Logic Programming of the Displacement Calculus

Glyn Morrill

Universitat Politècnica de Catalunya
morrill@lsi.upc.edu
http: //www-lsi.upc.edu/˜morrill/

Abstract. The displacement calculus of Morrill, Valentín and Fadda (2011)[12] forms a foundation for type logical categorial grammar in which discontinuity is accommodated alongside continuity in a logic which is free of structural rules and which enjoys Cut-elimination, the subformula property, decidability, and the finite reading property. The calculus deploys a new kind of sequent calculus which we call hypersequent calculus in which types and configurations have not only external context but also internal context, in the case that they are discontinuous. In this paper we consider the logic programming of backward chaining hypersequent proof search for the displacement calculus. We show how focusing eliminates all spurious ambiguity in the fragment without antecedent tensors and we illustrate coding of the essential features of displacement. In this way we lay a basis for parsing/theorem proving for this calculus, which is being used and extended in a system CatLog currently under development.

Keywords. categorial grammar, discontinuity, focusing, parsing as deduction, sequent calculus.

1 Introduction

Morrill, Valentín and Fadda (2011)[12] exemplifies applications of the displacement calculus such as quantification, discontinuous idioms, cross-serial dependencies, gapping, parentheticals, and comparative subdeletion. In this paper we address logic programming of parsing/theorem proving for the calculus by backward chaining (hyper)sequent proof search. We see that all spurious ambiguity in the fragment without antecedent tensors is eliminated by *focusing* (Andreoli 1992[1]) — controlling the types to which successive rules of inference are allowed to apply. Logic programming of the focused proof search depends on predicates handling the discontinuity, i.e. internal contexts, of discontinuous types and configurations. In Section 2 we review the displacement calculus. In Section 3 we define the focusing strategy. In Section 4 we look at the logic programming. We conclude in Section 5. The appendix contains a cover grammar of the PTQ fragment and output generated by the Prolog implementation CatLog currently under development.

2 The Displacement Calculus

Let a *vocabulary* V be a set including a distinguished symbol 1 called the *separator*. We define the *sort* $\sigma(s)$ of a string s over the vocabulary as the number of separators it contains. We define the *sort domains* $L_i, i \in \mathcal{N}$, as follows:

$$L_i = \{s \in V^*|\ \sigma(s) = i\} \tag{1}$$

The concatenation of strings of sort i and j is a string of sort $i + j$, thus the operation $+$ of concatenation is of functionality $L_i, L_j \rightarrow L_{i+j}$. We consider also operations of first and last wrap. Where s is a string of sort at least one and t is a string, the *first wrap* $s \times_{>} t$ is the string that results from replacing the first (leftmost) separator in s by t; in the same way the *last wrap* $s \times_{<} t$ is the string that results from replacing the last (rightmost) separator in s by t. Clearly first and last wrap are of functionality $L_{i+1}, L_j \rightarrow L_{i+j}$.

In the displacement calculus we define connective families by residuation with respect to concatenation and first and last wrap. The types so-defined are sorted according to the functionality of these operations. Thus there are sorted types $\mathcal{F}_i, i \in \mathcal{N}$ as follows, where 0 is the empty string and $k \in \{>, <\}$:[1]

$$\begin{array}{rlrll}
\mathcal{F}_j & := \mathcal{F}_i\backslash\mathcal{F}_{i+j} & [A\backslash C] & = \{s_2|\ \forall s_1 \in [A], s_1{+}s_2 \in [C]\} & \text{under} \\
\mathcal{F}_i & := \mathcal{F}_{i+j}/\mathcal{F}_j & [C/B] & = \{s_1|\ \forall s_2 \in [B], s_1{+}s_2 \in [C]\} & \text{over} \\
\mathcal{F}_{i+j} & := \mathcal{F}_i{\cdot}\mathcal{F}_j & [A{\cdot}B] & = \{s_1{+}s_2|\ s_1 \in [A]\ \&\ s_2 \in [B]\} & \text{product} \\
\mathcal{F}_0 & := I & [I] & = \{0\} & \text{product unit} \\
\mathcal{F}_j & := \mathcal{F}_{i+1}{\downarrow_k}\mathcal{F}_{i+j} & [A{\downarrow_k}C] & = \{s_2|\ \forall s_1 \in [A], s_1{\times_k}s_2 \in [C]\} & \text{infix} \\
\mathcal{F}_{i+1} & := \mathcal{F}_{i+j}{\uparrow_k}\mathcal{F}_j & [C{\uparrow_k}B] & = \{s_1|\ \forall s_2 \in [B], s_1{\times_k}s_2 \in [C]\} & \text{circumfix} \\
\mathcal{F}_{i+j} & := \mathcal{F}_{i+1}{\odot_k}\mathcal{F}_j & [A{\odot_k}B] & = \{s_1{\times_k}s_2|\ s_1 \in [A]\ \&\ s_2 \in [B]\} & \text{wrap} \\
\mathcal{F}_1 & := J & [J] & = \{1\} & \text{wrap unit}
\end{array} \tag{2}$$

Where A is a type, sA denotes its sort.

Sequent calculus for displacement, which we call hypersequent calculus, involves some novelties. The set $\mathcal{O}$ of *configurations* is defined as follows, where Λ is the metalinguistic empty string (product unit) and $[]$ is the metalinguistic separator (wrap unit):

$$\mathcal{O} ::= \Lambda \mid [] \mid \mathcal{F}_0 \mid \mathcal{F}_{i+1}\{\underbrace{\mathcal{O} : \ldots : \mathcal{O}}_{i+1\ \mathcal{O}\text{'s}}\} \mid \mathcal{O}, \mathcal{O} \tag{3}$$

The fourth clause refers to a discontinuous type of sort $i + 1$ intercalated by $i + 1$ configurations corresponding to the positions of its separators. The sort of a configuration is the number of (metalinguistic) separators it contains. Where Δ is a configuration of sort at least one and Γ is a configuration, the metalinguistic first wrap $\Delta|_{>}\Gamma$ is the configuration that results from replacing the first (leftmost) metalinguistic separator in Δ by Γ; in the same way, the metalinguistic last wrap $\Delta|_{<}\Gamma$ is the configuration that results from replacing the last (rightmost) metalinguistic separator in Δ by Γ.

[1] We may abbreviate $\{\downarrow_{>}, \odot_{>}, \uparrow_{>}\}$ as $\{\downarrow, \odot, \uparrow\}$.

Where A is a type, its *vector* $\overrightarrow{A}$ is defined by:

$$\overrightarrow{A} = \begin{cases} A & \text{if } sA = 0 \\ A\{\underbrace{[\,] : \ldots : [\,]}_{sA\ [\,]\text{'s}}\} & \text{if } sA > 0 \end{cases} \tag{4}$$

Where Δ is a configuration of sort i and $\Gamma_1, \ldots, \Gamma_i$ are configurations, the *fold* $\Delta \otimes \langle \Gamma_1, \ldots, \Gamma_i \rangle$ is the configuration that results from replacing the metalinguistic separators in Δ by $\Gamma_1, \ldots, \Gamma_i$ respectively left-to-right. The standard distinguished occurrence notation $\Delta(\Gamma)$ signifies an occurrence of subconfiguration Γ with context Δ. This context is external. In displacement calculus a distinguished occurrence can be discontinuous, i.e. have internal contexts also. We notate a distinguished potentially discontinuous *hyperoccurrence* of a subconfiguation Γ of sort i with external and internal context Δ as $\Delta\langle\Gamma\rangle$ which stands for $\Delta_0(\Gamma \otimes \langle \Delta_1, \ldots, \Delta_i \rangle)$ where Δ_0 is the external context and $\Delta_1, \ldots, \Delta_n$ are the internal contexts.

A (hyper)sequent $\Gamma \Rightarrow A$ comprises an antecedent configuration Γ and a succedent type A of the same sort. The (hyper)sequent calculus for displacement is as shown in Figure 1, where $k \in \{>, <\}$.

3 Focusing

For the Lambek calculus (Lambek 1958[8]) the property of Cut-elimination, that every theorem has a Cut-free proof, establishes decidability because the Cut-free sequent search space is finite, but within this space there are still multiple equivalent proofs differing in inessential rule orderings, resulting in 'spurious' ambiguity and computational inefficiency. For example, a complex type identity axiom instance can alternate with a proof involving succesive left and right inferences:

$$N\backslash S \Rightarrow N\backslash S \quad \text{vs.} \quad \cfrac{\cfrac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\,\backslash L}{N\backslash S \Rightarrow N\backslash S}\,\backslash R \tag{5}$$

Left and right inferences can sometimes permute, for example:

$$\cfrac{N \Rightarrow N \qquad \cfrac{N, N\backslash S \Rightarrow S}{N\backslash S \Rightarrow N\backslash S}\,\backslash R}{(N\backslash S)/N, N \Rightarrow N\backslash S}\,/L \quad \text{vs.} \quad \cfrac{\cfrac{N \Rightarrow N \qquad N, N\backslash S \Rightarrow S}{N, (N\backslash S)/N, N \Rightarrow S}\,/L}{(N\backslash S)/N, N \Rightarrow N\backslash S}\,\backslash R \tag{6}$$

And two left inferences can sometimes permute, for example:

$$\cfrac{CN \Rightarrow CN \qquad \cfrac{N \Rightarrow N \qquad S \Rightarrow S}{N, N\backslash S \Rightarrow S}\,\backslash L}{N/CN, CN, N\backslash S \Rightarrow S}\,/L \quad \text{vs.} \quad \cfrac{\cfrac{CN \Rightarrow CN \qquad N \Rightarrow N}{N/CN, CN \Rightarrow N}\,/L \qquad S \Rightarrow S}{N/CN, CN, N\backslash S \Rightarrow S}\,\backslash L \tag{7}$$

$$\dfrac{}{\overrightarrow{A} \Rightarrow A}\,id \qquad \dfrac{\mathit{\Gamma} \Rightarrow A \qquad \mathit{\Delta}\langle \overrightarrow{A} \rangle \Rightarrow B}{\mathit{\Delta}\langle \mathit{\Gamma} \rangle \Rightarrow B}\,Cut$$

$$\dfrac{\mathit{\Gamma} \Rightarrow A \qquad \mathit{\Delta}\langle \overrightarrow{C} \rangle \Rightarrow D}{\mathit{\Delta}\langle \mathit{\Gamma}, \overrightarrow{A\backslash C} \rangle \Rightarrow D}\,\backslash L \qquad \dfrac{\overrightarrow{A}, \mathit{\Gamma} \Rightarrow C}{\mathit{\Gamma} \Rightarrow A\backslash C}\,\backslash R$$

$$\dfrac{\mathit{\Gamma} \Rightarrow B \qquad \mathit{\Delta}\langle \overrightarrow{C} \rangle \Rightarrow D}{\mathit{\Delta}\langle \overrightarrow{C/B}, \mathit{\Gamma} \rangle \Rightarrow D}\,/L \qquad \dfrac{\mathit{\Gamma}, \overrightarrow{B} \Rightarrow C}{\mathit{\Gamma} \Rightarrow C/B}\,/R$$

$$\dfrac{\mathit{\Delta}\langle \overrightarrow{A}, \overrightarrow{B} \rangle \Rightarrow D}{\mathit{\Delta}\langle \overrightarrow{A{\cdot}B} \rangle \Rightarrow D}\,{\cdot}L \qquad \dfrac{\mathit{\Gamma}_1 \Rightarrow A \qquad \mathit{\Gamma}_2 \Rightarrow B}{\mathit{\Gamma}_1, \mathit{\Gamma}_2 \Rightarrow A{\cdot}B}\,{\cdot}R$$

$$\dfrac{\mathit{\Delta}\langle \mathit{\Lambda} \rangle \Rightarrow A}{\mathit{\Delta}\langle \overrightarrow{I} \rangle \Rightarrow A}\,IL \qquad \dfrac{}{\mathit{\Lambda} \Rightarrow I}\,IR$$

$$\dfrac{\mathit{\Gamma} \Rightarrow A \qquad \mathit{\Delta}\langle \overrightarrow{C} \rangle \Rightarrow D}{\mathit{\Delta}\langle \mathit{\Gamma}|_k \overrightarrow{A{\downarrow_k}C} \rangle \Rightarrow D}\,{\downarrow_k}L \qquad \dfrac{\overrightarrow{A}|_k \mathit{\Gamma} \Rightarrow C}{\mathit{\Gamma} \Rightarrow A{\downarrow_k}C}\,{\downarrow_k}R$$

$$\dfrac{\mathit{\Gamma} \Rightarrow B \qquad \mathit{\Delta}\langle \overrightarrow{C} \rangle \Rightarrow D}{\mathit{\Delta}\langle \overrightarrow{C{\uparrow_k}B}|_k \mathit{\Gamma} \rangle \Rightarrow D}\,{\uparrow_k}L \qquad \dfrac{\mathit{\Gamma}|_k \overrightarrow{B} \Rightarrow C}{\mathit{\Gamma} \Rightarrow C{\uparrow_k}B}\,{\uparrow_k}R$$

$$\dfrac{\mathit{\Delta}\langle \overrightarrow{A}|_k \overrightarrow{B} \rangle \Rightarrow D}{\mathit{\Delta}\langle \overrightarrow{A{\odot_k}B} \rangle \Rightarrow D}\,{\odot_k}L \qquad \dfrac{\mathit{\Gamma}_1 \Rightarrow A \qquad \mathit{\Gamma}_2 \Rightarrow B}{\mathit{\Gamma}_1|_k \mathit{\Gamma}_2 \Rightarrow A{\odot_k}B}\,{\odot_k}R$$

$$\dfrac{\mathit{\Delta}\langle [\,] \rangle \Rightarrow A}{\mathit{\Delta}\langle \overrightarrow{J} \rangle \Rightarrow A}\,JL \qquad \dfrac{}{[\,] \Rightarrow J}\,JR$$

Fig. 1. (Hyper)sequent calculus for displacement

To deal with this König (1989)[7], Hepple (1990, 1990)[5][4] and Hendriks (1993)[3] developed normalization for Lambek calculus theorem-proving, in an approach equivalent to focusing in linear logic where inferences are organized into alternate phases for synchronous and asynchronous connectives. Consider the fragment of the Lambek calculus with only antecedent and succedent divisions and succedent product. Focusing normalization comprises the following strategy:

- The identity axiom id is restricted to atomic types.
- In trying to prove a sequent, right rules are applied first, until the succedent type is atomic.
- Once the succedent type is atomic, an antecedent type is chosen as the *focus*. Successive left division rules will be applied to this focused type, with its value subtype becoming the focus of the major premise in each application. (8)

This focusing technique eliminates all spurious ambiguity in the fragment (i.e. without antecedent products) and is complete (Hepple 1990[5], 1990[4] appendix for the divisions; Hendriks 1993[3]). The focusing strategy for this fragment of Lambek calculus can be represented by marking types with boxes as shown in Figure 2; a sequent $\Gamma \Rightarrow A$ is proved by proving $\Gamma \Rightarrow \boxed{A}$.[2]

$$\frac{}{\boxed{P} \Rightarrow P}\,id \qquad \frac{\Gamma(\boxed{\mathrm{A}}) \Rightarrow P}{\Gamma(A) \Rightarrow \boxed{\mathrm{P}}}\,F$$

$$\frac{\Gamma \Rightarrow \boxed{A} \qquad \Delta(\boxed{\mathrm{C}}) \Rightarrow D}{\Delta(\Gamma, \boxed{A\backslash C}) \Rightarrow D}\,\backslash L \qquad \frac{A, \Gamma \Rightarrow \boxed{C}}{\Gamma \Rightarrow \boxed{A\backslash C}}\,\backslash R$$

$$\frac{\Gamma \Rightarrow \boxed{B} \qquad \Delta(\boxed{C}) \Rightarrow D}{\Delta(\boxed{C/B}, \Gamma) \Rightarrow D}\,/L \qquad \frac{\Gamma, B \Rightarrow \boxed{C}}{\Gamma \Rightarrow \boxed{C/B}}\,/R$$

$$\frac{\Gamma \Rightarrow \boxed{A} \qquad \Delta \Rightarrow \boxed{B}}{\Gamma, \Delta \Rightarrow \boxed{A{\cdot}B}}\,{\cdot}R$$

Fig. 2. Focusing for Lambek calculus

The hypersequent calculus for displacement shares the shape of the Lambek calculus in respect of residuation. Distinguished occurrences become distinguished hyperoccurrences and antecedent active types become vectors, but

[2] Note that a necessary condition for a focusing choice to lead to a proof is that the eventual value of the focus be the atomic succedent, because the last major premise in the chain of left division inferences must be an identity axiom. Hence focusing choices (rule F) not satisfying this condition can be discounted.

otherwise the rules for the continuous connectives of displacement calculus have exactly the same form as those of the Lambek calculus. And the rules for the discontinuous connectives in displacement calculus also have this form with the metalinguistic comma representing concatenation on which the continuous connectives hinge replaced by the defined metalinguistic operations of wrap on which the discontinuous connectives hinge. In all cases the pattern of residuated triples and object language/metalanguage interaction is the same in the respects that effect derivational equivalence. The main contribution of the present paper is the observation that, consequentially, the same focusing strategy that serves for Lambek calculus also serves for displacement calculus. For the fragment without antecedent tensors, focusing for the displacement calculus can be represented as shown in Figure 3.

$$\frac{}{\boxed{\overrightarrow{P}} \Rightarrow P}\,id \qquad \frac{\Gamma\langle \boxed{\overrightarrow{A}} \rangle \Rightarrow P}{\Gamma\langle A\rangle \Rightarrow \boxed{\mathrm{P}}}\,F$$

$$\frac{\Gamma \Rightarrow \boxed{A} \qquad \Delta\langle \boxed{\overrightarrow{C}}\rangle \Rightarrow D}{\Delta\langle \Gamma, \boxed{\overrightarrow{A\backslash C}}\rangle \Rightarrow D}\,\backslash L \qquad \frac{\overrightarrow{A}, \Gamma \Rightarrow \boxed{C}}{\Gamma \Rightarrow \boxed{A\backslash C}}\,\backslash R$$

$$\frac{\Gamma \Rightarrow \boxed{B} \qquad \Delta\langle \boxed{\overrightarrow{C}}\rangle \Rightarrow D}{\Delta\langle \boxed{\overrightarrow{C/B}}, \Gamma\rangle \Rightarrow D}\,/L \qquad \frac{\Gamma, \overrightarrow{B} \Rightarrow \boxed{C}}{\Gamma \Rightarrow \boxed{C/B}}\,/R$$

$$\frac{\Gamma_1 \Rightarrow \boxed{A} \qquad \Gamma_2 \Rightarrow \boxed{B}}{\Gamma_1, \Gamma_2 \Rightarrow \boxed{A{\cdot}B}}\,\cdot R$$

$$\frac{\Gamma \Rightarrow \boxed{A} \qquad \Delta\langle \boxed{\overrightarrow{C}}\rangle \Rightarrow D}{\Delta\langle \Gamma|_k \boxed{\overrightarrow{A{\downarrow_k}C}}\rangle \Rightarrow D}\,{\downarrow_k}L \qquad \frac{\overrightarrow{A}|_k\Gamma \Rightarrow \boxed{C}}{\Gamma \Rightarrow \boxed{A{\downarrow_k}C}}\,{\downarrow_k}R$$

$$\frac{\Gamma \Rightarrow \boxed{B} \qquad \Delta\langle \boxed{\overrightarrow{C}}\rangle \Rightarrow D}{\Delta\langle \boxed{\overrightarrow{C{\uparrow_k}B}}|_k\Gamma\rangle \Rightarrow D}\,{\uparrow_k}L \qquad \frac{\Gamma|_k\overrightarrow{B} \Rightarrow \boxed{C}}{\Gamma \Rightarrow \boxed{C{\uparrow_k}B}}\,{\uparrow_k}R$$

$$\frac{\Gamma_1 \Rightarrow \boxed{A} \qquad \Gamma_2 \Rightarrow \boxed{B}}{\Gamma_1|_k\Gamma_2 \Rightarrow \boxed{A{\odot_k}B}}\,{\odot_k}R$$

Fig. 3. Focusing for displacement calculus

4 Logic Programming

Logic programming of Lambek calculus sequent proof search is straightforward (Moortgat 1988[9] appendix). Sequent rules are coded as Horn clauses and configurations as lists of types, as illustrated in the following for /L, where for brevity we use pattern matching for concatenation $\oplus$:

$$\frac{\Gamma \Rightarrow A \qquad \Delta(C) \Rightarrow D}{\Delta(C/B, \Gamma) \Rightarrow D}/L \tag{9}$$

$$\begin{aligned} & p(\Delta_1 \oplus [C/B|\Gamma] \oplus \Delta_2, D) \leftarrow \\ & \qquad p(\Gamma, B), \\ & \qquad p(\Delta_1 \oplus [C] \oplus \Delta_2, D). \end{aligned} \tag{10}$$

Since complex instances of the identity axiom can always be decomposed, it is appropriate to restrict the identity axiom to atomic types:

$$\frac{}{P \Rightarrow P}id \tag{11}$$

$$\begin{aligned} & p([P], P) \leftarrow \\ & \qquad atomic(P). \end{aligned} \tag{12}$$

The programming can be refined to implement the focusing technique by defining for a goal sequent a first phase in which right rules are applied, followed by choice of a focus and a second phase in which a left (division) rule may be applied to the focused type.[3] Such an application generates a subgoal at the first phase in the minor premise and a subgoal at the second phase in the major premise.[4]

Because the displacement calculus shares the pattern of the Lambek calculus, the general logic programming considerations carry over. However, in addition the matching of hyperoccurrences, with internal as well as external contexts, must be handled. Types in displacement configurations are either of sort zero A_0 or of sort at least one $A_{i+1}\{\Gamma_1 : \ldots : \Gamma_{i+1}\}$ and a type occurrence may be encoded as a term $l(A, Ls)$ where A is the type and Ls is a list of its internal

[3] As remarked in fn. 1, since in the Lambek calculus a focusing choice can only lead to a proof if the eventual value of the focused type is the atomic succedent of the goal sequent, focusing choices can be restricted to such cases without loss of completeness. This makes for considerable savings in the search space, because otherwise left division rules must test all factorizations of unprovable sequents. The same observation applies for the displacement calculus.

[4] A necessary property for a Lambek sequent to be provable is that it have the same number of positive (succedent) and negative (antecedent) type occurrences of each atomic type (the van Benthem count invariant; van Benthem 1991[15]). This is so because positive and negative occurrences of the same atom must be matched in identity axioms, and because the rules are multiplicative in the sense of linear logic. Thus, the proof search for a sequent may be pruned if the sequent does not satisfy this invariant. Again this makes for considerable savings in the search space because otherwise left division rules must test all factorizations of unprovable sequents. The same observation applies for the displacement calculus.

context configurations (empty if $sA = 0$). If the type is focused we represent this by encoding it $f(A, Ls)$. A displacement configuration is a list of such terms and separators 1. Matching hypersequents against the displacement rules depends on the fold operation in terms of which hyperoccurrences are defined. This is coded as the following predicate $\mathit{fold}(+\mathit{Sin}, ?\mathit{Sout}, -\varGamma, -\varDelta\mathit{sin}, ?\varDelta\mathit{sout}, +\varGamma_1)$ which means that configuration $\varGamma$ is of the sort given by the unary notation difference list $\mathit{Sin} - \mathit{Sout}$ and $\varGamma_1$ is the result of replacing in order the separators of $\varGamma$ by the configurations given by the difference list $\varDelta\mathit{sin} - \varDelta\mathit{sout}$:

$$\begin{array}{l}
\mathit{fold}(S, S, [], \varDelta s, \varDelta s, []).\\[1em]
\mathit{fold}([1|S], \mathit{Sout}, [1|\varGamma], [\varDelta|\varDelta s], \varDelta\mathit{sout}, \varDelta \oplus \varGamma_1) \leftarrow\\
\qquad\qquad \mathit{fold}(S, \mathit{Sout}, \varGamma, \varDelta s, \varDelta\mathit{sout}, \varGamma_1).\\[1em]
\mathit{fold}(\mathit{Sin}, \mathit{Sout}, [l(A, Ls)|\varGamma], \varDelta\mathit{sin}, \varDelta\mathit{sout}, [l(A, Ls1)|\varGamma_1]) \leftarrow\\
\qquad\qquad \mathit{foldlst}(\mathit{Sin}, S, Ls, \varDelta\mathit{sin}, \varDelta s, Ls1),\\
\qquad\qquad \mathit{fold}(S, \mathit{Sout}, \varGamma, \varDelta s, \varDelta\mathit{sout}, \varGamma_1).\\[1em]
\mathit{foldlst}(S, S, [], \varDelta s, \varDelta s, []).\\[1em]
\mathit{foldlst}(\mathit{Sin}, \mathit{Sout}, [L|Ls], \varDelta\mathit{sin}, \varDelta\mathit{sout}, [L1|Ls1]) \leftarrow\\
\qquad\qquad \mathit{fold}(\mathit{Sin}, S, L, \varDelta\mathit{sin}, \varDelta s, L1),\\
\qquad\qquad \mathit{foldlst}(S, \mathit{Sout}, Ls, \varDelta s, \varDelta\mathit{sout}, Ls1).
\end{array} \tag{13}$$

To represent the logic programming of displacement rules we give successive notational transformations starting with the hypersequent calculus metalanguage and ending with Prolog. For /L this is as shown in Figure 4. Here $\mathit{pleft}(+\varGamma, -\varGamma_1)$ means that a left rule replaces a subconfiguration[5] $\varGamma$ in the conclusion by $\varGamma_1$ in the major premise. The transformation for /R is shown in Figure 5. Here $\mathit{vector}(+B, -\mathit{Bvec})$ means that Bvec is the vector of the type B. For $\downarrow_> L$ the programming transformations are as shown in Figure 6. Here $\mathit{firstins}(-\varGamma, +A, +\varGamma_1)$ means that $\varGamma_1$ is the result of replacing the first (leftmost) metalinguistic separator in $\varGamma$ by type A. For $\downarrow_>$R the programming transformation is as shown in Figure 7. The other rules are implemented similarly on the basis of the same ideas.

5 Conclusion

The principles presented in this paper form the basis of a Prolog categorial parser/theorem prover CatLog under development.[6] CatLog deals with many

[5] an ordinary occurrence, not a hyperoccurrence

[6] An earlier such program Catlog for generalized Lambek calculus based on sequent calculus normalization was developed in the first half of the 90s, but was not brought to term because of the absence of a satisfactory treatment of discontinuity. The introduction of the displacement calculus has allowed this line of investigation to resume.

$$\frac{\Gamma \Rightarrow B \qquad \Delta\langle\overrightarrow{C}\rangle \Rightarrow D}{\Delta\langle\overrightarrow{C/B}, \Gamma\rangle \Rightarrow D}\,/L$$

$$\frac{\Gamma \Rightarrow B \qquad \Delta_0(\overrightarrow{C} \otimes \langle\Delta_1, \ldots, \Delta_{sC}\rangle) \Rightarrow D}{\Delta_0((\overrightarrow{C/B}, \Gamma) \otimes \langle\Delta_1, \ldots, \Delta_{sC}\rangle) \Rightarrow D}\,/L$$

$$\frac{\Gamma \Rightarrow B \qquad \Delta_0(\overrightarrow{C} \otimes \langle\Delta_1, \ldots, \Delta_{sC}\rangle) \Rightarrow D}{\Delta_0(\overrightarrow{C/B} \otimes \langle\Delta_1 : \ldots : \Delta_{sC-sB}\rangle, \Gamma \otimes \langle\Delta_{1+sC-sB}, \ldots, \Delta_{sC}\rangle) \Rightarrow D}\,/L$$

$$\frac{\Gamma \Rightarrow B \qquad \Delta_0(C\{\Delta_1 : \ldots : \Delta_{sC}\}) \Rightarrow D}{\Delta_0(C/B\{\Delta_1 : \ldots : \Delta_{sC-sB}\}, \Gamma \otimes \langle\Delta_{1+sC-sB}, \ldots, \Delta_{sC}\rangle) \Rightarrow D}\,/L$$

```
$pleft([f(C/B, Ls1)] \oplus \Gamma', [f(C, Ls1 \oplus Ls2)]) \leftarrow$
    $ssort(B, SB),$
    $fold(SB, [], \Gamma, Ls2, [], \Gamma'),$
    $p(\Gamma, B).$
```

```
pleft([f(C/B, Ls1)|Gamma1], [f(C, Ls)]) :-
      ssort(B, SB),
      fold(SB, [], Gamma, Ls2, [], Gamma1),
      p(Gamma, B),
      append(Ls1, Ls2, Ls).
```

Fig. 4. Programming of /L

$$\frac{\Gamma, \overrightarrow{B} \Rightarrow C}{\Gamma \Rightarrow C/B}\,/R$$

$$\frac{\Gamma, B\{\underbrace{[\,] : \ldots : [\,]}_{sB\ [\,]\text{'s}}\} \Rightarrow C}{\Gamma \Rightarrow C/B}\,/R$$

```
$p(\Gamma, C/B) \leftarrow$
    $vector(B, Bvec),$
    $p(\Gamma \oplus [Bvec], C).$
```

```
p(Gamma, C/B) :-
      vector(B, Bvec),
      append(Gamma, [Bvec], Gamma1),
      p(Gamma1, C).
```

Fig. 5. Programming of /R

$$\frac{\Gamma \Rightarrow A \qquad \Delta\langle\overrightarrow{C}\rangle \Rightarrow D}{\Delta\langle\Gamma|_{>}\overrightarrow{A{\downarrow_{>}}C}\rangle \Rightarrow D}\,{\downarrow_{>}}L$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta_0(\overrightarrow{C} \otimes \langle\Delta_1, \ldots, \Delta_{sC}\rangle) \Rightarrow D}{\Delta_0((\Gamma|_{>}\overrightarrow{A{\downarrow_{>}}C}) \otimes \langle\Delta_1, \ldots, \Delta_{sC}\rangle) \Rightarrow D}\,{\downarrow_{>}}L$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta_0(\overrightarrow{C} \otimes \langle\Delta_1, \ldots, \Delta_{sC}\rangle) \Rightarrow D}{\Delta_0(\Gamma \otimes \langle\overrightarrow{A{\downarrow_{>}}C} \otimes \langle\Delta_1, \ldots, \Delta_{1+sC-sA}\rangle, \Delta_{2+sC-sA}, \ldots, \Delta_{sC}\rangle) \Rightarrow D}\,{\downarrow_{>}}L$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta_0(C\{\Delta_1, \ldots, \Delta_{sC}\}) \Rightarrow D}{\Delta_0(\Gamma \otimes \langle A{\downarrow_{>}}C\{\Delta_1, \ldots, \Delta_{1+sC-sA}\}, \Delta_{2+sC-sA}, \ldots, \Delta_{sC}\rangle) \Rightarrow D}\,{\downarrow_{>}}L$$

```
$pleft(\Gamma, [f(C, Ls1 \oplus Ls2)]) \leftarrow$
        $firstins(\Gamma_1, f(A{\downarrow_{>}}C, Ls1), \Gamma),$
        $ssort(A, SA),$
        $fold(SA, [], \Gamma_2, [[1]|Ls2], [], \Gamma_1),$
        $p(\Gamma_2, A).$
```

```
pleft(Gamma, [f(C, Ls1Ls2)]) :-
      firstins(Gamma1, f(A 'v<' C, Ls1), Gamma),
      ssort(A, SA),
      fold(SA, [], Gamma2, [[1]|Ls2], Gamma1),
      p(Gamma2, A),
      append(Ls1, Ls2, Ls1Ls2).
```

Fig. 6. Programming of $\downarrow_{>}$L

$$\frac{\overrightarrow{A}|_{>}\Gamma \Rightarrow C}{\Gamma \Rightarrow A{\downarrow_{>}}C}\,{\downarrow_{>}}R$$

$$\frac{A\{\underbrace{[\,]:[\,]:\ldots:[\,]}_{sA\ [\,]\text{'s}}\}|_{>}\Gamma \Rightarrow C}{\Gamma \Rightarrow A{\downarrow_{>}}C}\,{\downarrow_{>}}R$$

$$\frac{A\{\Gamma:\underbrace{[\,]:\ldots:[\,]}_{sA-1\ [\,]\text{'s}}\} \Rightarrow C}{\Gamma \Rightarrow A{\downarrow_{>}}C}\,{\downarrow_{>}}R$$

$$\begin{array}{l} p(\Gamma, A{\downarrow_{>}}C) \leftarrow \\ \qquad \mathit{vector}(A, l(A, [[1]|Ls])), \\ \qquad p([l(A, [\Gamma|Ls])], C). \end{array}$$

```
p(Gamma, A 'v<' C) :-
      vector(A, l(A, [[1]|Ls])),
      p([l(A, [Gamma|Ls])], C).
```

Fig. 7. Programming of $\downarrow_k$R

connectives over and above the displacement connectives, but it is the displacement hypersequent calculus, which forms a new multiplicative basis for categorial logic, which forms its core. The inclusion in CatLog of standard Curry-Howard type logical categorial semantics is straightforward because the semantic reading of a derivation is a homomorphic image of the syntactic proof. This paper has addressed the main technical question for displacement calculus, and hence displacement categorial logic, of backward chaining sequent proof search in a logic programming parsing as deduction paradigm.

Appendix

This appendix contains unedited LaTeX output of CatLog for a cover grammar of PTQ. A printout of the categorial lexicon is followed by the analyses computed for example sentences drawn from Dowty, Wall and Peters (1981, ch. 7)[2]. The categorial logic for the fragment extends displacement calculus with implicitly universally quantified first order term structure for features on atomic types (Morrill 1994, ch. 6[13]), the limited contraction for anaphora of Jäger 2005[6], a difference operator (cf. Morrill and Valentín 2010[11]) and modalities for intensionality (Morrill 1990[10], 1994[13] chs. 4 & 5, 2010[14] ch. 8) such that $\Box A$ is the type of expressions in intension of type A; $\blacksquare A$ represents such expressions in intension which are rigid designators.

a : $\blacksquare(((SA{\uparrow}\blacksquare NB){\downarrow}SA)/CNB) : \lambda C\lambda D\exists E[(C\ E) \wedge (D\ E)]$

and : $\blacksquare((SA\backslash SA)/SA) : \lambda B\lambda C[C \wedge B]$

and : $\blacksquare(((NA\backslash SB)\backslash(NA\backslash SB))/(NA\backslash SB)) : \lambda C\lambda D\lambda E[(D\ E) \wedge (C\ E)]$

believes : $\square((NA\backslash Sf)/CPthat) : \mathit{believe}$

bill : $\blacksquare Nm : b$

catch : $\square((NA\backslash Sb)/NB) : \mathit{catch}$

doesnt : $\square((NA\backslash s)/(NA\backslash s)) : {}^{\wedge}\lambda B\lambda C\neg(B\ C)$

eat : $\square((NA\backslash Sb)/NB) : \mathit{eat}$

every : $\blacksquare(((SA{\uparrow}NB){\downarrow}SA)/CNB) : \lambda C\lambda D\forall E[(C\ E) \rightarrow (D\ E)]$

finds : $\square((NA\backslash Sf)/NB) : \mathit{finds}$

fish : $\square CNn : \mathit{fish}$

he : $\blacksquare((\blacksquare SA|Nm)/(Nm\backslash SA)) : \lambda BB$

her : $\square(\square((s{\uparrow}Nf) - (J{\bullet}(Nf\backslash s))){\downarrow}(\square s|Nf)) : \lambda A\lambda B((A\ B)\ B)$

her : $\blacksquare(((((SA{\uparrow}Nf) - (J{\bullet}(Nf\backslash SA))){\uparrow}\blacksquare Nf) - (J{\bullet}((Nf\backslash s){\uparrow}Nf))){\downarrow_{<}}(SA{\uparrow}\blacksquare Nf)) : \lambda B\lambda C((B\ C)\ C)$

in : $\blacksquare(((NA\backslash SB)\backslash(NA\backslash SB))/NC) : \lambda D\lambda E\lambda F(({}^{\vee}\mathit{in}\ D)\ (E\ F))$

is : $\blacksquare((NA\backslash Sf)/NB) : \lambda C\lambda D[D = C]$

it : $\blacksquare((SA{\uparrow}Nn){\downarrow}(\blacksquare SA|Nn)) : \lambda BB$

it : $\blacksquare(((((SA{\uparrow}Nn) - (J{\bullet}(Nn\backslash SA))){\uparrow}\blacksquare Nn) - (J{\bullet}((Nn\backslash SB){\uparrow}Nn))){\downarrow_{<}}(SA{\uparrow}\blacksquare Nn)) : \lambda C\lambda D((C\ D)\ D)$

john : $\blacksquare Nm : j$

loses : $\square((NA\backslash Sf)/NB) : \mathit{loses}$

loves : $\square((NA\backslash Sf)/NB) : \mathit{loves}$

man : $\square CNm : \mathit{man}$

necessarily : $\blacksquare(SA/\square SA) : \mathit{nec}$

or : $\blacksquare((SA\backslash SA)/SA) : \lambda B\lambda C[C \vee B]$

or : $\blacksquare(((NA\backslash Sf)\backslash(NA\backslash Sf))/(NA\backslash Sf)) : \lambda B\lambda C\lambda D[(C\ D) \vee (B\ D)]$

park : $\square CNn : \mathit{park}$

seeks : $\blacksquare((NA\backslash Sf)/\square(((NB\backslash SC)/ND)\backslash(NB\backslash SC))) : \lambda E\lambda F(({}^{\vee}\mathit{tries}\ {}^{\wedge}(({}^{\vee}E\ {}^{\vee}\mathit{find})\ F))\ F)$

she : $\blacksquare((\blacksquare SA|Nf)/(Nf\backslash SA)) : \lambda BB$

slowly : $\square(\square(NA\backslash SB)\backslash(NA\backslash SB)) : \mathit{slowly}$

such+that : $\blacksquare((CNA\backslash CNA)/(Sf|NA)) : \lambda B\lambda C\lambda D[(C\ D) \wedge (B\ D)]$

talks : $\square(NA\backslash Sf) : \mathit{talk}$

that : $\blacksquare(CPthat/\square Sf) : \lambda AA$

the : $\blacksquare(NA/CNA) : \iota$

to : $\blacksquare((NA\backslash St)/(NA\backslash Sb)) : \lambda BB$

tries : $\blacksquare((NA\backslash Sf)/\square(NA\backslash St)) : \lambda B\lambda C(({}^{\vee}\mathit{tries}\ {}^{\wedge}({}^{\vee}B\ C))\ C)$

unicorn : $\square CNn : \mathit{unicorn}$

walk : $\square(NA\backslash Sb) : \mathit{walk}$

walks : $\square(NA\backslash Sf) : \mathit{walk}$

woman : $\square CNf : \mathit{woman}$

Example (7-16) involves a quantifier phrase in subject position. Here and always lexical types are modalized outermost because lexical meanings are intensions. Within its modality the type for the quantifier is a functor seeking a count noun to its right. (A feature variable transmits the gender value.) This yields a generalized quantifier type which will infix at the position of a nominal in a sentence, simulating term insertion S14. Example (7-32) involves subject quantification and verb phrase coordination. Example (7-60, 7-62) involves the lifted intensional object transitive verb 'seek' which is synonymous with 'try to find'. The sentence has a specific reading in which there is a unicorn which John is trying to find, and a non-specific reading in which John is just trying to bring it about that he finds some, any, unicorn. The latter reading does not have existential commitment: it can be true without any unicorn actually existing. The two readings are obtained from derivations in which the indefinite object term-inserts outside or within the scope of the intensional verb. Example (7-76) is the classic example of the interaction of the copula and indefinites in Montague grammar to assign *John is a man* a logical form equivalent to $({}^{\vee}\mathit{man}\ j)$. Our grammar conserves this, with a lowered object type for the copula. This same lower extensional transitive verb type for *find* means that (7-98), by contrast with (7-60, 7-62), has only one reading, with existential commitment.

(7-16) **every+man+talks** : Sf

$\blacksquare(((SA{\uparrow}NB){\downarrow}SA)/CNB) : \lambda C\lambda D\forall E[(C\ E) \rightarrow (D\ E)], \Box CNm : \mathit{man}, \Box(NF\backslash Sf) : \mathit{talk} \ \Rightarrow\ Sf$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{CNm \Rightarrow CNm}}{\boxed{\Box CNm} \Rightarrow CNm}\,\Box L \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nm \Rightarrow Nm} \quad \dfrac{}{Sf \Rightarrow Sf}}{Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf}\,\backslash L}{Nm, \boxed{\Box(Nm\backslash Sf)} \Rightarrow Sf}\,\Box L}{1, \Box(Nm\backslash Sf) \Rightarrow Sf{\uparrow}Nm}\,{\uparrow}R \qquad \dfrac{}{Sf \Rightarrow Sf}}{\boxed{(Sf{\uparrow}Nm){\downarrow}Sf}, \Box(Nm\backslash Sf) \Rightarrow Sf}\,{\downarrow}L}{\boxed{((Sf{\uparrow}Nm){\downarrow}Sf)/CNm}, \Box CNm, \Box(Nm\backslash Sf) \Rightarrow Sf}\,/L}{\boxed{\blacksquare(((Sf{\uparrow}Nm){\downarrow}Sf)/CNm)}, \Box CNm, \Box(Nm\backslash Sf) \Rightarrow Sf}\,\blacksquare L$$

$\forall C[(\breve{}\mathit{man}\ C) \rightarrow (\breve{}\mathit{talk}\ C)]$

(7-32) **every+man+walks+or+talks** : Sf

$\blacksquare(((SA{\uparrow}NB){\downarrow}SA)/CNB) : \lambda C\lambda D\forall E[(C\ E) \rightarrow (D\ E)], \Box CNm : \mathit{man}, \Box(NF\backslash Sf) : \mathit{walk}, \blacksquare((SG\backslash SG)/SG) : \lambda H\lambda I[I \vee H], \Box(NJ\backslash Sf) : \mathit{talk} \ \Rightarrow\ Sf$

$\blacksquare(((SA{\uparrow}NB){\downarrow}SA)/CNB) : \lambda C\lambda D\forall E[(C\ E) \rightarrow (D\ E)], \Box CNm : \mathit{man}, \Box(NF\backslash Sf) : \mathit{walk}, \blacksquare(((NG\backslash Sf)\backslash(NG\backslash Sf))/(NG\backslash Sf)) : \lambda H\lambda I\lambda J[(I\ J) \vee (H\ J)], \Box(NK\backslash Sf) : \mathit{talk} \ \Rightarrow\ Sf$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{CNm \Rightarrow CNm}}{\boxed{\Box CNm} \Rightarrow CNm}\,\Box L \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nm \Rightarrow Nm} \quad \dfrac{}{Sf \Rightarrow Sf}}{Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf}\,\backslash L}{Nm, \boxed{\Box(Nm\backslash Sf)} \Rightarrow Sf}\,\Box L}{\Box(Nm\backslash Sf) \Rightarrow Nm\backslash Sf}\,\backslash R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nm \Rightarrow Nm} \quad \dfrac{}{Sf \Rightarrow Sf}}{Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf}\,\backslash L}{Nm, \boxed{\Box(Nm\backslash Sf)} \Rightarrow Sf}\,\Box L}{\Box(Nm\backslash Sf) \Rightarrow Nm\backslash Sf}\,\backslash R \qquad \dfrac{\dfrac{}{Nm \Rightarrow Nm} \quad \dfrac{}{Sf \Rightarrow Sf}}{Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf}\,\backslash L}{Nm, \Box(Nm\backslash Sf), \boxed{(Nm\backslash Sf)\backslash(Nm\backslash Sf)} \Rightarrow Sf}\,\backslash L}{Nm, \Box(Nm\backslash Sf), \boxed{((Nm\backslash Sf)\backslash(Nm\backslash Sf))/(Nm\backslash Sf)}, \Box(Nm\backslash Sf) \Rightarrow Sf}\,/L}{Nm, \Box(Nm\backslash Sf), \boxed{\blacksquare(((Nm\backslash Sf)\backslash(Nm\backslash Sf))/(Nm\backslash Sf))}, \Box(Nm\backslash Sf) \Rightarrow Sf}\,\blacksquare L}{1, \Box(Nm\backslash Sf), \blacksquare(((Nm\backslash Sf)\backslash(Nm\backslash Sf))/(Nm\backslash Sf)), \Box(Nm\backslash Sf) \Rightarrow Sf{\uparrow}Nm}\,{\uparrow}R \qquad \dfrac{}{Sf \Rightarrow Sf}}{\boxed{(Sf{\uparrow}Nm){\downarrow}Sf}, \Box(Nm\backslash Sf), \blacksquare(((Nm\backslash Sf)\backslash(Nm\backslash Sf))/(Nm\backslash Sf)), \Box(Nm\backslash Sf) \Rightarrow Sf}\,{\downarrow}L}{\boxed{((Sf{\uparrow}Nm){\downarrow}Sf)/CNm}, \Box CNm, \Box(Nm\backslash Sf), \blacksquare(((Nm\backslash Sf)\backslash(Nm\backslash Sf))/(Nm\backslash Sf)), \Box(Nm\backslash Sf) \Rightarrow Sf}\,/L}{\boxed{\blacksquare(((Sf{\uparrow}Nm){\downarrow}Sf)/CNm)}, \Box CNm, \Box(Nm\backslash Sf), \blacksquare(((Nm\backslash Sf)\backslash(Nm\backslash Sf))/(Nm\backslash Sf)), \Box(Nm\backslash Sf) \Rightarrow Sf}\,\blacksquare L$$

$\forall C[(\breve{}\mathit{man}\ C) \rightarrow [(\breve{}\mathit{walk}\ C) \vee (\breve{}\mathit{talk}\ C)]]$

(7-60, 7-62) **john+seeks+a+unicorn** : Sf

$\blacksquare Nm : j, \blacksquare((NA\backslash Sf)/\Box(((NB\backslash SC)/ND)\backslash(NB\backslash SC))) : \lambda E\lambda F((\breve{}\mathit{tries}\ \hat{}((\breve{}E\ \breve{}\mathit{find})\ F))\ F), \blacksquare(((SG{\uparrow}\blacksquare NH){\downarrow}SG)/CNH) : \lambda I\lambda J\exists K[(I\ K) \wedge (J\ K)], \Box CNn : \mathit{unicorn} \ \Rightarrow\ Sf$

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$CNn \Rightarrow CNn$}
\RightLabel{$\square L$}
\UnaryInfC{$\boxed{\square CNn} \Rightarrow CNn$}
\AxiomC{}
\UnaryInfC{$Nn \Rightarrow Nn$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\boxed{\blacksquare Nn} \Rightarrow Nn$}
\AxiomC{}
\UnaryInfC{$NA \Rightarrow NA$}
\AxiomC{}
\UnaryInfC{$SA \Rightarrow SA$}
\RightLabel{$\backslash L$}
\BinaryInfC{$NA, \boxed{NA\backslash SB} \Rightarrow SB$}
\RightLabel{$/L$}
\BinaryInfC{$NA, \boxed{(NA\backslash SB)/Nn}, \blacksquare Nn \Rightarrow SB$}
\RightLabel{$\backslash R$}
\UnaryInfC{$(NA\backslash SB)/Nn, \blacksquare Nn \Rightarrow NA\backslash SB$}
\RightLabel{$\backslash R$}
\UnaryInfC{$\blacksquare Nn \Rightarrow ((NA\backslash SB)/Nn)\backslash(NA\backslash SB)$}
\RightLabel{$\square R$}
\UnaryInfC{$\blacksquare Nn \Rightarrow \square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))$}
\AxiomC{}
\UnaryInfC{$Nm \Rightarrow Nm$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\boxed{\blacksquare Nm} \Rightarrow Nm$}
\AxiomC{}
\UnaryInfC{$Sf \Rightarrow Sf$}
\RightLabel{$\backslash L$}
\BinaryInfC{$\blacksquare Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf$}
\RightLabel{$/L$}
\BinaryInfC{$\blacksquare Nm, \boxed{(Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))}, \blacksquare Nn \Rightarrow Sf$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\blacksquare Nm, \boxed{\blacksquare((Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB)))}, \blacksquare Nn \Rightarrow Sf$}
\RightLabel{$\uparrow R$}
\UnaryInfC{$\blacksquare Nm, \blacksquare((Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))), 1 \Rightarrow Sf\uparrow\blacksquare Nn$}
\AxiomC{}
\UnaryInfC{$Sf \Rightarrow Sf$}
\RightLabel{$\downarrow L$}
\BinaryInfC{$\blacksquare Nm, \blacksquare((Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))), \boxed{(Sf\uparrow\blacksquare Nn)\downarrow Sf} \Rightarrow Sf$}
\RightLabel{$/L$}
\BinaryInfC{$\blacksquare Nm, \blacksquare((Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))), \boxed{((Sf\uparrow\blacksquare Nn)\downarrow Sf)/CNn}, \square CNn \Rightarrow Sf$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\blacksquare Nm, \blacksquare((Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))), \boxed{\blacksquare(((Sf\uparrow\blacksquare Nn)\downarrow Sf)/CNn)}, \square CNn \Rightarrow Sf$}
\end{prooftree}
$$

$\exists C[(\check{}unicorn\ C) \wedge ((\check{}tries\ \hat{}((\check{}find\ C)\ j))\ j)]$

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$Nm \Rightarrow Nm$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\boxed{\blacksquare Nm} \Rightarrow Nm$}
\AxiomC{}
\UnaryInfC{$Sf \Rightarrow Sf$}
\RightLabel{$\backslash L$}
\BinaryInfC{$\blacksquare Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf$}
\AxiomC{}
\UnaryInfC{$CNn \Rightarrow CNn$}
\RightLabel{$\square L$}
\UnaryInfC{$\boxed{\square CNn} \Rightarrow CNn$}
\AxiomC{}
\UnaryInfC{$Nn \Rightarrow Nn$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\boxed{\blacksquare Nn} \Rightarrow Nn$}
\AxiomC{}
\UnaryInfC{$NA \Rightarrow NA$}
\AxiomC{}
\UnaryInfC{$SA \Rightarrow SA$}
\RightLabel{$\backslash L$}
\BinaryInfC{$NA, \boxed{NA\backslash SB} \Rightarrow SB$}
\RightLabel{$/L$}
\BinaryInfC{$NA, \boxed{(NA\backslash SB)/Nn}, \blacksquare Nn \Rightarrow SB$}
\RightLabel{$\uparrow R$}
\UnaryInfC{$NA, (NA\backslash SB)/Nn, 1 \Rightarrow SB\uparrow\blacksquare Nn$}
\AxiomC{}
\UnaryInfC{$SA \Rightarrow SA$}
\RightLabel{$\downarrow L$}
\BinaryInfC{$NA, (NA\backslash SB)/Nn, \boxed{(SB\uparrow\blacksquare Nn)\downarrow SB} \Rightarrow SB$}
\RightLabel{$/L$}
\BinaryInfC{$NA, (NA\backslash SB)/Nn, \boxed{((SB\uparrow\blacksquare Nn)\downarrow SB)/CNn}, \square CNn \Rightarrow SB$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$NA, (NA\backslash SB)/Nn, \boxed{\blacksquare(((SB\uparrow\blacksquare Nn)\downarrow SB)/CNn)}, \square CNn \Rightarrow SB$}
\RightLabel{$\backslash R$}
\UnaryInfC{$(NA\backslash SB)/Nn, \blacksquare(((SB\uparrow\blacksquare Nn)\downarrow SB)/CNn), \square CNn \Rightarrow NA\backslash SB$}
\RightLabel{$\backslash R$}
\UnaryInfC{$\blacksquare(((SA\uparrow\blacksquare Nn)\downarrow SA)/CNn), \square CNn \Rightarrow ((NB\backslash SA)/Nn)\backslash(NB\backslash SA)$}
\RightLabel{$\square R$}
\UnaryInfC{$\blacksquare(((SA\uparrow\blacksquare Nn)\downarrow SA)/CNn), \square CNn \Rightarrow \square(((NB\backslash SA)/Nn)\backslash(NB\backslash SA))$}
\RightLabel{$/L$}
\BinaryInfC{$\blacksquare Nm, \boxed{(Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB))}, \blacksquare(((SB\uparrow\blacksquare Nn)\downarrow SB)/CNn), \square CNn \Rightarrow Sf$}
\RightLabel{$\blacksquare L$}
\UnaryInfC{$\blacksquare Nm, \boxed{\blacksquare((Nm\backslash Sf)/\square(((NA\backslash SB)/Nn)\backslash(NA\backslash SB)))}, \blacksquare(((SB\uparrow\blacksquare Nn)\downarrow SB)/CNn), \square CNn \Rightarrow Sf$}
\end{prooftree}
$$

$((\check{}tries\ \hat{}\exists G[(\check{}unicorn\ G) \wedge ((\check{}find\ G)\ j)])\ j)$

(7-76) **john+is+a+man** : Sf

$\blacksquare Nm : j, \blacksquare((NA\backslash Sf)/NB) : \lambda C\lambda D[D = C], \blacksquare(((SE\uparrow\blacksquare NF)\downarrow SE)/CNF) : \lambda G\lambda H\exists I[(G\ I) \wedge (H\ I)], \square CNm : man \Rightarrow Sf$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boxed{\Box CNm} \Rightarrow CNm}}{}\!\!}{}}{}}{}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{CNm \Rightarrow CNm}}{\boxed{\Box CNm} \Rightarrow CNm}\,\Box L \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nm \Rightarrow Nm}}{\boxed{\blacksquare Nm} \Rightarrow Nm}\,\blacksquare L \qquad \dfrac{\dfrac{\dfrac{}{Nm \Rightarrow Nm}}{\boxed{\blacksquare Nm} \Rightarrow Nm}\,\blacksquare L \qquad \dfrac{}{Sf \Rightarrow Sf}}{\blacksquare Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf}\,\backslash L}{\blacksquare Nm, \boxed{(Nm\backslash Sf)/Nm}, \blacksquare Nm \Rightarrow Sf}\,/L}{\blacksquare Nm, \boxed{\blacksquare((Nm\backslash Sf)/Nm)}, \blacksquare Nm \Rightarrow Sf}\,\blacksquare L}{\blacksquare Nm, \blacksquare((Nm\backslash Sf)/Nm), 1 \Rightarrow Sf\uparrow\blacksquare Nm}\,\uparrow R \qquad \dfrac{}{Sf \Rightarrow Sf}}{\blacksquare Nm, \blacksquare((Nm\backslash Sf)/Nm), \boxed{(Sf\uparrow\blacksquare Nm)\downarrow Sf} \Rightarrow Sf}\,\downarrow L}{\blacksquare Nm, \blacksquare((Nm\backslash Sf)/Nm), \boxed{((Sf\uparrow\blacksquare Nm)\downarrow Sf)/CNm}, \Box CNm \Rightarrow Sf}\,/L}{\blacksquare Nm, \blacksquare((Nm\backslash Sf)/Nm), \boxed{\blacksquare(((Sf\uparrow\blacksquare Nm)\downarrow Sf)/CNm)}, \Box CNm \Rightarrow Sf}\,\blacksquare L$$

$\exists C[({}^{\vee}man\ C) \wedge [j = C]]$

(7-98) **john+finds+a+unicorn** : Sf

$\blacksquare Nm : j, \Box((NA\backslash Sf)/NB) : \mathit{finds}, \blacksquare(((SC\uparrow\blacksquare ND)\downarrow SC)/CND) : \lambda E\lambda F\exists G[(E\ G) \wedge (F\ G)], \Box CNn : \mathit{unicorn} \Rightarrow Sf$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{CNn \Rightarrow CNn}}{\boxed{\Box CNn} \Rightarrow CNn}\,\Box L \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nn \Rightarrow Nn}}{\boxed{\blacksquare Nn} \Rightarrow Nn}\,\blacksquare L \qquad \dfrac{\dfrac{\dfrac{}{Nm \Rightarrow Nm}}{\boxed{\blacksquare Nm} \Rightarrow Nm}\,\blacksquare L \qquad \dfrac{}{Sf \Rightarrow Sf}}{\blacksquare Nm, \boxed{Nm\backslash Sf} \Rightarrow Sf}\,\backslash L}{\blacksquare Nm, \boxed{(Nm\backslash Sf)/Nn}, \blacksquare Nn \Rightarrow Sf}\,/L}{\blacksquare Nm, \boxed{\Box((Nm\backslash Sf)/Nn)}, \blacksquare Nn \Rightarrow Sf}\,\Box L}{\blacksquare Nm, \Box((Nm\backslash Sf)/Nn), 1 \Rightarrow Sf\uparrow\blacksquare Nn}\,\uparrow R \qquad \dfrac{}{Sf \Rightarrow Sf}}{\blacksquare Nm, \Box((Nm\backslash Sf)/Nn), \boxed{(Sf\uparrow\blacksquare Nn)\downarrow Sf} \Rightarrow Sf}\,\downarrow L}{\blacksquare Nm, \Box((Nm\backslash Sf)/Nn), \boxed{((Sf\uparrow\blacksquare Nn)\downarrow Sf)/CNn}, \Box CNn \Rightarrow Sf}\,/L}{\blacksquare Nm, \Box((Nm\backslash Sf)/Nn), \boxed{\blacksquare(((Sf\uparrow\blacksquare Nn)\downarrow Sf)/CNn)}, \Box CNn \Rightarrow Sf}\,\blacksquare L$$

$\exists C[({}^{\vee}unicorn\ C) \wedge (({}^{\vee}finds\ C)\ j)]$

Space limitations prevent further illustration, but note for example that the type for the indefinite has a modal hypothetical subtype, which will allow an indefinite to take scope at a higher clause, but that the type for *every* has a nonmodal hypothetical subtype and will therefore only take scope in its local clause (Montague did not capture this distinction), and note that the accusative pronoun *her* has a type allowing it to take a preceding antecedent in its own clause or in an embedded clause by secondary wrap, but with use of the difference operator expressing the Principle B condition that the antecedent cannot be the subject (Montague did not capture this effect either).

Acknowledgement I thank three LACL'11 reviewers for valuable comments. The research reported in the present paper was supported by DGICYT project SESAAME-BAR (TIN2008-06582-C03-01).

References

1. J. M. Andreoli. Logic programming with focusing in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.
2. David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics*, volume 11 of *Synthese Language Library*. D. Reidel, Dordrecht, 1981.
3. H. Hendriks. *Studied flexibility. Categories and types in syntax and semantics.* PhD thesis, Universiteit van Amsterdam, ILLC, Amsterdam, 1993.
4. Mark Hepple. *The Grammar and Processing of Order and Dependency.* PhD thesis, University of Edinburgh, 1990.
5. Mark Hepple. Normal form theorem proving for the Lambek calculus. In H. Karlgren, editor, *Proceedings of COLING*, Stockholm, 1990.
6. Gerhard Jäger. *Anaphora and Type Logical Grammar*, volume 24 of *Trends in Logic – Studia Logica Library.* Springer, Dordrecht, 2005.
7. E. König. Parsing as natural deduction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vancouver, 1989.
8. Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958. Reprinted in Buszkowski, Wojciech, Wojciech Marciszewski, and Johan van Benthem, editors, 1988, *Categorial Grammar*, Linguistic & Literary Studies in Eastern Europe volume 25, John Benjamins, Amsterdam, 153–172.
9. Michael Moortgat. *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Foris, Dordrecht, 1988. PhD thesis, Universiteit van Amsterdam.
10. Glyn Morrill. Intensionality and Boundedness. *Linguistics and Philosophy*, 13(6):699–726, 1990.
11. Glyn Morrill and Oriol Valentín. On Anaphora and the Binding Principles in Categorial Grammar. In Anuj Dawar and Ruy de Queiroz, editors, *Proceedings of the 17th International Workshop on Logic, Language, Information and Computation, WoLLIC 2010, Brasília*, number LNAI 6188 in Lecture Notes in Artificial Intelligence, pages 176–190, Berlin, 2010. Springer.
12. Glyn Morrill, Oriol Valentín, and Mario Fadda. The Displacement Calculus. *Journal of Logic, Language and Information*, 20(1):1–48, 2011. Doi 10.1007/s10849-010-9129-2.
13. Glyn V. Morrill. *Type Logical Grammar: Categorial Logic of Signs.* Kluwer Academic Publishers, Dordrecht, 1994.
14. Glyn V. Morrill. *Categorial Grammar: Logical Syntax, Semantics, and Processing.* Oxford University Press, 2010.
15. J. van Benthem. *Language in Action: Categories, Lambdas, and Dynamic Logic.* Number 130 in Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1991. Revised student edition printed in 1995 by the MIT Press.