Clausal Proofs and Discontinuity

Glyn Morrill

Secció d'Intel·ligència Artificial
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Pau Gargallo, 5
08028 Barcelona

morrill@lsi.upc.es

Abstract

We consider the task of theorem proving in Lambek calculi and their generalisation to "multimodal residuation calculi". These form an integral part of categorial logic, a logic of signs stemming from categorial grammar, on the basis of which language processing is essentially theorem proving. The demand of this application is not just for efficient processing of some or other specific calculus, but for methods that will be generally applicable to categorial logics.

It is proposed that multimodal cases be treated by dealing with the highest common factor of all the connectives as linear (propositional) validity. The prosodic (sublinear) aspects are encoded in labels, in effect the term-structure of quantified linear logic. The correctness condition on proof nets ("long trip condition") can be implemented by SLD resolution in linear logic with unification on labels/terms *limited to one way matching.* A suitable unification strategy is obtained for calculi of discontinuity by normalisation of the ground goal term followed by recursive decent and redex pattern matching on the head term.

Clausal Proofs and Discontinuity[1]

The associative Lambek calculus (Lambek 1958) and non-associative Lambek calculus (Lambek 1961) were originally proposed as "syntactic calculi" for characterisation of the well-formedness of (respectively) sequential (semigroup structure) and binary hierarchical (groupoid structure) expressions, and were provided with single-conclusioned Gentzen-style sequent presentations which lack the usual structural rules of weakening (or: thinning, or: monotonicity), contraction, and permutation (or: exchange), and which directly provide Cut-free backward-chaining decision procedures for theoremhood.

More recently it has become possible to locate the Lambek calculi within a space of "substructural logics" (logics lacking structural rules; Došen and Schroeder-Heister 1993) of which linear logic (Girard 1987) is a prominent instance. At the same time, Lambek calculi have been extended in their linguistic application to categorial logics (Morrill 1994d), versions of categorial grammar characterising prosodic and semantic dimensions, for which the task of parsing is essentially theorem proving. In particular we can identify as a generalisation "residuation calculi" in which the Lambek connectives (corresponding to linear logic multiplicatives) are defined in a number of potentially interactive modes (Moortgat and Morrill 1991). In Morrill (1993, 1994d) an improvement of the logic of discontinuity of Moortgat (1988) is developed in this way. Given Cut-elimination, decidability is directly demonstrable from sequent formulations, but in applications to natural language processing our further objective is efficiency.

There are two main approaches in existence: sequent proof normalisation, and proof nets. The former, which builds proofs backwards from the goal sequent, even if somehow broadly generalisable, necessarily faces non-determinism with information from subformulas only made available serially according to the construction of formulas. The latter provides a phase of unfolding in which all the parts of a formula are made available in parallel, and then a non-deterministic phase of linking which builds proofs from the axioms, but requires a certain correctness condition. Roorda (1991) expresses this condition by reference to labelling by lambda terms corresponding to proofs under the Curry-Howard correspondence. Roorda (1991) and Moortgat (1990, 1992) do so by reference to labelling by groupoid terms of the algebras in which we interpret by residuation. We aim to improve the latter method, which as it stands presents the task of correctness checking in terms of intractable problems such as semigroup unification, i.e. it leaves some more specific structuring of the task, indicating an efficient strategy, to be desired. Moortgat (1990) presents a scheme for gathering groupoid-labelled unfoldings into definite clauses directly executable in Prolog, and Moortgat (1992) proposes multimodal generalisation with unification under theory. In Morrill (1994a) such a compilation is achieved by a more direct structuring of unfolding relating to Horn clause resolution in linear logic, showing how one term in such unification can always be kept ground, and multimodality is exemplified with the logic of discontinuity of Morrill (1993, 1994d). This refinement however shares with the Moortgat proposals transformation into first order clauses, resulting in an inflation of the resolution database at compile time to deal with higher order type inferences. In Morrill (1994b) the situation is improved by compiling into higher order clauses such that hypotheticals are emitted dynamically only as they become germane. The present paper aims to explain and motivate these proposals.

[1] To appear in the *Bulletin of the Interest Group in Propositional and Predicate Logics*. I thank Michael Moortgat and Dick Oehrle for comments on this work.

1 Residuation Calculi

1.1 Lambek Calculi

The types (or: formulas) of (product-free) Lambek calculus are freely generated from a set of primitives by binary infix connectives / ("over") and \ ("under"). Models can be given in a variety of structures; we deal here with a simple and transparent interpretation in groupoids. With respect to a groupoid algebra $\langle L, +\rangle$ (i.e. a set L closed under a binary operation +) for the non-associative Lambek calculus **NL**, and with respect to a semigroup algebra $\langle L, +\rangle$ (i.e. a set L closed under an associative binary operation +) for the associative Lambek calculus **L**, each formula A is "prosodically" interpreted as a subset $D(A)$ of L by residuation as follows (Lambek 1988).

(1)
$$\begin{array}{rcl} D(A\backslash B) & = & \{s \mid \forall s' \in D(A), s'+s \in D(B)\} \\ D(B/A) & = & \{s \mid \forall s' \in D(A), s+s' \in D(B)\} \end{array}$$

A sequent, $\Gamma \vdash A$, comprises a succedent formula A and one or more formula occurrences in the antecedent configuration Γ which is organised as a binary bracketed sequence for **NL**, and as a sequence for **L**. A sequent is valid if and only if in all interpretations applying the prosodic construction indicated by the antecedent configuration to objects inhabiting its formulas always yields an object inhabiting the succedent formula. The Gentzen-style sequent presentations for **NL** in (2) and for **L** in (3) are sound and complete for this interpretation (Buszkowski 1986, Došen 1992); furthermore they enjoy Cut-elimination: every theorem can be generated without the use of Cut. In the following the parenthetical notation $\Gamma(\Delta)$ represents a configuration containing a distinguished subconfiguration Δ.

(2) a. $A \vdash A$ id $\qquad \dfrac{\Gamma \vdash A \quad \Delta(A) \vdash B}{\Delta(\Gamma) \vdash B}\text{Cut}$

b. $\dfrac{\Gamma \vdash A \quad \Delta(B) \vdash C}{\Delta([\Gamma, A\backslash B]) \vdash C}\backslash\text{L} \qquad \dfrac{[A, \Gamma] \vdash B}{\Gamma \vdash A\backslash B}\backslash\text{R}$

c. $\dfrac{\Gamma \vdash A \quad \Delta(B) \vdash C}{\Delta([B/A, \Gamma]) \vdash C}/\text{L} \qquad \dfrac{[\Gamma, A] \vdash B}{\Gamma \vdash B/A}/\text{R}$

(3) a. $A \vdash A$ id $\qquad \dfrac{\Gamma \vdash A \quad \Delta(A) \vdash B}{\Delta(\Gamma) \vdash B}\text{Cut}$

b. $\dfrac{\Gamma \vdash A \quad \Delta(B) \vdash C}{\Delta(\Gamma, A\backslash B) \vdash C}\backslash\text{L} \qquad \dfrac{A, \Gamma \vdash B}{\Gamma \vdash A\backslash B}\backslash\text{R}$

c. $\dfrac{\Gamma \vdash A \quad \Delta(B) \vdash C}{\Delta(B/A, \Gamma) \vdash C}/\text{L} \qquad \dfrac{\Gamma, A \vdash B}{\Gamma \vdash B/A}/\text{R}$

By way of example, "lifting" A $\vdash$ B/(A\B) is generated as follows in **NL**; it is similarly derivable in **L**.

(4)
$$\dfrac{\dfrac{\mathrm{A} \vdash \mathrm{A} \quad \mathrm{B} \vdash \mathrm{B}}{[\mathrm{A}, \mathrm{A}\backslash\mathrm{B}] \vdash \mathrm{B}}\backslash\mathrm{L}}{\mathrm{A} \vdash \mathrm{B}/(\mathrm{A}\backslash\mathrm{B})}/\mathrm{R}$$

On the other hand "composition" A\B, B\C $\vdash$ A\C, while derivable as follows in **L**, is **NL**-underivable in its non-associative form: [A\B, B\C] $\vdash$ A\C.

(5)

$$\dfrac{A \vdash A \qquad \dfrac{B \vdash B \qquad C \vdash C}{B,\ B\backslash C \vdash C}\backslash L}{\dfrac{A,\ A\backslash B,\ B\backslash C \vdash C}{A\backslash B,\ B\backslash C\ \vdash A\backslash C}\backslash R}\backslash L$$

1.2 Multimodal Lambek Calculi

In a slightly different formulation of the sequent calculus for **L** we may configure antecedents with binary bracketing, and then use the **NL** rules together with an explicit structural rule of associativity (the double bar indicates bidirectionality):

(6)

$$\frac{\Gamma([\Delta_1, [\Delta_2, \Delta_3]]) \vdash A}{\Gamma([[\Delta_1, \Delta_2], \Delta_3]) \vdash A}\mathrm{A}$$

From here it is a small step to give sequent calculus for "multimodal" Lambek calculi in which we have several families of connectives $\{/_i, \backslash_i\}_{i\in\{1,\ldots,n\}}$, each defined by residuation with respect to their adjunction in a "multigroupoid" $\langle L, \{+_i\}_{i\in\{1,\ldots,n\}}\rangle$ (Moortgat and Morrill 1991):

(7)

$$\begin{array}{rcl} D(A\backslash_i B) & = & \{s|\forall s' \in D(A), s'+_i s \in D(B)\} \\ D(B/_i A) & = & \{s|\forall s' \in D(A), s+_i s' \in D(B)\} \end{array}$$

Sequent calculus can be given by indexing the brackets of **NL**-presentations to indicate mode of adjunction (and adding structural rules, including interaction postulates between different modes, as appropriate):

(8)

$$\frac{}{A \vdash A}\mathrm{id} \qquad\qquad \frac{\Gamma \vdash A \qquad \Delta(A) \vdash B}{\Delta(\Gamma) \vdash B}\mathrm{Cut}$$

(9) a.

$$\frac{\Gamma \vdash A \qquad \Delta(B) \vdash C}{\Delta([_i\Gamma, A\backslash_i B]) \vdash C}\backslash_i \mathrm{L} \qquad\qquad \frac{[_i A, \Gamma] \vdash B}{\Gamma \vdash A\backslash_i B}\backslash_i \mathrm{R}$$

b.

$$\frac{\Gamma \vdash A \qquad \Delta(B) \vdash C}{\Delta([_i B/_i A, \Gamma]) \vdash C}/_i\mathrm{L} \qquad\qquad \frac{[_i\Gamma, A] \vdash B}{\Gamma \vdash B/_i A}/_i\mathrm{R}$$

In particular cases of course we may choose non-composite notations for the connectives and brackets. With two modes interpreted in a "bigroupoid" understood as distinguishing left-headed and right-headed adjunction we have a "headed" calculus (Moortgat and Morrill 1991). With families $\{/, \backslash\}$ and $\{<, >\}$ for adjunctions $+$ (associative) and $(.,.)$ (not assumed to be associative) respectively in a bigroupoid $\langle L, +, (.,.)\rangle$ we have a partially associative calculus **L+NL** (Oehrle and Zhang 1989, Morrill 1990). This latter forms two-thirds of the discontinuity calculus of Morrill (1993, 1994d) which we shall be considering.

1.3 Labelled Sequent Presentations

"Labelling" (Gabbay 1991) is a means of presenting proof theory which will enable us to factor out the antecedent formulas of a sequent, and its associated prosodic construction, which is made more explicit. No essential use of sequent labelling is made here, in that the labelled presentation of calculus is just notational variation of ordered presentation. However, labelling is a step on the path to implementing residuation calculi. We notate a sequent $\Gamma \vdash A$ as $a_1\colon A_1, \ldots, a_n\colon A_n \vdash \alpha\colon A$ where the multiset $\{A_1, \ldots, A_n\}_m$ comprises the formula occurrences in Γ, $a_1, \ldots, a_n$ are distinct atomic labels, and α is a term over these labels representing explicitly the prosodic construction that was represented implicitly by the structured configuration Γ. The labelled sequent calculus for **NL** is as follows:

(10) a. $a\colon A \vdash a\colon A \qquad \text{id}$

b. $$\frac{\Gamma \vdash \alpha\colon A \qquad a\colon A, \Delta \vdash \beta(a)\colon B}{\Gamma, \Delta \vdash \beta(\alpha)\colon B}\text{Cut}$$

c. $$\frac{\Gamma \vdash \alpha\colon A \qquad b\colon B, \Delta \vdash \gamma(b)\colon C}{\Gamma, d\colon A\backslash B, \Delta \vdash \gamma((\alpha + d))\colon C}\backslash\text{L}$$

d. $$\frac{\Gamma, a\colon A \vdash (a + \gamma)\colon B}{\Gamma \vdash \gamma\colon A\backslash B}\backslash\text{R}$$

e. $$\frac{\Gamma \vdash \alpha\colon A \qquad b\colon B, \Delta \vdash \gamma(b)\colon C}{\Gamma, d\colon B/A, \Delta \vdash \gamma((d + \alpha))\colon C}/\text{L}$$

f. $$\frac{\Gamma, a\colon A \vdash (\gamma + a)\colon B}{\Gamma \vdash \gamma\colon B/A}/\text{R}$$

To obtain **L** an associativity equation on terms may be added:

(11) $((\alpha_1{+}\alpha_2){+}\alpha_3) = (\alpha_1{+}(\alpha_2{+}\alpha_3))$

Or equivalence classes of terms represented by flattening terms into lists.

1.4 Labelled Natural Deduction

For labelled Fitch-style categorial derivation (Morrill 1993), there are lexical assignment, subderivation hypothesis, and term label equation rules thus ($\alpha - \phi\colon A$ represents assignment to type A of the paired prosodic term α and semantic term ϕ; we include Curry-Howard semantic annotation intermittently in what follows; full explications are available in the references):

(12) a. $n.\quad \alpha - \phi\colon A \quad$ for any lexical entry

b.
$$\begin{array}{ll|ll}
n. & & a_1 - x_1\colon A_1 & \text{H} \\
 & & \vdots & \vdots \\
n+m. & & a_m - x_m\colon A_n & \text{H} \\ \cline{3-3}
\end{array}$$

c.
$$\begin{array}{lll}
n. & \alpha - \phi\colon A & \\
 & \alpha' - \phi'\colon A & = n, \text{if } \alpha = \alpha' \ \&\ \phi = \phi'
\end{array}$$

The logical rules are:

(13) a.
$$\begin{array}{lll}
n. & \alpha - \phi\colon A & \\
m. & \gamma - \chi\colon A\backslash B & \\ \cline{2-2}
 & (\alpha{+}\gamma) - (\chi\ \phi)\colon B & \text{E}\backslash\ n, m
\end{array}$$

b.
$$\begin{array}{ll|l}
n. & & a - x\colon A \quad \text{H} \\ \cline{3-3}
m. & & (a{+}\gamma) - \psi\colon B \quad \text{unique } a \text{ as indicated} \\
 & \multicolumn{2}{l}{\gamma - \lambda x\psi\colon A\backslash B \quad \text{I}\backslash\ n, m}
\end{array}$$

(14) a.
$$\begin{array}{lll}
n. & \alpha - \phi\colon A & \\
m. & \gamma - \chi\colon B/A & \\ \cline{2-2}
 & (\gamma{+}\alpha) - (\chi\ \phi)\colon B & \text{E}/\ n, m
\end{array}$$

$$
\begin{array}{lll}
\text{b.} \quad n. & \quad | \; a - x\colon A & \text{H} \\
\quad\;\; m. & \quad | \; (\gamma{+}a) - \psi\colon B & \text{unique } a \text{ as indicated} \\
 & \gamma - \lambda x \psi\colon B/A & \text{I}/\ n, m
\end{array}
$$

A Fitch-style labelled calculus for the associative Lambek calculus **L** can be obtained from that for the non-associative calculus by adding a prosodic label equation. Alternatively, the associative Lambek calculus can be given by dropping parentheses in prosodic labels. By way of example, a simple instance of relativisation can be derived by hypothetical deduction as follows:

(15)

$$
\begin{array}{lll}
1. & \mathit{which} - \lambda x \lambda y \lambda z[(y\ z) \wedge (x\ z)]\colon (\mathrm{CN}\backslash\mathrm{CN})/(\mathrm{S}/\mathrm{N}) & \\
2. & \mathit{John} - \mathbf{j}\colon \mathrm{N} & \\
3. & \mathit{talked} - \mathbf{talk}\colon (\mathrm{N}\backslash\mathrm{S})/\mathrm{PP} & \\
4. & \mathit{about} - \mathbf{about}\colon \mathrm{PP}/\mathrm{N} & \\
5. & | \; a - x\colon \mathrm{N} & \text{H} \\
6. & | \; \mathit{about}{+}a - (\mathbf{about}\ x)\colon \mathrm{PP} & 4, 5\ \text{E}/ \\
7. & | \; \mathit{talked}{+}\mathit{about}{+}a - (\mathbf{talk}\ (\mathbf{about}\ x))\colon \mathrm{N}\backslash\mathrm{S} & 3, 6\ \text{E}/ \\
8. & | \; \mathit{John}{+}\mathit{talked}{+}\mathit{about}{+}a - ((\mathbf{talk}\ (\mathbf{about}\ x))\ \mathbf{j})\colon \mathrm{S} & 2, 7\ \text{E}\backslash \\
9. & \mathit{John}{+}\mathit{talked}{+}\mathit{about} - \lambda x((\mathbf{talk}\ (\mathbf{about}\ x))\ \mathbf{j})\colon \mathrm{S}/\mathrm{N} & 5, 8\ \text{I}/ \\
10. & \mathit{which}{+}\mathit{John}{+}\mathit{talked}{+}\mathit{about} - & \\
 & (\lambda x \lambda y \lambda z[(y\ z) \wedge (x\ z)]\ \lambda x((\mathbf{talk}\ (\mathbf{about}\ x))\ \mathbf{j}))\colon \mathrm{CN}\backslash\mathrm{CN} & 1, 9\ \text{E}/ \\
11. & \mathit{which}{+}\mathit{John}{+}\mathit{talked}{+}\mathit{about} - & \\
 & \lambda y \lambda z[(y\ z) \wedge ((\mathbf{talk}\ (\mathbf{about}\ z))\ \mathbf{j})]\colon \mathrm{CN}\backslash\mathrm{CN} & = 10
\end{array}
$$

1.4.1 Multimodal Fitch natural deduction

Multimodal calculi can be presented Fitch-style by giving the same rules for each family of connectives with their associated adjunctions:

(16)

$$
\begin{array}{lll}
\text{a.} \quad n. & \alpha - \phi\colon A & \\
\quad\;\; m. & \gamma - \chi\colon A\backslash_i B & \\
\hline
 & (\alpha{+}_i\gamma) - (\chi\ \phi)\colon B & \text{E}\backslash_i\ n, m
\end{array}
$$

$$
\begin{array}{lll}
\text{b.} \quad n. & \quad | \; a - x\colon A & \text{H} \\
\quad\;\; m. & \quad | \; (a{+}_i\gamma) - \psi\colon B & \text{unique } a \text{ as indicated} \\
 & \gamma - \lambda x \psi\colon A\backslash_i B & \text{I}\backslash_i\ n, m
\end{array}
$$

(17)

$$
\begin{array}{lll}
\text{a.} \quad n. & \alpha - \phi\colon A & \\
\quad\;\; m. & \gamma - \chi\colon B/_i A & \\
\hline
 & (\gamma{+}_i\alpha) - (\chi\ \phi)\colon B & \text{E}/_i\ n, m
\end{array}
$$

$$
\begin{array}{lll}
\text{b.} \quad n. & \quad | \; a - x\colon A & \text{H} \\
\quad\;\; m. & \quad | \; (\gamma{+}_i a) - \psi\colon B & \text{unique } a \text{ as indicated} \\
 & \gamma - \lambda x \psi\colon B/_i A & \text{I}/_i\ n, m
\end{array}
$$

Label equations are to be added according to the algebras of interpretation.

1.4.2 Multimodal labelled Prawitz natural deduction

Labelled deduction can also be presented Prawitz-style; for the multimodal case (without semantics) there is the following.

(18)

$$
\dfrac{\begin{array}{cc}\vdots & \vdots \\ \gamma\colon B/_i A & \alpha\colon A\end{array}}{\gamma{+}_i\alpha\colon B}/_i\text{E}
\qquad\qquad
\dfrac{\begin{array}{cc}\vdots & \vdots \\ \alpha\colon A & \gamma\colon A\backslash_i B\end{array}}{\alpha{+}_i\gamma\colon B}\backslash_i\text{E}
$$

(19)

$$\dfrac{\begin{array}{cc} \vdots & \overline{a\colon A}^{\,n} \\ \gamma{+_i}a\colon B & \end{array}}{\gamma\colon B/_iA}/_i\mathrm{I}^n \qquad\qquad \dfrac{\begin{array}{cc} \overline{a\colon A}^{\,n} & \vdots \\ & a{+_i}\gamma\colon B \end{array}}{\gamma\colon A\backslash_i B}\backslash_i\mathrm{I}^n$$

Each of the two styles of natural deduction have their merits so far as presentation of proofs is concerned: Fitch-style reasons serially while Prawitz-style indicates parallel (unordered) branches of inference; both avoid the sequent calculus reiteration of context formulas and both are practical for linguistic derivations. However, reading from premises to conclusion natural deduction offers only retrospective justification of hypothetical assumptions: looking at the premises only we do not know which hypotheses might turn out to be useful. Sequent calculus is superior so far as proof discovery, as oppposed to presentation, is concerned because it shows which hypotheses are worth trying.

2 Automated Deduction

Given Cut-elimination (the property that every theorem has a Cut-free proof; Lambek 1958, 1961 showed this for his calculi) the sequent calculi give decision procedures for determining whether a given sequent is a theorem. Backward-chaining Cut-free labelled sequent proof search admits only a finite number of possible rule applications for a given sequent, eliminating the principle connective of one of the (finite number of) formulas, and choosing (one of the finite number of) antecedent partitionings in the case of binary rules. This creates subgoals the complexity of which in terms of connective occurrences totals exactly one connective occurrence less. Thus for any sequent there is a finite space of proof search.

There are two sources of non-determinism however in (Cut-free) backward-chaining sequent proof search: in choosing on which formula to key rule application, and in choosing how to partition sequents in binary rules. With respect to the former, different sequences of choice can converge on the same subproblems; with respect to the latter, considerable space may need to be searched before determining whether a partitioning terminated in initial identity axiom sequents or not. The former, but not the latter, problem is addressed by "proof normalisation" (for the case of Lambek calculus see Hepple 1990, which refines König 1989, see also Hendriks 1993): fixing priorities of rule ordering to determine distinguished representatives of equivalence classes of proofs. Both drawbacks are addressed by proof nets (in linear logic), and the matrix methods of Bibel (1981) and Wallen (1990) (Bibel and Eder 1993). In these, formulas are unfolded, and proofs built *from the initial sequents*. Through unfolding, the parts comprising a formula are made available for examination in parallel, rather than only in serial according to the particular nesting of connectives. By building from initial sequents we ensure effectively that only rule applications are tried which are already known to terminate successfully in initial sequents.

In the context of linear logic then proof nets have been developed as a method of eliminating redundancy in the sequent representation of proofs. For these however a correctness check (the "long trip condition") is required. Corresponding proposals have been made for Lambek calculus by Roorda (1991), in which correctness is checked by semantic labelling, and Roorda (1991) and Moortgat (1990, 1992), in which correctness is checked by prosodic labelling. As such however, the method reduces proof net correctness to checking, by unification, satisfiability of equations in groupoids, semigroups, and so on. Yet such problems as semigroup unification are in general intractable, even though the sequent formulations of the calculi show decidability. Somewhere the method loses control of constraints, and improved management is required in order to achieve efficiency. We shall provide the necessary structure by organising the proof nets used as clauses, in fact (higher order) Horn clauses, of linear logic, for which a resolution strategy is available in which at each unification step one term is ground, i.e. variable-free. This prepares the way for computational theorem proving in residuation calculi generally and illustration includes regular and head-oriented discontinuity calculi.

3 Sequent Calculus for Classical Linear Logic

The multiplicative fragment of linear logic, with which we shall be concerned, contains binary infix connectives $\otimes$ (a conjunction "times") and $\maltese$ (a disjunction "par") and a unary postfix negation $^{\perp}$ ("neg"). Sequents are of the form $\Gamma \vdash \Delta$ where configurations Γ and Δ are sequences of zero or more formulas. There are the following sequent rules, which are sound for classical logic, (and which would also be complete for classical logic if the structural rules of contraction and weakening were included). The calculus enjoys Cut-elimination.

(20) a. $$\frac{}{A \vdash A}\text{id}$$ b. $$\frac{\Gamma \vdash A, \Delta \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\text{Cut}$$

(21) a. $$\frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta}\text{P}_L$$ b. $$\frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, B, A, \Delta'}\text{P}_R$$

(22) a. $$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\otimes\text{L}$$ b. $$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\otimes\text{R}$$

(23) a. $$\frac{A, \Delta \vdash \Gamma \qquad B, \Delta' \vdash \Gamma'}{A \maltese B, \Delta, \Delta' \vdash \Gamma, \Gamma'}\maltese\text{L}$$ b. $$\frac{\Delta \vdash \Gamma, A, B}{\Delta \vdash \Gamma, A \maltese B}\maltese\text{R}$$

(24) a. $$\frac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta}{}^{\perp}\text{L}$$ b. $$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta}{}^{\perp}\text{R}$$

The linear implication $\multimap$ is defined by $A \multimap B = A^{\perp} \maltese B$, so any $\{\otimes, \maltese, ^{\perp}, \multimap\}$ formula can be considered an abbreviation for a $\{\otimes, \maltese, ^{\perp}\}$ one.

3.1 Negation Normal Form

A $\{\otimes, \maltese, ^{\perp}\}$ formula is in negation normal form if and only if no connective occurrence falls within the scope of a negation, i.e. negations may only be immediately attached to (unnegated) atoms. We have the following proofs of the involution of negation:

(25) a. $$\frac{\dfrac{A \vdash A}{A, A^{\perp} \vdash}{}^{\perp}\text{L}}{A \vdash A^{\perp\perp}}{}^{\perp}\text{R}$$ b. $$\frac{\dfrac{A \vdash A}{\vdash A^{\perp}, A}{}^{\perp}\text{R}}{A^{\perp\perp} \vdash A}{}^{\perp}\text{L}$$

And there are the following proofs of a de Morgan law:

(26) a. $$\frac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, A^{\perp}}{}^{\perp}\text{R} \quad \dfrac{B \vdash B}{\vdash B, B^{\perp}}{}^{\perp}\text{L}}{\vdash A \otimes B, A^{\perp}, B^{\perp}}\otimes\text{R}}{\vdash A \otimes B, A^{\perp} \maltese B^{\perp}}\maltese\text{R}}{(A \otimes B)^{\perp} \vdash A^{\perp} \maltese B^{\perp}}{}^{\perp}\text{L}$$ b. $$\frac{\dfrac{\dfrac{\dfrac{A \vdash A}{A^{\perp}, A \vdash}{}^{\perp}\text{L} \quad \dfrac{B \vdash B}{B^{\perp}, B \vdash}{}^{\perp}\text{L}}{A^{\perp} \maltese B^{\perp}, A, B \vdash}\maltese\text{L}}{A^{\perp} \maltese B^{\perp}, A \otimes B \vdash}\otimes\text{L}}{A^{\perp} \maltese B^{\perp} \vdash (A \otimes B)^{\perp}}{}^{\perp}\text{R}$$

The other de Morgan law is obtained similarly. Hence using the equivalences (27) to reduce redexes of the form on the left to contractums of the form on the right, any formula is converted to a negation normal form.

(27)
$$
\begin{array}{rcl}
A^{\perp\perp} & = & A \\
(A \otimes B)^{\perp} & = & A^{\perp} \parr B^{\perp} \\
(A \parr B)^{\perp} & = & A^{\perp} \otimes B^{\perp}
\end{array}
$$

So $^{\perp}$ need not really be considered a connective in formulas but as a means of abbreviating $\{\otimes, \parr\}$ formulas in which atoms come in two flavours: positive and negative.

4 Proof Nets for Linear Logic

To validate a sequent using proof nets all formulas are first put on one side of the sequent turnstyle using the negation rules:

(28) $\Gamma \vdash \Delta$ if and only if $\Gamma, \Delta^{\perp} \vdash$ if and only if $\vdash \Gamma^{\perp}, \Delta$

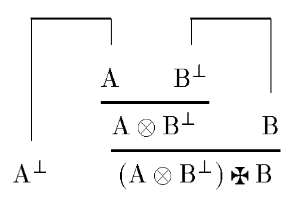
They are converted to negation normal form, after which a phase of "unfolding" ensues:

(29) a. $\dfrac{A \quad B}{A \otimes B}$ b. $\dfrac{A \quad B}{A \parr B}$

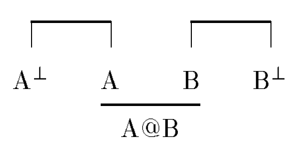
We refer to the result of unfolding until all leaves are atomic as a *proof frame*. Then an attempt must be made to connect (or: link) all the (positive and negative) atoms in such a way that each is connected with exactly one other, of complementary polarity. Such connections correspond to initial sequent axioms. A connection of all the atoms is called a *proof structure*. The existence of a proof structure is a necessary, but not sufficient, condition for theoremhood (note for example that conjunction and disjunction are not distinguished!): we shall further require the "long trip condition" which effectively checks partitioning, i.e. that we have connected as initial sequent axioms atomic formula occurrences which were meant to be in the same sequent subproofs, and not ones meant to be in different subproofs.

By way of example of unfolding and linking, there is the following:

(30)
$$
\begin{array}{ll}
\mathrm{A} \vdash (\mathrm{A} \multimap \mathrm{B}) \multimap \mathrm{B} & \text{if and only if} \\
\mathrm{A} \vdash (\mathrm{A}^{\perp} \parr \mathrm{B})^{\perp} \parr \mathrm{B} & \text{if and only if} \\
\vdash \mathrm{A}^{\perp}, (\mathrm{A} \otimes \mathrm{B}^{\perp}) \parr \mathrm{B} &
\end{array}
$$

(31)


The proof structure (31) is also a proof net. The proof structure (32) is a proof net for @= $\otimes$ (A, B $\vdash$ A $\otimes$ B is a theorem) but not for @= $\parr$ (A, B $\vdash$ A $\parr$ B is not a theorem).

(32)


A restriction to *planar* proof nets, ones with nested (i.e. non-crossing) connections, characterises (together with the long trip condition) the theorems of *cyclic* linear logic, i.e. linear logic in which exchange is limited to circular permutations.

5 Proof Nets for Lambek Calculus

In the previous section proof nets were given by moving all formulas to the right of the sequent turnstyle. Here we shall do the converse: move all formulas to the left, that is we shall perform refutation proofs. From considerations of symmetry we see that the choice is not important, but it will enable us to present our proposal in the familiar context of resolution refutation. We consider the presentation of proof nets for Lambek calculus of Roorda (1991), but our polarities are reversed. Formulas composed from the implicational connectives / and \ are signed positive for antecedent occurrences and negative for succedent occurrences, and unfolded as follows.

(33)
$$\frac{B^+ \quad A^-}{B/A^+} \qquad \frac{A^- \quad B^+}{A\backslash B^+} \qquad \frac{A^+ \quad B^-}{B/A^-} \qquad \frac{B^- \quad A^+}{A\backslash B^-}$$

The transmission of polarities can be understood when we see an implication as the disjunction of its consequent with the negation of its antecedent. The steps given are compilations of decomposition accordingly, with unfolding, involution of negation, and de Morgan laws for multiplicative conjunction or disjunction compiled in. The ordering given, which swaps the components of negative (i.e. succedent) occurrences of implications allows restriction to planar connections.

The following, for example, are proof nets for lifting A $\vdash$ B/(A\B) and composition A\B, B\C $\vdash$ A\C in **L**.

(34)
$$\mathrm{A}^+ \qquad \frac{\dfrac{\mathrm{A}^- \quad \mathrm{B}^+}{\mathrm{A}\backslash\mathrm{B}^+} \quad \mathrm{B}^-}{\mathrm{B}/(\mathrm{A}\backslash\mathrm{B})^-}$$

(35)
$$\frac{\mathrm{A}^- \quad \mathrm{B}^+}{\mathrm{A}\backslash\mathrm{B}^+} \qquad \frac{\mathrm{B}^- \quad \mathrm{C}^+}{\mathrm{B}\backslash\mathrm{C}^+} \qquad \frac{\mathrm{C}^- \quad \mathrm{A}^+}{\mathrm{A}\backslash\mathrm{C}^-}$$

Just by considerations of symmetry however we can see that there will also be a proof structure for the invalid "lowering": B/(A\B) $\vdash$ A. A long trip condition can express the required constraint. Roorda (1991) expresses such a condition in terms of the lambda terms that are notational variants of intuitionistic natural deductions corresponding to Lambek proofs, and which provide the semantic dimension of categorial logic. But which condition on axiom linking corresponds to **NL**? How would we express the **L** + **NL** hybrid? And what about multimodal calculi with interaction postulates? Since these varieties relate more directly to the groupoid prosodic dimension we shall follow Roorda/Moortgat in using prosodic terms to express the correctness conditions.

5.1 Labelled Proof Nets

Roorda (1991) and Moortgat (1990, 1992) present unfolding with prosodic labelling as follows (see also Hendriks 1993 and Oehrle 1994).

(36) a.
$$\frac{\gamma{+}a\colon B^+ \quad a\colon A^-}{\gamma\colon B/A^+} \qquad \frac{a\colon A^- \quad a{+}\gamma\colon B^+}{\gamma\colon A\backslash B^+} \qquad a \text{ new variable}$$

b.
$$\frac{k\colon A^+ \quad \gamma{+}k\colon B^-}{\gamma\colon B/A^-} \qquad \frac{k{+}\gamma\colon B^- \quad k\colon A^+}{\gamma\colon A\backslash B^-} \qquad k \text{ new constant}$$

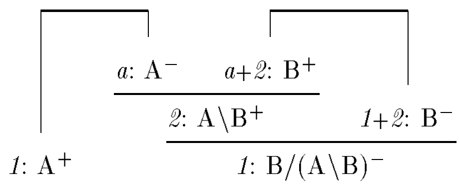
The succedent (negative) unfoldings introduce (Skolem-like) constants. The antecedent (positive) unfoldings introduce variables. Linking identifies the labels of linked atoms and the correctness condition is that a proof structure is a proof net if and only if the set of equations induced by linking is satisfiable. This can be checked by unification.

Consider lifting, for which we assume labelling thus by a constant *1*:

(37) 1: A $\vdash$ 1: B/(A\B)

Then there is the proof net (38).

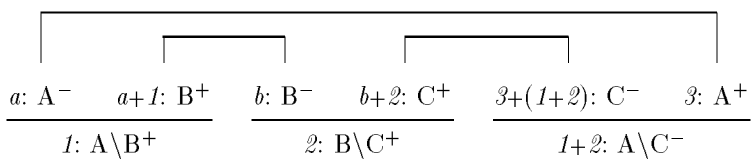
(38)



The linking yields the equations $1{=}a$ and $a{+}2{=}\ 1{+}2$ which are clearly satisfied. For composition as in (39) we obtain (40).

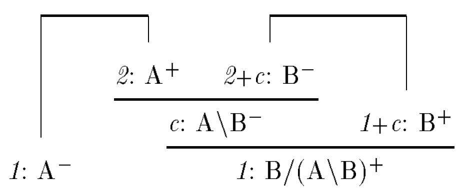
(39) 1: A\B, 2: B\C $\vdash$ $1{+}2$: A\C

(40)



This yields the equations $a{=}3$, $a{+}1{=}b$ and $b{+}2\ =\ 3{+}(1{+}2)$ which are satisfied with $b{=}3{+}1$ in the associative case, but not in the non-associative one. For the invalid lowering however we have:

(41)



The equations $1{=}2$ and $1{+}c{=}2{+}c$ are clearly not satisfiable since they require the identification of two distinct constants.

The method is attractive because we can see how it adapts to different residuation calculi, such as the partially associative calculus **L+NL**, just by unifying prosodic terms according to the laws of the groupoid algebras of interpretation. Such direct association of proof theory with interpretation is precisely the point of labelling: "bringing semantics back into syntax". Generality is obtained because we have identified the highest common factor of sublinear residuation calculi, linear validity. The degrees of variation are in effect built in as non-logical properties of term structure of quantifier-free first-order linear logic: unfolded signed labelled atoms (literals) α: A^+ and β: B^- are atomic formulas of predicational linear logic $A(\alpha)$ and $B(\beta)^\perp$ respectively. Linking corresponds to an application of the resolution principle, together with unification of terms. It is this relation which we shall exploit to resolve the computational shortcoming of the method as it stands: that testing satisfiability of the linking equations appears to demand solution to such problems as semigroup unification, which are quite intractable.

5.2 Clausal Proofs

In trying to shed light on how to compute the relevant labelled proof nets it is instructive to identify their rational: why they represent the relevant conditions. Recall the groupoid interpretation of the Lambek connectives:

$$(42)\quad \begin{array}{lcl} D(A\backslash B) & = & \{s|\forall s' \in D(A), s'{+}s \in D(B)\} \\ D(B/A) & = & \{s|\forall s' \in D(A), s{+}s' \in D(B)\} \end{array}$$

In labelled calculi categorial type assignment statements comprise a groupoid term α and a type A; we write α: A. Now we can formalise the metalanguage of the interpretation clauses in labels in order to translate type assignments to complex categorial types to statements of first order logic:

$$(43)\quad \begin{array}{lll} \gamma\colon A\backslash B & \text{iff} & \forall a[a\colon A \rightarrow a{+}\gamma\colon B] \\ \gamma\colon B/A & \text{iff} & \forall a[a\colon A \rightarrow \gamma{+}a\colon B] \end{array}$$

When we translate the type assignment statements of a labelled sequent in this way the problem of categorial theorem proving is compiled into that of theorem proving for a particular fragment of first order logic. The quantifiers may turn out to be in positive or negative positions according to antecedent or succedent origen, and nesting within the negations implicit in implications. Hence the object variables correspond to either metavariables or Skolem constants in clausal refutation, and this is predictable from the transmission of polarities at the time of compilation. This explains the Roorda/Moortgat unfolding so far as it goes, but also suggests how to go further, preserving the relations between parts of formulas in such a way that a flow of information can be exploited. We can therefore adapt the compilation to include the implication (written in the logic programming direction), which is linear, dealing with occurrences. The polar translation functions are identity functions on atomic assignments; on complex category predicates they are defined mutually as follows; $\overline{p}$ indicates the polarity complementary to p:

$$(44)\quad \text{a.}\quad \frac{\alpha{+}\gamma\colon B^p \quad \circ\!\!- \quad \alpha\colon A^{\overline{p}}}{\gamma\colon A\backslash B^p}\,\alpha \text{ new variable/constant as } p\ {+}/{-}$$

$$\text{b.}\quad \frac{\gamma{+}\alpha\colon B^p \quad \circ\!\!- \quad \alpha\colon A^{\overline{p}}}{\gamma\colon B/A^p}\,\alpha \text{ new variable/constant as } p\ {+}/{-}$$

The unfolding transformations have the same general form for the positive (configuration/database) and negative (succedent/agenda) occurrences; the polarity is used to indicate whether new symbols introduced for quantified variables in the interpretation clauses are metavariables or Skolem constants. The result of compilation is contained in a higher order logic programming fragment. The program clauses and agenda are read directly off the unfoldings, with the only manipulation being a flattening of positive implications into uncurried form:

$$(45)\quad ((X^+ \circ\!\!- Y_1^-) \circ\!\!- \ldots) \circ\!\!- Y_n^- \quad \triangleright \quad X^+ \circ\!\!- Y_1^- \otimes \ldots \otimes Y_n^-$$

This means that matching against the head and assembly of subgoals does not require recursion and restructuring as run time. We shall also allow unit program clauses $X \circ\!\!-$ to be abbreviated X.

We define this fragment and a suitable execution mechanism. Assume a set $\mathcal{ATOM}$ of atomic formulas, 0-ary, 1-ary, etc., formula constructors $\{\cdot \otimes \ldots \otimes \cdot\}_{n \in \{0,1,\ldots\}}$ and a binary (infix) formula constructor $\circ\!\!-$. A sequent comprises an agenda formula A and a bag (or: multiset) database $\Gamma = \{B_1, \ldots, B_n\}_m, n \geq 0$ of program clauses; we write $\Gamma \vdash A$. The set $\mathcal{AGENDA}$ of agendas is defined by:

$$(46)\quad \mathcal{AGENDA} ::= \mathcal{GOAL} \otimes \ldots \otimes \mathcal{GOAL}$$

The set $\mathcal{PCLS}$ of program clauses is defined by:

$$(47)\quad \mathcal{PCLS} ::= \mathcal{ATOM} \circ\!\!- \mathcal{AGENDA}$$

For first order programming the set $\mathcal{GOAL}$ of goals is defined by:

$$(48)\quad \mathcal{GOAL} ::= \mathcal{ATOM}$$

For the higher order case the notion of $\mathcal{GOAL}$ is generalised to include implications:

(49) $\mathcal{GOAL} ::= \mathcal{ATOM} \mid \mathcal{GOAL} \multimapinv \mathcal{PCLS}$

In linear logic programming each rule is resource-conscious. The termination condition is:

(50) $$\frac{}{A \vdash A}\text{id}$$

I.e. an atomic agenda is a consequence of its unit database; all program clauses must be "used up" by the resolution rule:

(51) $$\frac{\Gamma \vdash B_1 \otimes \ldots \otimes B_n \otimes C_1 \otimes \ldots \otimes C_m}{\Gamma, A \multimapinv B_1 \otimes \ldots \otimes B_n \vdash A \otimes C_1 \otimes \ldots \otimes C_m}\text{RES}$$

I.e. a program clause disappears from the database once it is resolved upon: each is used exactly once. The deduction theorem rule for higher-order clauses is also sensitised to the employment of antecedent contexts:

(52) $$\frac{\Gamma, B \vdash A \qquad \Delta \vdash C_1 \otimes \ldots \otimes C_m}{\Gamma, \Delta \vdash (A \multimapinv B) \otimes C_1 \otimes \ldots \otimes C_m}\text{DT}$$

With respect to unification, the goal in resolution will always be ground so that potentially problematic unification problems reduce to simpler one way cases. The DT rule appears to need to hypothesise partitionings, however the form of compilation is such that A and B share a Skolem constant; so B can and must be used to prove A and a mechanism for the lazy splitting of contexts can be effected.

Starting from the initial database and agenda, a proof will be represented as a list of agendas, avoiding the context repetition of sequent proofs by indicating where the resolution rule retracts from the database (superscript coindexed overline), and where the deduction theorem rule adds to it (subscript coindexation):

(53)

database	$\Gamma, \overline{A \multimapinv B_1 \otimes \ldots \otimes B_n}^{i}$	
agenda		
i.	$A \otimes C_1 \otimes \ldots \otimes C_n$	RES
i+1.	$B_1 \otimes \ldots \otimes B_n \otimes C_1 \otimes \ldots \otimes C_m$	

(54)

database	Γ, B_i	
agenda		
i.	$(A \multimapinv B) \otimes C_1 \otimes \ldots \otimes C_m$	DT
i+1.	$A \otimes C_1 \otimes \ldots \otimes C_m$	

Consider the case of lifting:

(55) **k**: A $\vdash$ **k**: B/(A\B)

The succedent unfolds as shown in (56).

(56) $$\frac{\mathbf{k}{+}\mathbf{l}\text{: B} \quad \multimapinv \quad \dfrac{a{+}\mathbf{l}\text{: B} \quad \multimapinv \quad a\text{: A}}{\mathbf{l}\text{: A}\backslash\text{B}^{+}}}{\mathbf{k}\text{: B}/(\text{A}\backslash\text{B})^{-}}$$

Then the proof runs thus:

$$(57)\quad \begin{array}{lll} \text{database} & \overline{\mathbf{k}\colon A}^{3}, & \\ & \overline{a{+}\mathbf{l}\colon \mathrm{B} \circ\!- a\colon A_1}^{2} & \\ \text{agenda} & & \\ 1. & \mathbf{k}{+}\mathbf{l}\colon \mathrm{B} \circ\!- (a{+}\mathbf{l}\colon \mathrm{B} \circ\!- a\colon A) & \mathrm{DT} \\ 2. & \mathbf{k}{+}\mathbf{l}\colon \mathrm{B} & \mathrm{RES}\ a = \mathbf{k} \\ 3. & \mathbf{k}\colon \mathrm{A} & \end{array}$$

The unification is simply term unification: lifting is **NL**- and **L**-valid. Composition depends on associativity and is only **L**-valid:

$$(58)\quad \mathbf{k}\colon \mathrm{A}\backslash\mathrm{B}^{+},\ \mathbf{l}\colon \mathrm{B}\backslash\mathrm{C}^{+} \vdash \mathbf{k}{+}\mathbf{l}\colon \mathrm{A}\backslash\mathrm{C}^{-}$$

$$(59)\quad \frac{a{+}\mathbf{k}\colon \mathrm{B} \quad \circ\!- \quad a\colon \mathrm{A}}{\mathbf{k}\colon \mathrm{A}\backslash\mathrm{B}^{+}} \qquad \frac{b{+}\mathbf{l}\colon \mathrm{C} \quad \circ\!- \quad b\colon \mathrm{B}}{\mathbf{l}\colon \mathrm{B}\backslash\mathrm{C}^{+}} \qquad \frac{\mathbf{m}{+}\mathbf{k}{+}\mathbf{l}\colon \mathrm{C} \quad \circ\!- \quad \mathbf{m}\colon \mathrm{A}}{\mathbf{k}{+}\mathbf{l}\colon \mathrm{A}\backslash\mathrm{C}^{-}}$$

$$(60)\quad \begin{array}{lll} \text{database} & \overline{a{+}\mathbf{k}\colon \mathrm{B} \circ\!- a\colon \mathrm{A}}^{3}, & \\ & \overline{b{+}\mathbf{l}\colon \mathrm{C} \circ\!- b\colon \mathrm{B}}^{2}, & \\ & \overline{\mathbf{m}\colon \mathrm{A}_1}^{4} & \\ \text{agenda} & & \\ 1. & \mathbf{m}{+}\mathbf{k}{+}\mathbf{l}\colon \mathrm{C} \circ\!- \mathbf{m}\colon \mathrm{A} & \mathrm{DT} \\ 2. & \mathbf{m}{+}\mathbf{k}{+}\mathbf{l}\colon \mathrm{C} & \mathrm{RES}\ b = \mathbf{m}{+}\mathbf{k} \\ 3. & \mathbf{m}{+}\mathbf{k}\colon \mathrm{B} & \mathrm{RES}\ a = \mathbf{m} \\ 4. & \mathbf{m}\colon \mathrm{A} & \end{array}$$

6 Linguistic Examples

Linguistic application cannot be explained in a few lines; for motivation see for example Moorgat (1988) and Morrill (1994d). Illustration should be instructive however given even a little familiarity. We consider linguistic examples starting with pure Lambek fragments. We shall then go on to multimodality and discontinuity. There is the following derivation of 'John walks' as a sentence in **L** or **NL**.

$$(61)\quad \begin{array}{l} \mathbf{John}\colon \mathrm{N}^{+} \\ \mathbf{walks}\colon \mathrm{N}\backslash\mathrm{S}^{+} = a{+}\mathbf{walks}\colon \mathrm{S}^{+} \circ\!- a\colon \mathrm{N}^{-} \end{array}$$

$$(62)\quad \begin{array}{lll} \text{database} & \overline{\mathbf{John}\colon \mathrm{N}}^{2}, & \\ & \overline{a{+}\mathbf{walks}\colon \mathrm{S} \circ\!- a\colon \mathrm{N}}^{1} & \\ \text{agenda} & & \\ 1. & \mathbf{John}{+}\mathbf{walks}\colon \mathrm{S} & \mathrm{RES}\ a{=}\mathbf{John} \\ 2. & \mathbf{John}\colon \mathrm{N} & \end{array}$$

In the derivation of 'John likes Bill' the transitive verb gives rise to an agenda of length two. Assuming an associative context, parentheses are omitted from the prosodic terms.

$$(63)\quad \begin{array}{l} \mathbf{John}\colon \mathrm{N}^{+} \\ \mathbf{likes}\colon (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}^{+} = \mathbf{likes}{+}a\colon \mathrm{N}\backslash\mathrm{S}^{+} \circ\!- a\colon \mathrm{N}^{-} = (b{+}\mathbf{likes}{+}a\colon \mathrm{S}^{+} \circ\!- b\colon \mathrm{N}^{-}) \circ\!- a\colon \mathrm{N}^{-} \end{array}$$

$$(64)\quad \begin{array}{lll} \text{database} & \overline{\mathbf{John}\colon \mathrm{N}}^{2}, & \\ & \overline{b{+}\mathbf{likes}{+}a\colon \mathrm{S} \circ\!- b\colon \mathrm{N} \otimes a\colon \mathrm{N}}^{1}, & \\ & \overline{\mathbf{Bill}\colon \mathrm{N}}^{3} & \\ \text{agenda} & & \\ 1. & \mathbf{John}{+}\mathbf{likes}{+}\mathbf{Bill}\colon \mathrm{S} & \mathrm{RES}\ b{=}\mathbf{John},\ a{=}\mathbf{Bill} \\ 2. & \mathbf{John}\colon \mathrm{N} \otimes \mathbf{Bill}\colon \mathrm{N} & \mathrm{RES} \\ 3. & \mathbf{Bill}\colon \mathrm{N} & \end{array}$$

The next example 'John will walk' illustrates the effect for an auxiliary verb treated as a functor over a verb phrase, which is itself a functor. The auxiliary creates an antecedent literal labelled by a Skolem constant, and this resolves with the subject literal of the embedded verb phrase.

(65) $\textbf{will}\text{: } (N\backslash S)/(N\backslash S)^{+} = \textbf{will}+a\text{: } N\backslash S^{+} \circ\!\!- a\text{: } N\backslash S^{-} =$
$(b+\textbf{will}+a\text{: } S^{+} \circ\!\!- b\text{: } N^{-}) \circ\!\!- (\textbf{k}+a\text{: } S^{-} \circ\!\!- \textbf{k}\text{: } N^{+})$

(66)

database	$\overline{\textbf{John}\text{: N}}^{2}$,	
	$\overline{b+\textbf{will}+a\text{: S} \circ\!\!- b\text{: N} \otimes (\textbf{k}+a\text{: S} \circ\!\!- \textbf{k}\text{: N})}^{1}$,	
	$\overline{c+\textbf{walk}\text{: S} \circ\!\!- c\text{: N}}^{4}$,	
	$\overline{\textbf{k}\text{: N}_3}^{5}$	
agenda		
1.	$\textbf{John+will+walk}$: S	RES b=**John**, a=**walk**
2.	**John**: N $\otimes$ (**k+walk**: S $\circ\!\!-$ **k**: N)	RES
3.	**k+walk**: S $\circ\!\!-$ **k**: N	DT
4.	**k+walk**: S	RES c=**k**
5.	**k**: N	

For the following minimal example of object relativisation 'which John likes' associativity is essential. The relative pronoun is a higher order functor and the positive antecedent literal arising from its argument corresponds to an "empty category" or "trace" of the extraction. But note that such a literal arose in the non-extraction example (66) also.

(67) $\textbf{which}\text{: } R/(S/N)^{+} = \textbf{which}+a\text{: } R^{+} \circ\!\!- a\text{: } S/N^{-} =$
$\textbf{which}+a\text{: } R^{+} \circ\!\!- (a+\textbf{k}\text{: } S^{-} \circ\!\!- \textbf{k}\text{: } N^{+})$

(68)

database	$\overline{\textbf{which}+a\text{: R} \circ\!\!- (a+\textbf{k}\text{: S} \circ\!\!- \textbf{k}\text{: N})}^{1}$,	
	$\overline{\textbf{John}\text{: N}}^{4}$,	
	$\overline{b+\textbf{likes}+c\text{: S} \circ\!\!- b\text{: N} \otimes c\text{: N}}^{3}$,	
	$\overline{\textbf{k}\text{: N}_2}^{5}$	
agenda		
1.	**which+John+likes**: R	RES a=**John+likes**
2.	**John+likes+k**: S $\circ\!\!-$ **k**: N	DT
3.	**John+likes+k**: S	RES b=**John**, c=**k**
4.	**John**: N $\otimes$ **k**: N	RES
5.	**k**: N	

7 Multimodal Lambek calculi

In multimodal calculi families of connectives $\{/_i, \backslash_i\}_{i \in \{1,\ldots,n\}}$ are each defined by residuation with respect to their adjunction in a "polygroupoid" $\langle L, \{+_i\}_{i\in\{1,\ldots,n\}}\rangle$ (Moortgat and Morrill 1991):

(69)
$$\begin{array}{rcl} D(A\backslash_i B) & = & \{s \mid \forall s' \in D(A), s'+_i s \in D(B)\} \\ D(B/_i A) & = & \{s \mid \forall s' \in D(A), s+_i s' \in D(B)\} \end{array}$$

Multimodal groupoid compilation is immediate:

(70) a.
$$\frac{\alpha+_i\gamma\text{: } B^p \quad \circ\!\!- \quad \alpha\text{: } A^{\overline{p}}}{\gamma\text{: } A\backslash_i B^p}\alpha \text{ new variable/constant as } p +/-$$

b.
$$\frac{\gamma+_i\alpha\text{: } B^p \quad \circ\!\!- \quad \alpha\text{: } A^{\overline{p}}}{\gamma\text{: } B/_i A^p}\alpha \text{ new variable/constant as } p +/-$$

This is entirely general. Any multimodal calculus can be implemented this way provided we have a (just one way) unification algorithm specialised according to the structural communication axioms. By way of example we shall deal with multimodality for discontinuity which involves varying internal structural properties (associativity vs. non-associativity) as well as "split/wrap" interaction between modes. We shall also consider unary operators projecting bracketed string structure and their use for head-oriented discontinuity. But before considering these we look at the simple multimodality of the partially associative calculus **L+NL** with operators / and \ defined with respect to +, and $<$ and $>$ defined with respect to $(.,.)$ in a bigroupoid $\langle L, +, (.,.)\rangle$ in which + (but not $(.,.)$) is associative. Assignment of a coordinator to (S>S)/S characterises a coordinate structure as a non-associative domain. For example in the following the complementised sentence 'that it rains and it shines' is analysed as containing a domain ['it rains and it shines'] composed of subconstituents 'it rains' and 'and it shines' (the latter is itself unstructured).

(71) **that**: CP/S$^+$= **that**+a: CP$^+$ $\circ\!\!-$ a: S$^-$
and: (S>S)/S$^+$= **and**+b: S>S$^+$ $\circ\!\!-$ b: S$^-$= ((c, **and**+b): S$^+$ $\circ\!\!-$ c: S$^-$) $\circ\!\!-$ b: S$^-$

(72)

database	$\overline{\textbf{that}+a\text{: CP} \circ\!\!- a\text{: S}}^{1}$,	
	$\overline{\textbf{it+rains}\text{: S}}^{3}$,	
	$\overline{(c, \textbf{and}+b)\text{: S} \circ\!\!- c\text{: S} \otimes b\text{: S}}^{2}$,	
	$\overline{\textbf{it+shines}\text{: S}}^{4}$	
agenda		
1.	**that**+(**it**+**rains**, **and**+**it**+**shines**): CP	RES a=(**it**+**rains**, **and**+**it**+**shines**)
2.	(**it**+**rains**, **and**+**it**+**shines**): S	RES c=**it**+**rains**, b=**it**+**shines**
3.	**it**+**rains**: S $\otimes$ **it**+**shines**: S	RES
4.	**it**+**shines**: S	

8 Discontinuity

We consider residuation calculi for two kinds of discontinuity: *regular*, for discontinuous functors, and for infix binders as in quantifier raising, reflexivisation, pied piping and gapping; and *head-oriented*, such as head infixation and head extraction in Germanic verb clusters and verb fronting. In each case the essential strategy is to specify discontinuous adjunction as a *primitive* (as opposed to derived) operation in the multigroupoid prosodic algebra of multimodal Lambek calculi, with respect to which discontinuity operators are defined by residuation.

8.1 Regular Discontinuity

In the discontinuity calculus of Morrill (1993, 1994d) connectives $\{/, \backslash\}$, $\{<, >\}$ and $\{\uparrow, \downarrow\}$ are interpreted by residuation with respect to adjunctions +, $(.,.)$ and W respectively in a trigroupoid $\langle L^*, +, (.,.), W, \epsilon\rangle$ where + is associative and has (left and right) identity $\epsilon \in L^*$, and $(.,.)$ and W satisfy the "split-wrap" equation: $(s_1, s_3)Ws_2 = s_1{+}s_2{+}s_3$. We see that (ϵ, ϵ) is a left identity for W; ϵ is required in the interest of linguistic generalisation: to include peripherality as a special case of discontinuity. Also for linguistic reasons however, formulas are interpreted as subsets of $L = L^* \backslash \{\epsilon\}$, preventing ϵ (but not (ϵ, ϵ)) from inhabiting types. The prosodic label equations then are as follows:

(73) a. $s_1{+}(s_2{+}s_3) = (s_1{+}s_2){+}s_3$
b. $s{+}\epsilon = \epsilon{+}s = s$
c. $(s_1, s_3)Ws_2 = s_1{+}s_2{+}s_3$

A verb-particle construction is derived Fitch-style as in (74).

(74)
1. $(rang, up)$ – **phone**: $(N\backslash S)\uparrow N$
2. $John$ – **j**: N
3. $Mary$ – **m**: N
4. $((rang, up)W John)$ – (**phone j**): $N\backslash S$ 1, 2 E$\uparrow$
5. $rang+John+up$ – (**phone j**): $N\backslash S$ = 4
6. $Mary+rang+John+up$ – ((**phone j**) **m**): S 3, 5 E$\backslash$

Discussion of semantics would take us outside the direct concerns of the present article; the reader is referred to e.g. Moortgat (1988, 1991) and Morrill (1993, 1994d) for explication. The effect of quantifier-raising, whereby quantifiers are to take sentential scope, is achieved by assignment of a quantifier phrase such as 'everyone' to a "quantifying-in" type $(S\uparrow N)\downarrow S$. A simple instance of quantifier-raising is shown in (75).

(75)
1. $John$ – **j**: N
2. $likes$ – **like**: $(N\backslash S)/N$
3. $everything$ – $\lambda x \forall y(x\ y)$: $(S\uparrow N)\downarrow S$
4. | a – x: N H
5. | $likes+a$ – (**like** x): $N\backslash S$ 2, 4 E/
6. | $John+likes+a$ – ((**like** x) **j**): S 1, 5 E$\backslash$
7. | $John+likes+a+\epsilon$ – ((**like** x) **j**): S = 6
8. | $((John+likes, \epsilon)W a)$ – ((**like** x) **j**): S = 7
9. $(John+likes, \epsilon)$ – λx((**like** x) **j**): $S\uparrow N$ 4, 8 I$\uparrow$
10. $((John+likes, \epsilon)W\, everything)$ – 3, 9 E$\downarrow$
$(\lambda x\forall y(x\ y)\ \lambda x($(**like** x) **j**)): S
11. $John+likes+everything$ – $\forall y$((**like** y) **j**): S = 10

The techniques given here show how proof net theorem-proving in implicational residuation calculi, and hence parsing in certain categorial logics, can be compiled into a form suitable for SLD-resolution in linear logic. This indicates a general strategy for checking the correctness of a proof net by unification in which one term is always ground, but leaves open the problem of computing unifiers in any particular case. This will be illustrated with respect to discontinuity. The unidirectionality of unification using the clausal implementation makes it manageable through normalisation of and recursive decent through ground terms. For the regular discontinuity calculus the task concerns us with the equations (73). The unification procedure is to compute, for a given ground term α, all the distinct assignments σ of ground terms to variables in another term α' such that $\alpha = \alpha'[\sigma]$. We treat the process in two stages; (75b) and (75c) are considered normalisation rules (with redex on the left and contractum on the right); associativity as in (75a) could be naturally treated by representing equivalence classes of associative bracketings as lists, though we shall not choose to do so.

In the first phase, the ground term α is normalised by transforming redexes to their contractums. The second stage proceeds by recursion on the structure of α'. There are the cases that α' is a variable, a constant, or has principle operator one of the three prosodic adjunctions. Finally there are the cases that α' is, or can be instantiated to, a redex.

If α' is a variable v then simply put $v = \alpha$. If α' is a constant k then if α is k succeed, otherwise fail. If α' is of the form (α'_1, α'_2) then if α is of the form (α_1, α_2) unify α_1 and α'_1 and α_2 and α'_2. If α' is of the form $\alpha'_1 W \alpha'_2$ then if α is of the form $\alpha_1 W \alpha_2$ unify α_1 and α'_1 and α_2 and α'_2. If α' is of the form $\alpha'_1+\alpha'_2$ then find representatives α_1 and α_2 satisfying $\alpha = \alpha_1+\alpha_2$ (using associativity) and unify α_1 and α'_1, and α_2 and α'_2.

It remains to consider the cases where α' has the form of, or can be instantiated to the form of, a redex (the redexes in α having been already removed in the first phase). If α' can be instantiated to $\beta+\epsilon$ unify α and β. If α' can be instantiated to $\epsilon+\beta$ unify α and β. If α' can be instantiated to $(\alpha'_1, \alpha'_3)W\alpha'_2$ then find representatives α_1, α_2 and α_3 satisfying $\alpha = \alpha_1+\alpha_2+\alpha_3$ and unify α_1 with α'_1, α_2 with α'_2, and α_3 with α'_3.

As seen earlier a simple instance of discontinuity is obtained by assigning the compound particle verb prosodic form (**rang**, **up**) to a wrapping transitive verb type $(\mathrm{N}\backslash\mathrm{S})\uparrow\mathrm{N}$. In the following derivation **Mary+rang+John+up** is unified with b+((**rang**, **up**)Wa) by the unifier $\{b = \textbf{Mary}, a = \textbf{John}\}$.

(76) (**rang**, **up**): $(\mathrm{N}\backslash\mathrm{S})\uparrow\mathrm{N}^+ =$ ((**rang**, **up**)Wa): $\mathrm{N}\backslash\mathrm{S}^+ \multimap a\colon \mathrm{N}^- =$
(b+((**rang**, **up**)Wa): $\mathrm{S}^+ \multimap b\colon \mathrm{N}^-$) $\multimap a\colon \mathrm{N}^-$

(77)

database	$\overline{\textbf{Mary}\colon \mathrm{N}}^2$,	
	$\overline{b+((\textbf{rang}, \textbf{up})Wa)\colon \mathrm{S} \multimap b\colon \mathrm{N} \otimes a\colon \mathrm{N}}^1$,	
	$\overline{\textbf{John}\colon \mathrm{N}}^3$	
agenda		
1.	**Mary+rang+John+up**: S	RES b=**Mary**, a=**John**
2.	**Mary**: N $\otimes$ **John**: N	RES
3.	**John**: N	

In the next derivation, for 'John likes everything', **John+likes+everything** is unified with cW**everything** by $c =$ (**John+likes**, ϵ), and (**John+likes**, ϵ)W**k** is subsequently unified with b+**likes**+a by $\{b = \textbf{John}, a = \textbf{k}\}$.

(78) **everything**: $(\mathrm{S}\uparrow\mathrm{N})\downarrow\mathrm{S}^+ =$ (cW**everything**): $\mathrm{S}^+ \multimap c\colon \mathrm{S}\uparrow\mathrm{N}^- =$
(cW**everything**): $\mathrm{S}^+ \multimap$ ((cW**k**): $\mathrm{S}^- \multimap$ **k**: N^+)

(79)

database	$\overline{\textbf{John}\colon \mathrm{N}}^4$,	
	$\overline{b+\textbf{likes}+a\colon \mathrm{S} \multimap b\colon \mathrm{N} \otimes a\colon \mathrm{N}}^3$,	
	$\overline{(cW\textbf{everything})\colon \mathrm{S} \multimap ((cW\textbf{k})\colon \mathrm{S} \multimap \textbf{k}\colon \mathrm{N})}^1$,	
	$\overline{\textbf{k}\colon \mathrm{N}_2}^5$	
agenda		
1.	**John+likes+everything**: S	RES $c =$ (**John+likes**, ϵ)
2.	((**John+likes**, ϵ)W**k**): S $\multimap$ **k**: N	DT
3.	**John+likes+k**: S	RES b=**John**, a=**k**
4.	**John**: N $\otimes$ **k**: N	RES
5.	**k**: N	

8.2 Bracket Operators and Head-Oriented Discontinuity

To treat head-oriented discontinuity we shall require bracket operators, interpreted in a prosodic algebra such as $\langle L, +, [\,.\,]\rangle$ where $[\,.\,]$ is a unary operation (Morrill 1992); for nice symmetric proof theory we require that this is a 1-1 function (permutation) so that its inverse $[\,.\,]^{-1}$ is total (Morrill 1994d, 1994c).

(80) $[[s]^{-1}] = [[s]]^{-1} = s$

(81)
$$\begin{array}{lclcl} D([\,]A) & = & \{[s] \mid s \in D(A)\} & & \\ D([\,]^{-1}A) & = & \{s \mid [s] \in D(A)\} & = & \{[s]^{-1} \mid s \in D(A)\} \end{array}$$

Intuitively $[\,]A$ is the result of appointing or crystalising prosodic objects as domains or constituents, and $[\,]^{-1}A$ the result of annulling or dissolving appointment as a domain. Labelled Prawitz-style natural deduction for bracket operators is as follows, including $[\,.\,]$ and its inverse $[\,.\,]^{-1}$ in prosodic labels, for which there is a prosodic label equation $[[\alpha]^{-1}] = [[\alpha]]^{-1} = \alpha$.

(82)
$$\frac{\begin{array}{c}\vdots\\ \alpha\colon A\end{array}}{[\alpha]\colon [\,]A}\mathrm{I}[\,] \qquad \frac{\begin{array}{c}\vdots\\ \alpha\colon [\,]A\end{array}}{[\alpha]^{-1}\colon A}\mathrm{E}[\,]$$

(83)
$$\dfrac{\begin{array}{c}\vdots\\ \alpha\colon A\end{array}}{[\,\alpha\,]^{-1}\colon [\,]^{-1}A}\,\mathrm{I}[\,]^{-1} \qquad\qquad \dfrac{\begin{array}{c}\vdots\\ \alpha\colon [\,]^{-1}A\end{array}}{[\,\alpha\,]\colon A}\,\mathrm{E}[\,]^{-1}$$

Head-oriented discontinuity is obtained by combining a bracketing operation and a primitive head adjunction in a headed calculus (Moortgat and Morrill 1991): the prosodic algebra is $\langle L, +_l, +_r, +_h, [\,_v\cdot], \epsilon\rangle$ with implications $\{\backslash_l, /_l\}$, $\{\backslash_r, /_r\}$, and $\{\uparrow, \downarrow\}$ interpreted by residuation w.r.t. $+_l$, $+_r$ and $+_h$ respectively. There are the following interaction axioms:

(84) $(\alpha +_r \beta) +_h \gamma = \alpha +_r (\beta +_h \gamma)$
$(\alpha +_l \beta) +_h \gamma = (\alpha +_h \beta) +_l \gamma$

And $[\,v\,]$ and $[\,v\,]^{-1}$ are bracketing operators with respect to permutation $[\,_v\cdot]$; ϵ is a left and right identity for $+_l$ and $+_r$; the bottoming-out interaction for head adjunction is (for our Dutch example):

(85) $[\,_v\alpha\,] +_h \beta = [\,_v\beta +_l \alpha\,]$

The following is a simple example of Dutch head-infixation:

(86) $\mathbf{boeken}+_r[_v\mathbf{kan}+_l\mathbf{lezen}]$
books can read
"is able to read books"

It has the Prawitz-style derivation (87).

(87)
$$\dfrac{\dfrac{\dfrac{\text{boeken}}{\mathbf{boeken}\colon \mathrm{N}} \qquad \dfrac{\dfrac{\text{lezen}}{\mathbf{lezen}\colon [\,v\,]^{-1}(\mathrm{N}\backslash_r\mathrm{IVi})}}{[_v\mathbf{lezen}]\colon \mathrm{N}\backslash_r\mathrm{IVi}}\,\mathrm{E}[\,v\,]^{-1}}{\mathbf{boeken}+_r[_v\mathbf{lezen}]\colon \mathrm{IVi}}\,\mathrm{E}\backslash_r \qquad \dfrac{\text{kan}}{\mathbf{kan}\colon \mathrm{IVi}{\downarrow}\mathrm{VP}}}{\dfrac{(\mathbf{boeken}+_r[_v\mathbf{lezen}])+_h\mathbf{kan}\colon \mathrm{VP}}{\dfrac{\mathbf{boeken}+_r([_v\mathbf{lezen}]+_h\mathbf{kan})\colon \mathrm{VP}}{\mathbf{boeken}+_r[_v\mathbf{kan}+_l\mathbf{lezen}]\colon \mathrm{VP}}=}=}\,\mathrm{E}{\downarrow}$$

The existing methods are applicable to head-oriented discontinuity, for which we also require bracket unfolding:

(88)
$$\dfrac{[\,\alpha\,]^{-1}\colon A^+}{\alpha\colon [\,]A^+} \qquad\qquad \dfrac{[\,\alpha\,]\colon A^+}{\alpha\colon [\,]^{-1}A^+}$$

(89)
$$\dfrac{[\,\alpha\,]^{-1}\colon A^-}{\alpha\colon [\,]A^-} \qquad\qquad \dfrac{[\,\alpha\,]\colon A^-}{\alpha\colon [\,]^{-1}A^-}$$

Then the verb-infixing example is derived as follows by clausal resolution:

(90) $\mathbf{lezen}\colon [\,v\,]^{-1}(\mathrm{N}\backslash_r\mathrm{IVi})^+ = [\,_v\mathbf{lezen}]\colon \mathrm{N}\backslash_r\mathrm{IVi}^+ = a+_r[\,_v\mathbf{lezen}]\colon \mathrm{IVi}^+ \circ\!\!- a\colon \mathrm{N}^-$
$\mathbf{kan}\colon \mathrm{IVi}{\downarrow}\mathrm{VP}^+ = b+_h\mathbf{kan}\colon \mathrm{VP}^+ \circ\!\!- b\colon \mathrm{IVi}^-$

(91) database $\overline{\mathbf{boeken}\colon \mathrm{N}}^{3}$,
$\overline{a+_r[\,_v\mathbf{lezen}]\colon \mathrm{IVi} \circ\!\!- a\colon \mathrm{N}}^{2}$,
$\overline{b+_h\mathbf{kan}\colon \mathrm{VP} \circ\!\!- b\colon \mathrm{IVi}}^{1}$

agenda

1.	$\mathbf{boeken}+_r[\,_v\mathbf{kan}+_l\mathbf{lezen}]\colon \mathrm{VP}$	RES $b = \mathbf{boeken}+_r[\,_v\mathbf{lezen}]$
2.	$\mathbf{boeken}+_r[\,_v\mathbf{lezen}]\colon \mathrm{IVi}$	RES $a{=}\mathbf{boeken}$
3.	$\mathbf{boeken}\colon \mathrm{N}$	

The only non-trivial step is that resolving the initial unit goal, for which the unification is explicated by the following:

(92) $\textbf{boeken}+_r[_v\textbf{kan}+_l\textbf{lezen}] = \textbf{boeken}+_r([_v\textbf{lezen}]+_h\textbf{kan}) =$
$(\textbf{boeken}+_r[_v\textbf{lezen}])+_h\textbf{kan}$

9 Conclusion

We hope that what we have shown here goes some way towards providing general and powerful methods for categorial parsing-as-deduction. There are some technical issues: to prove formally properties like completeness and the correctness of the recursive descent/redex pattern matching unification algorithm used, especially in relation to classes of systems and not just specific ones. It is also appropriate to look at how unification would work out for interpretation in ordered groupoids, which allow more sensitive structural interaction, allowing inclusion rather than only equation relations between modes; there is good linguistic motivation for such interaction. More generally, we can look at compilation with respect to other kinds of models. For example Morrill (1994b) uses the binary relational interpretation of **L**, and thereby avoids altogether the need to compute unifiers under associativity; that paper also makes some progress with respect to products, not considered here.

Finally, we note that labelling and unfolding with respect to the semantic dimension, using typed lambda terms, renders the task of generation-as-deduction in the same light as parsing-as-deduction. That is, the problem of generation (computing prosodics given semantics), as opposed to parsing (computing semantics given prosodics), is rendered precisely as the task of one way unification in the semantic algebra of typed lambda terms. This appears to offer the prospect of systematic generation in Lambek categorial grammar by simply one way unification in typed lambda calculus.

Bibliography

Bibel, W.: 1981, 'On matrices with connections', *Journal of the Association for Computing Machinery* **28**, 633–645.

Bibel, W. and E. Eder: 1993, 'Methods and calculi for deduction', in D.M. Gabbay, C.J. Hogger and J.A. Robinson (eds.) *Handbook of Logic in Artificial Intelligence and Logic Programming Volume 1: Foundations*, Clarendon Press, Oxford, 67–102.

Buszkowski, K.: 1986, 'Competeness results for Lambek syntactic calculus', *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **32**, 13–28.

Došen, Kosta: 1992, 'A Brief Survey of Frames for the Lambek Calculus', *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **38**, 179–187.

Došen, Kosta and Peter Schroeder-Heister (eds.): 1993, *Substructural Logics*, Oxford University Press, Oxford.

Gabbay, D.: 1991, *Labelled Deductive Systems*, to appear, Oxford University Press, Oxford.

Girard, Jean-Yves: 1987, 'Linear Logic', *Theoretical Computer Science* **50**, 1–102.

Hendriks, Herman: 1993, *Studied Flexibility: Categories and Types in Syntax and Semantics*, Ph.D dissertation, Institute for Logic, Language and Computation, Universiteit van Amsterdam.

Hepple, Mark: 1990, 'Normal form theorem proving for the Lambek calculus', in H. Karlgren (ed.), *Proceedings of COLING 1990*, Stockholm.

König, E.: 1989, 'Parsing as natural deduction', in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vancouver.

Lambek, J.: 1958, 'The mathematics of sentence structure', *American Mathematical Monthly* **65**, 154–170, also in W. Buszkowski, W. Marciszewski, and J. van Benthem (eds.): 1988, *Categorial Grammar*, Linguistic & Literary Studies in Eastern Europe Volume 25, John Benjamins, Amsterdam, 153–172.

Lambek, J.: 1961, 'On the calculus of syntactic types', in R. Jakobson (ed.) *Structure of language and its mathematical aspects*, Proceedings of the Symposia in Applied Mathematics **XII**, American Mathematical Society, 166–178.

Lambek, J.: 1988, 'Categorial and Categorical Grammars', in Richard T. Oehrle, Emmon Bach, and Deidre Wheeler (eds.) *Categorial Grammars and Natural Language Structures*, Studies in Linguistics and Philosophy Volume 32, D. Reidel, Dordrecht, 297–317.

Moortgat, Michael: 1988, *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht.

Moortgat, Michael: 1990, 'Categorial Logics: a computational perspective', Proceedings *Computer Science in the Netherlands*.

Moortgat, Michael: 1991, 'Generalised Quantification and Discontinuous type constructors', to appear in Sijtsma and Van Horck (eds.) *Proceedings Tilburg Symposium on Discontinuous Constituency*, Walter de Gruyter, Berlin.

Moortgat, Michael: 1992, 'Labelled Deductive Systems for categorial theorem proving', OTS Working Paper OTS–WP–CL–92–003, Rijksuniversiteit Utrecht, also in *Proceedings of the Eighth Amsterdam Colloquium*, Institute for Language, Logic and Computation, Universiteit van Amsterdam.

Moortgat, Michael and Glyn Morrill: 1991, 'Heads and Phrases: Type Calculus for Dependency and Constituent Structure', to appear in *Journal of Language, Logic, and Information.*

Morrill, Glyn: 1990, 'Rules and Derivations: Binding Phenomena and Coordination in Categorial Logic', in Deliverable R1.2.D of DYANA Dynamic Interpretation of Natural Language, ESPRIT Basic Research Action BR3175.

Morrill, Glyn: 1992, 'Categorial Formalisation of Relativisation: Pied Piping, Islands, and Extraction Sites', Report de Recerca LSI–92–23–R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Morrill, Glyn: 1993, *Discontinuity and Pied-Piping in Categorial Grammar*, Report de Recerca LSI–93–18–R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, to appear in *Linguistics and Philosophy.*

Morrill, Glyn: 1994a, 'Clausal Proof Nets and Discontinuity', Report de Recerca LSI–94–21–R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Morrill, Glyn: 1994b, 'Higher-Order Linear Logic Programming of Categorial Deduction, Report de Recerca LSI–94–42–R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, to appear in *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Dublin 1995.

Morrill, Glyn: 1994c, 'Structural Facilitation and Structural Inhibition', Report de Recerca LSI–94–26–R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, also in Michele Abrusci, Claudia Casadio and Michael Moortgat (eds.) *Linear Logic and Lambek Calculus: Proceedings 1993 Rome Workshop*, Dyana Occasional Publications, DYANA-2, Dynamic Interpretation of Natural Language, ESPRIT Basic Research Project 6852, 183–210.

Morrill, Glyn: 1994d, *Type Logical Grammar: Categorial Logic of Signs*, Kluwer Academic Publishers, Dordrecht.

Oehrle, Dick: 1994, 'Term labelled categorial type systems', to appear in *Linguistics and Philosophy.*

Oehrle, Richard T. and Shi Zhang: 1989, 'Lambek Calculus and Preposing of Embedded Subjects', *Chicago Linguistic Society* **25**, Chicago.

Roorda, Dirk: 1991, *Resource Logics: proof-theoretical investigations*, Ph.D. dissertation, Universiteit van Amsterdam.

Wallen, L.A.: 1990, *Automated proof search in non-classical logics: efficient matrix proof methods for modal and intuitionistic logics*, MIT Press.