists of the rules of use and of proof for them, together with the identity axiom scheme for arbitrary types. Being a subsystem of positive implicational intuitionistic logic, the history of a proof in the Lambek calculus can be recorded in λ-terms, using the Curry-Howard correspondence.

Definition 1.1 (**LL**: Sequent presentation)

$$\frac{}{x : A \Rightarrow x : A}\, id$$

$$\frac{X \Rightarrow M : A \qquad Y, x : B, Z \Rightarrow N : C}{Y, y : B/A, X, Z \Rightarrow N[x \leftarrow (yM)] : C}\, /L \qquad \frac{X, x : A \Rightarrow M : B}{X \Rightarrow \lambda x.M : B/A}\, /R$$

$$\frac{X \Rightarrow M : A \qquad Y, x : B, Z \Rightarrow N : C}{Y, X, y : A \setminus B, Z \Rightarrow N[x \leftarrow (yM)] : C}\, \setminus L \qquad \frac{x : A, X \Rightarrow M : B}{X \Rightarrow \lambda x.M : A \setminus B}\, \setminus R$$

Lambek originally added the additional constraint that the premises of each sequent be non-empty. We omit this requirement as an essentially superfluous complication, thus arriving at the product free version of the calculus **LL**.

In [6] it is proved that the Cut rule is admissible in the Lambek calculus. The proof carries over to the version of **LL** considered here without complications.

1.2 Natural Deduction

Alternatively, **LL** can be presented in a Natural Deduction format, i.e. the rules of use for the implicational slashes can be replaced by explicit Modus Ponens rules. Here Curry-Howard labels mirror the proof history directly.

DEFINITION 1.2 (**LL**: Natural Deduction)

$$\frac{}{x : A \Rightarrow x : A}\, id$$

$$\frac{X \Rightarrow M : B/A \qquad Y \Rightarrow N : A}{X, Y \Rightarrow (MN) : B}\, /E \qquad \frac{X, x : A \Rightarrow M : B}{X \Rightarrow \lambda x.M : B/A}\, /I$$

$$\frac{X \Rightarrow M : A \qquad Y \Rightarrow N : A \setminus B}{X, Y \Rightarrow (NM) : B}\, \setminus E \qquad \frac{x : A, X \Rightarrow M : B}{X \Rightarrow \lambda x.M : A \setminus B}\, \setminus I$$

It is easy to show that the two inference formats are equivalent. Even stronger, sequent derivations and ND derivations generate the same set of proof terms.

Since λ-terms record the structure of ND proofs, Curry-Howard labeling thus gives us a compact representation of this proof format. However, due to the non-commutativity of **LL**, we have to distinguish between rightward and leftward abstraction and application. To get a complete match between ND proofs and Curry-Howard terms, this distinction has to be coded in the term language.

In [10], Wansing shows how this can be done. We write $\lambda^l x.M$ in Curry-Howard terms rather than $\lambda x.M$ if the main functor of the type of the whole term is "\"; otherwise we write $\lambda^r x.M$. Furthermore, if M has type $B \setminus A$ and N has type B, we write (NM) rather than (MN). If we talk about the rightmost or leftmost free variable occurrence in a term, we assume the linearization induced by this modified notation. This notation has the obvious advantage that the sequence of free variables in a derivable proof term directly mirrors the sequence of premises of the corresponding proof (a feature that is shared by the combinatory system).

Following these conventions, the labeled ND calculus for **LL** looks as follows.

DEFINITION 1.3 (**LL**: Natural Deduction with directed terms)

$$\frac{}{x : A \Rightarrow x : A}\, id$$

$$\frac{X \Rightarrow M : B/A \qquad Y \Rightarrow N : A}{X, Y \Rightarrow (MN) : B}\, /E \qquad \frac{X, x : A \Rightarrow M : B}{X \Rightarrow \lambda^r x.M : B/A}\, /I$$

$$\frac{X \Rightarrow N : A \qquad Y \Rightarrow M : A \setminus B}{X, Y \Rightarrow (NM) : B}\, \setminus E \qquad \frac{x : A, X \Rightarrow M : B}{X \Rightarrow \lambda^l x.M : A \setminus B}\, \setminus I$$

In [10], Wansing proves the following lemma.

LEMMA 1.4
A term M is a proof term of an **LL**-proof iff

1. Every λ binds exactly one variable occurrence.
2. For every subterm $\lambda^l x.N$ of M, x is the leftmost free variable occurrence in N.
3. For every subterm $\lambda^r x.N$ of M, x is the rightmost free variable occurrence in N.

Figure 1 displays a sample derivation of the relative clause construction. For better readability, the deduction is given in tree format rather than as a sequent derivation.

(1) book that John liked

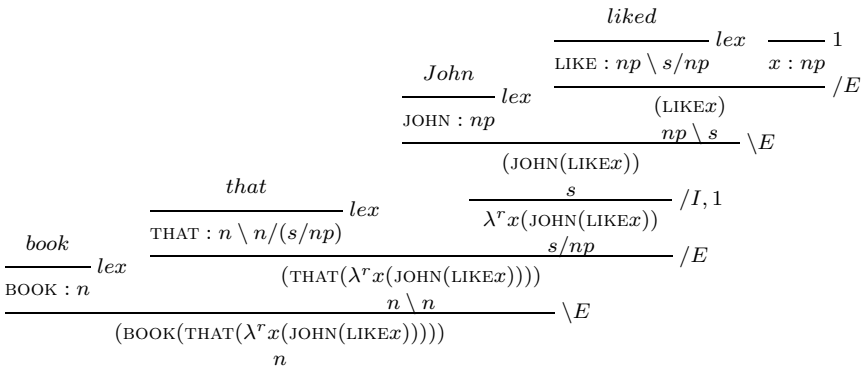

FIG. 1. **LL**-derivation of *book that john liked*

Here both introduction rules and elimination rules are involved. As the example illustrates, the directionalized Curry-Howard labels encode two kinds of linguistic information: They may serve as input for the semantic component (by ignoring the directional information), and they also record prosodic information (deleting all lambdas and variables results in a prosodic term).

2 Combinatory Categorial systems

The Combinatory branch of Categorial grammar was initiated by the work of Ades and Steedman ([1]). As in Basic Categorial Grammars, CCG (Combinatory Categorial Grammar) makes use of the identity axiom and the slash elimination rules [/E] and [\E], but it does without the slash introduction rules [/I] and [\I]. The Basic Categorial core is extended by other schemes of inference though. Most work in the CCG tradition assumes some—possibly restricted—version of Type Raising **T** and (generalized) Function Composition **B** (for ease of comparison, we employ a sequent style notation here). As in the systems discussed above, the history of a derivation can be recorded in a label in CCG; labels are terms in the closure of the set of typed variables under the set of combinators.

Definition 2.1 (Type Raising)

$$\frac{X \Rightarrow M : A}{X \Rightarrow \mathbf{T}_{>}(M) : B/(A \setminus B)}\,\mathbf{T}_{>} \qquad \frac{X \Rightarrow M : A}{X \Rightarrow \mathbf{T}_{<}(M) : (B/A) \setminus B}\,\mathbf{T}_{<}$$

Definition 2.2 (Function Composition)

$$\frac{X \Rightarrow M : A/B \qquad Y \Rightarrow N : C_{n_l} \setminus \cdots \setminus C_1 \setminus B/D_1/\cdots/D_{n_r}}{XY \Rightarrow \mathbf{B}_{>}(M,N) : C_{n_l} \setminus \cdots \setminus C_1 \setminus A/D_1/\cdots/D_{n_r}}\,\mathbf{B}_{>}$$

$$\frac{X \Rightarrow N : C_{n_l} \setminus \cdots \setminus C_1 \setminus B/D_1/\cdots/D_{n_r} \qquad M : B \setminus A \Rightarrow Y}{XY \Rightarrow \mathbf{B}_{<}(N,M) : C_{n_l} \setminus \cdots \setminus C_1 \setminus A/D_1/\cdots/D_{n_r}}\,\mathbf{B}_{<}$$

It is easy to see that the well-known schemes of ordinary function composition

$$A/B, B/C \Rightarrow A/C$$

and crossed composition

$$A/B, C \setminus B \Rightarrow C \setminus A$$

(and their mirror images) are special cases of the more general operations given above.

Next to these combinatory inference schemes, CCGs might contain empty categories, i.e. axioms of the form

$$\frac{}{\Rightarrow M : A}$$

for some term M and type A. They may be considered as 0-place combinators.

CCG is a non-logical deductive system for two reasons. First, the categorial slashes are not interpreted as implications or any other known logical connectives. Second and more important, the applicability of combinators may be restricted to certain type instances. To illustrate this point, in [9] Steedman restricts the applicability of the type raising rule $A \Rightarrow B/(A \setminus B)$ to those instances where A is an "argument category". The complex category $np \setminus s$ for instance would be excluded there. Such fine-grainedness is beyond the expressive capabilities of a logical approach like Lambek's, since there theorems are always closed under uniform substitution of atoms. For the purposes of the present paper, we will not make use of this feature of CCG and treat combinators as schemata over all their type instances.

Figure 2 gives a derivation of the linguistic example (1) in CCG.

3 Combinatory presentation of LL

Even though both branches of modern Categorial grammar are based on the same basic intuitions, the formal properties of the the systems differ considerably. This is most obvious when we consider issues of weak generative capacity: Since [8] it is known that Lambek grammar recognize exactly the context free languages (and the

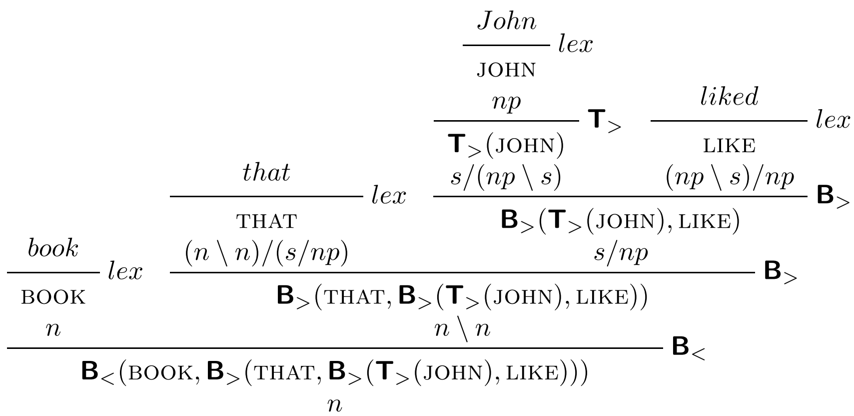

Fig. 2. CCG-derivation of *book that John liked*

proof carries over directly to **LL**), while the generative capacity of CCGs is mildly context sensitive (cf. [5]). So the two approaches seem to be irreconcilable.

Recent developments in type logical grammar have changed this rigid picture. The most important new concept within this paradigm is the notion of multi-modality. The idea is to use several families of type logical connectives that communicate with each other via interaction postulates. This leads to hybrid logics where certain powerful structural rules might be applicable under certain conditions but not in general. The use of multi-modality (see for instance [7]) increases the generative capacity of type logical grammars in principle to Turing completeness (cf. [4]), so there is no principal obstacle to simulate CCGs in the type logical framework.

In this section, we will pursue the opposite strategy. It will be shown that under a slight modifcation and generalization of the notion of function composition as it is used in the CCG tradition, **LL** can equivalently be reformulated as a combinatory system.

The route that we will pursue is as follows. In the first step we will design a combinatory system, $\mathbf{LL}_C$, that can be seen as a version of CCG. Then we will develop a hybrid system $\mathbf{LL}_H$ that comprises both the inference rules of the ND presentation of **LL** and the combinatory rules from $\mathbf{LL}_C$. Finally we will prove that the set of theorems derivable $\mathbf{LL}_H$ is identical both to the theorems of **LL** and the theorems of $\mathbf{LL}_C$. In this way we indirectly establish the equivalence of **LL** with $\mathbf{LL}_C$. The structure of the argument is schematically shown in figure 3.

Some terminology before we start in earnest: we will call the first operand of forward function composition $\mathbf{B}_>$ and the second operand of backward composition $\mathbf{B}_<$ ("M" in the schemes above) the *functor* and the other operand ("N") the *argument* of the composition operation. In the schemes given above, the argument is always matched with an outermost argument place of of the functor (the rightmost one in the case of $\mathbf{B}_>$ and the leftmost one in the case of $\mathbf{B}_<$). The notion of function composition to be employed here generalizes this aspect: $\mathbf{B}_>$ can target any forward looking (and $\mathbf{B}_<$ any backward looking) argument slot of the functor. To keep track of the argument slot to be addressed, we use a superscript notation.

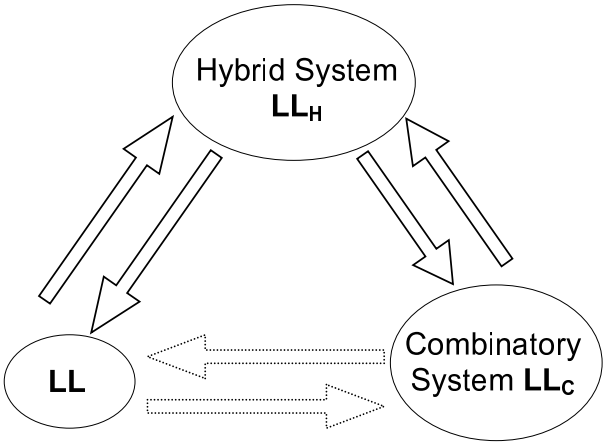

FIG. 3. Relation between $\mathbf{LL}, \mathbf{LL}_C$ and $\mathbf{LL}_H$

Some instances of CCG's notion of function composition are not theorems of **LL**, and the same holds for the generalization to be presented below. Therefore Generalized Function Composition is subject to certain constraints; certain instances are only licit if one or both of the operands are closed, i.e. do not contain free variables. This ensures that in fact all instances are admissible in **LL**—a fact that will be proved later on.

DEFINITION 3.1 (Generalized Function Composition)

$$\frac{X \Rightarrow M : A/B_1/\cdots/B_i \qquad Y \Rightarrow N : C_{n_l} \setminus \cdots \setminus C_1 \setminus B_1/D_1/\cdots/D_{n_r}}{XY \Rightarrow \mathbf{B}^i_>(M,N) : C_{n_l} \setminus \cdots \setminus C_1 \setminus A/D_1/\cdots/D_{n_r}/B_2/\cdots/B_i}\,\mathbf{B}^i_>$$

$(i = 1 \vee n_l = 0) \wedge$
$(i > 1 \rightarrow N \text{ is closed}) \wedge$
$(n_l > 0 \rightarrow M \text{ is closed})$

$$\frac{X \Rightarrow N : C_{n_l} \setminus \cdots \setminus C_1 \setminus B_1/D_1/\cdots/D_{n_r} \qquad Y \Rightarrow M : B_i \setminus \cdots \setminus B_1 \setminus A}{XY \Rightarrow \mathbf{B}^i_<(N,M) : B_i \setminus \cdots \setminus B_2 \setminus C_{n_l} \setminus \cdots \setminus C_1 \setminus A/D_1/\cdots/D_{n_r}}\,\mathbf{B}^i_<$$

$(i = 1 \vee n_r = 0) \wedge$
$(i > 1 \rightarrow N \text{ is closed}) \wedge$
$(n_r > 0 \rightarrow M \text{ is closed})$

Furthermore we assume any type instance of the identity combinator **I**:

DEFINITION 3.2 (Identity Combinator)

$$\frac{}{\Rightarrow \mathbf{I}^A_> : A/A}\,\mathbf{I}^A_>$$

$$\frac{}{\Rightarrow \mathbf{I}^A_< : A \setminus A}\,\mathbf{I}^A_<$$

Finally, an appropriate combinatory reformulation of **LL** would require generalizations of the "swapping" rule $A\backslash(B/C) \Leftrightarrow (A\backslash B)/C$. Since this rule is reversible, it induces an equivalence relation on types. We thus suppress the swapping rule. Instead we tacitly consider each type in a derivation as a representative of its equivalence class induced by swapping.

Let us say that a sequent $X \Rightarrow A$ is derivable in $\mathbf{LL}_C$ iff it is derivable by using only the identity axiom $[id]$ and the combinatory schemes $\mathbf{I}^A_>, \mathbf{I}^A_<, \mathbf{B}^i_>$, and $\mathbf{B}^i_<$ for arbitrary types A and natural numbers i.

As indicated above, we will establish the equivalence between **LL** and $\mathbf{LL}_C$ by embedding both systems into a hybrid system that will be shown to be equivalent to both original systems. This hybrid system, $\mathbf{LL}_H$, is simply the combination of **LL** and $\mathbf{LL}_C$. So a sequent is derivable in $\mathbf{LL}_H$ iff it is derivable from instances of the identity axiom by means of the ND rules of **LL** and the combinatory rules of $\mathbf{LL}_C$.

The equivalence of $\mathbf{LL}_H$ with **LL** is fairly easy to show.

LEMMA 3.3
A sequent $X \Rightarrow A$ is derivable in $\mathbf{LL}_H$ iff it is derivable in **LL**.

PROOF. The *if* direction is obvious since the axioms and rules of **LL** are also axioms or rules of $\mathbf{LL}_H$. To prove the *only if* direction, we have to show that all rules of $\mathbf{LL}_H$ are admissible in **LL**. For the logical rules this is trivial; but we have to show it for the combinatory rules **B** and **I** as well. As for **B**, six cases have to be distinguished, depending on the directionality and the values of the parameters n_l and n_r. For forward composition $\mathbf{B}_>$ the three cases are 1. $i = 1$ and $n_l = 0$, 2. $i = 1$ and $n_l > 0$, and $i > 1$ and $n_l = 0$. For backward composition $\mathbf{B}_<$ the three cases are analogous except that n_l is to be replaced by n_r.

We prove the admissibility of $\mathbf{B}_>$ for each of the three cases separately.

1. $\mathbf{B}^i_>, i = 1, n_l = 0$

$$\dfrac{\dfrac{X \Rightarrow A/B_1 \qquad \dfrac{\dfrac{Y \Rightarrow B_1/D_1/\cdots/D_n}{\vdots}\,/E}{Y, D_n, \cdots, D_1 \Rightarrow B_1}\,/E}{X, Y, D_n, \cdots, D_1 \Rightarrow A}\,/E}{\dfrac{\vdots}{X, Y, \Rightarrow A/D_1/\cdots/D_n}\,/I}\,/I$$

2. $\mathbf{B}^i_>, i = 1, n_l > 0, X = \varepsilon$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Y \Rightarrow C_{n_l} \setminus \cdots \setminus C_1 \setminus B_1/D_1/\cdots/D_{n_r}}{\vdots}\setminus E}{C_1,\cdots,C_{n_l},Y \Rightarrow B_1/D_1/\cdots/D_{n_r}}/E}{\vdots}/E}{C_1,\cdots,C_{n_l},Y,D_{n_r},\cdots,D_1 \Rightarrow B_1} \qquad \Rightarrow A/B_1}/E}{C_1,\cdots,C_{n_l},Y,D_{n_r},\cdots,D_1 \Rightarrow A}/I}{\vdots}/I}{C_1,\cdots,C_{n_l},Y \Rightarrow A/D_1/\cdots/D_{n_r}}\setminus I}{\vdots}\setminus I}{Y \Rightarrow C_{n_l} \setminus \cdots \setminus C_1 \setminus A/D_1/\cdots/D_{n_r}}\setminus I$$

3. $\mathbf{B}^i_>, i > 1, n_l = 0, Y = \varepsilon$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X \Rightarrow A/B_1/\cdots/B_i}{\vdots}/E \quad \dfrac{\Rightarrow B_1/D_1/\cdots/D_{n_r}}{\vdots}/E}{X,B_i,\cdots,B_2 \Rightarrow A/B_1} \quad \dfrac{}{D_{n_r},\cdots,D_1 \Rightarrow B_1}/E}/E}{X,B_i,\cdots,B_2,D_{n_r},\cdots,D_1 \Rightarrow A}/I}{\vdots}/I}{X \Rightarrow A/D_1/\cdots/D_{n_r}/B_2/\cdots/B_i}/I$$

The three subcases of backward composition $\mathbf{B}_<$ can be proved by mirror images of the above proofs.

The identity combinators can be derived directly from the identity axioms:

$$\dfrac{A \Rightarrow A}{\Rightarrow A/A}/I \qquad \dfrac{A \Rightarrow A}{\Rightarrow A \setminus A}\setminus I$$

∎

It remains to be shown that $\mathbf{LL}_H$ is also equivalent to $\mathbf{LL}_C$. Here we define a translation between terms (and thus implicitly between proofs) to establish the result.

LEMMA 3.4
A sequent is derivable in $\mathbf{LL}_H$ iff it is derivable in $\mathbf{LL}_C$.

PROOF. The *if* direction is obvious again. To prove the other direction, we define a reduction relation on hybrid proof terms that eventually transforms every $\mathbf{LL}_H$-proof

term into a purely combinatory term. We use the notation $M[x]$ to indicate that the term M contains a free occurrence of the variable x.

Definition 3.5 (Reduction)

$$
\begin{aligned}
(M : A/B, N : B) &\rightsquigarrow \mathbf{B}^1_>(M, N) && (3.1)\\
(N : B, M : B \setminus A) &\rightsquigarrow \mathbf{B}^1_<(N, M) && (3.2)\\
\lambda^l x.x : A &\rightsquigarrow \mathbf{I}_< : A \setminus A && (3.3)\\
\lambda^l x.\mathbf{B}^i_>(M[x], N) &\rightsquigarrow \mathbf{B}^i_>(\lambda^l x.M, N) && (3.4)\\
\lambda^l x.\mathbf{B}^i_>(M, N[x]) &\rightsquigarrow \mathbf{B}^i_>(M, \lambda^l x.N) && (3.5)\\
\lambda^l x.\mathbf{B}^i_<(N[x], M) &\rightsquigarrow \mathbf{B}^i_<(\lambda^l x.N, M) && (3.6)\\
\lambda^l x.\mathbf{B}^i_<(N, M[x]) &\rightsquigarrow \mathbf{B}^{i+1}_<(N, \lambda^l x.M) && (3.7)\\
\lambda^r x.x : A &\rightsquigarrow \mathbf{I}_> : A/A && (3.8)\\
\lambda^r x.\mathbf{B}^i_>(M[x], N) &\rightsquigarrow \mathbf{B}^{i+1}_>(\lambda^r x.M, N) && (3.9)\\
\lambda^r x.\mathbf{B}^i_>(M, N[x]) &\rightsquigarrow \mathbf{B}^i_>(M, \lambda^r x.N) && (3.10)\\
\lambda^r x.\mathbf{B}^i_<(N[x], M) &\rightsquigarrow \mathbf{B}^i_<(\lambda^r x.N, M) && (3.11)\\
\lambda^r x.\mathbf{B}^i_<(N, M[x]) &\rightsquigarrow \mathbf{B}^i_<(N, \lambda^r x.M) && (3.12)
\end{aligned}
$$

The reduction relation is generalized to subterms in the obvious way: if $M \rightsquigarrow M'$, then $N[M/x] \rightsquigarrow N[M'/x]$.

Next it has to be shown that this reduction relation preserves the type of the term, its sequence of premises, and derivability in $\mathbf{LL}_H$. In clause (3.1) this is obviously the case since $\mathbf{B}^1_>$ is applicable to any well-formed terms $M : A/B, N : B$, and the type of the resulting term is A. The same holds likewise for clause (3.2). In clause (3.3) both the redex and the resulting term are derivable, have type $A \setminus A$, and do not contain FVOs. As for clause (3.4), note that as in pure labeled $\mathbf{LL}$, every λ^l binds exactly one FVO, and this is the leftmost FVO in its scope (and likewise for λ^r). Hence the sequence of FVOs is identical in the redex and the result, and the only occurrence of x is the leftmost FVO in M. Hence $\lambda^l M$ is a derivable $\mathbf{LL}_H$-term. Furthermore, for $\mathbf{B}^i_>(M[x], N)$ to be derivable, the type of M must be $A/B_i/\cdots/B_1$, and the type of N is $B_i/\vec{D}$ (there are no C since $M[x]$ is not closed). Thus the type of $\mathbf{B}^i_>(M[x], N)$ is $A/\vec{D}/B_2/\cdots/B_1$. If $x : E$, the type of the redex is thus $E \setminus A/\vec{D}/B_2/\cdots/B_1$. The type of $\lambda^l x M$ is thus $E \setminus A/B_i/\cdots/B_1$. This means that $\mathbf{B}^i_>(\lambda^l x.M, N)$ is defined and has the type $E \setminus A/\vec{D}/B_2/\cdots/B_1$ as well. The side condition $i > 1 \rightarrow FVO(N) = \emptyset$ applies both to redex and result, so if it is fulfilled for the redex, it is also fulfilled for the result.

Note that in clauses (3.5) – (3.7), $\lambda^l x$ binds the leftmost FVO in its scope for the redex to be a derivable proof term. Thus M in (3.5) and N in (3.7) must be closed, $\lambda^l x$ in the result term also binds the leftmost FVO in its scope, and the sequences of FVO in the redex and in the result are identical in all three cases. As for (3.5), type identity between redex and result is easy to check. The side conditions on the applicability of $\mathbf{B}^i_>$ require M to be closed in the result, but this is guaranteed since otherwise $\lambda^l x$ would not bind the leftmost FVO in the redex.

In (3.6), $N[x]$ is not closed and thus $i = 1$ for the redex to be well-formed. Under these conditions, type identity between redex and result is easy to check, and the side condition $n_r > 0 \rightarrow FVO(M) = \emptyset$ is identical for redex and result.

In clause (3.7) N must be closed, since otherwise $\lambda^l x$ would not bind the leftmost FVO in the redex. Furthermore $FVO(M[x]) \neq \emptyset$, hence $n_r = 0$ for the redex to be well-formed. Thus the result is well-formed if the redex is, and type identity between redex and result is easily checked.

Preservation of well-formedness, type, and sequence of FVO can be proved similarly for the mirror images of (3.3) – (3.7), i.e. (3.8) – (3.12).

Next we show that every $\mathbf{LL}_H$-term is either a purely combinatory term, or it contains a redex. By definition, every use of function application ($/E$ or $\backslash E$) creates a redex according to clauses (3.1, 3.2), so a term in normal form does not contain function application. So we have to show that every λ in a well-formed term creates a redex. Suppose the scope of the λ is a variable. Since every λ binds exactly one FVO, so such a configuration either matches clause (3.3) or (3.8). The scope of a λ cannot be an identity combinator since then the λ would bind no FVO. The remaining possibility is that the scope of a λ is a term headed by $\mathbf{B}$. There are eight possible sub-configurations, depending on whether the λ is λ^l or λ^r, whether $\mathbf{B}$ is $\mathbf{B}_>$ or $\mathbf{B}_>$, and whether the λ binds a FVO in the first or in the second operand of $\mathbf{B}$. These eight cases each correspond to one of the clauses (3.4) – (3.7) and (3.9) – (3.12).

Finally it remains to be shown that the reduction relation defined above strongly normalizes. It is easy to see that this is in fact the case since each reduction step either reduces the number of function applications, the number of λs, or it reduces the number of symbols that intervene between a λ and the FVO that it binds. Since these parameters are always natural numbers and none of them can ever be increased by any reduction step, any sequence of reduction steps eventually terminates. So any $\mathbf{LL}_H$-proofterm can effectively be transformed into an $\mathbf{LL}_C$-proofterm, and this means that any proof in $\mathbf{LL}_H$ can be translated into a proof in $\mathbf{LL}_C$. ∎

This leads directly to the main result of this paper.

THEOREM 3.6
A sequent is derivable in **LL** iff it is derivable in $\mathbf{LL}_C$.

PROOF. Immediately from the two preceding lemmas. ∎

We conclude this section with some examples for translations from **LL**-proofs into $\mathbf{LL}_C$-proofs, using the construction from the proof of lemma 3.4. We start with the type lifting theorem $A \Rightarrow (B/A) \setminus B$.

$$\begin{array}{rcl} x : A & \Rightarrow & (\lambda^l y.yx) : (B/A) \setminus B \\ & = & \lambda^l y.\mathbf{B}^1_>(y, x) \\ & = & \mathbf{B}^1_>(\lambda^l y.y, x) \\ & = & \mathbf{B}^1_>(\mathbf{I}_<, x) \end{array}$$

Note that the combinatory proof of this theorem makes use of the 0-place combinator $\mathbf{I}$, even though its proof in **LL** does without empty premises.

A somewhat more complex example is type lowering $A/(B/(C \setminus B)) \Rightarrow A/C$, which involves the management of several λs.

$$
\begin{array}{lcl}
x : A/(B/(C \setminus B)) & \Rightarrow & \lambda^r y.x(\lambda^r z.yz) : A/C \\
& = & \lambda^r y \mathbf{B}^1_>(x, (\lambda^r z \mathbf{B}^1_<(y, z))) \\
& = & \lambda^r y \mathbf{B}^1_>(x, (\mathbf{B}^1_<(y, \lambda^r zz))) \\
& = & \lambda^r y \mathbf{B}^1_>(x, \mathbf{B}^1_<(y, \mathbf{I}_>)) \\
& = & \mathbf{B}^1_>(x, \lambda^r y \mathbf{B}^1_<(y, \mathbf{I}_>)) \\
& = & \mathbf{B}^1_>(x, \mathbf{B}^1_<(\lambda^r yy, \mathbf{I}_>)) \\
& = & \mathbf{B}^1_>(x, \mathbf{B}^1_<(\mathbf{I}_>, \mathbf{I}_>))
\end{array}
$$

Finally, in figure 4 we give the translation of the **LL**-derivation of the linguistic example from the beginning.

$$
\dfrac{\dfrac{book}{\text{BOOK}}\,lex \quad \dfrac{\dfrac{that}{\text{THAT}}\,lex \quad \dfrac{\dfrac{John}{\text{JOHN}}\,lex \quad \dfrac{\dfrac{liked}{\text{LIKE}}\,lex \quad \dfrac{}{\mathbf{I}_>}\,\mathbf{I}_>}{np/np}}{np \setminus s/np}\,\mathbf{B}^1_>}{\mathbf{B}^1_<(\text{JOHN}, \mathbf{B}^1_>(\text{LIKE}, \mathbf{I}_>))}\,\mathbf{B}^1_<}{\mathbf{B}^1_>(\text{THAT}, \mathbf{B}^1_<(\text{JOHN}, \mathbf{B}^1_>(\text{LIKE}, \mathbf{I}_>)))}\,\mathbf{B}^1_>}{\mathbf{B}^1_<(\text{BOOK}, \mathbf{B}^1_>(\text{THAT}, \mathbf{B}^1_<(\text{JOHN}, \mathbf{B}^1_>(\text{LIKE}, \mathbf{I}_>))))}\,\mathbf{B}^1_<
$$

FIG. 4. $\mathbf{LL}_C$-derivation of *book that John liked*

4 Conclusion

The present paper presented a novel Combinatory Categorial system making use of the identity combinator, a modified and generalized version of function composition and—implicitly—swapping. As main result, it was proved that under the appropriate restriction of generalized function composition, this combinatory system is equivalent to the product free version of the calculus **LL**, the variant of the associative Lambek calculus that does without the restriction to empty premises.

At the present point, the significance of the result lies in the fact that is makes a connection between the two main branches of Categorial Grammar; it is thus a base for transferring concrete analyses from the Lambek tradition into the Combinatory tradition. Two directions for further research suggest themselves: Are there similar constructions for other type logical calculi like full **LL**, **L**, the non-associative Lambek calculus **NL** etc.? And, does the combinatory reformulation offer new insights about the Lambek calculus as such? Especially investigations of the proof theoretic complexity of type logical grammars might profit from the combinatory perspective.

References

[1] Anthony E. Ades and Mark J. Steedman. On the order of words. *Linguistics and Philosophy*, 4:517–558, 1982.

[2] Kazimierz Ajdukiewicz. Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27, 1935.

[3] Yehoshua Bar-Hillel. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58, 1953.

[4] Bob Carpenter. The Turing-completeness of multimodal Categorial Grammars. unpublished manuscript, 1995.

[5] Aravind Joshi, K. Vijay-Shanker, and David Weir. The convergence of mildly context-sensitive grammar formalisms. In Tom Wasow and Peter Sells, editors, *Processing of Linguistic Structure*, pages 31–81. MIT Press, Cambridge(Mass.), 1990.

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

[7] Michael Moortgat. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, chapter 2, pages 93–178. Elsevier, MIT Press, 1997.

[8] Martin Pentus. Lambek grammars are context-free. In *Proceedings of the 8th Annual IEEE Symposium on Logic in Computer Science*. Montreal, 1993.

[9] Mark Steedman. *Surface Structure and Interpretation*. MIT Press, Cambridge (Mass.), 1996.

[10] Heinrich Theodor Wansing. *The Logic of Information Structures*. Springer Lecture Notes in Artificial Intelligence 681. Springer Verlag, Berlin, 1993.