

# Parsing with polymorphism \*

Martin Emms,  
The CIS  
Leopoldstr 139  
8000 Munchen 40  
Germany

## Abstract

Certain phenomena resist coverage within the Lambek Calculus, such as scope-ambiguity and non-peripheral extraction. I have argued in previous work that an extension called Polymorphic Lambek Calculus (PLC), which adds variables and their universal quantification, covers these phenomena. However, a major problem is the absence of a known decision procedure for PLC grammars. This paper proposes a decision procedure which covers a subset of all the possible PLC grammars, a subset which, however, includes the PLC grammars with wide coverage. The decision procedure is shown to be terminating, and correct, and a Prolog implementation of it is described.

## 1 The Lambek Calculus

To begin, I give a brief description of Lambek categorial grammar [Lambek, 1958]. The categories are built up from basic categories, using the binary categorial connectives  $/$  and  $\backslash$ .<sup>1</sup> Then a set of 'categorial rules' involving these categories is defined, of the form:  $x_1, \dots, x_n \Rightarrow y$  ( $n \geq 1$ ),  $x_i$  and  $y$  being categories. A distinctive feature is that the set of rules is defined *inductively*. Using a term adopted from

\*This work was done whilst the author was in receipt of a six month scholarship from the German Academic Exchange Service, whose support is gratefully acknowledged

<sup>1</sup>Lambek also considered a third connective, the 'product'. I, in common with several authors, use the name Lambek calculus to refer to what is really the product-free calculus

logic, *sequent*, in place of 'categorial rule', Lambek presented this inductive definition as a close variant of Gentzen's sequent calculus for propositional logic. Lambek's calculus,  $L^{(/, \backslash)}$ , is given below:

$$\begin{array}{l}
 (\text{Ax}) \quad x \Rightarrow x \\
 (/L) \quad \frac{U, y, V \Rightarrow w \quad T \Rightarrow x}{U, y/x, T, V \Rightarrow w} /L \\
 (\backslash L) \quad \frac{T \Rightarrow x \quad U, y, V \Rightarrow w}{U, T, y \backslash x, V \Rightarrow w} \backslash L \\
 (/R) \quad \frac{T, x \Rightarrow y}{T \Rightarrow y/x} /R \quad (\backslash R) \quad \frac{x, T \Rightarrow y}{T \Rightarrow y \backslash x} \backslash R
 \end{array}$$

Here  $U, T, V$  are sequences of categories ( $U, V$  possibly empty),  $w, x, y$  are categories. In the two premise rules, the  $T \Rightarrow x$  premise is called the *minor* premise. The fact that  $L^{(/, \backslash)}$  derives  $r$ , I will notate as  $L^{(/, \backslash)} \vdash r$ . With regard to the names of the rules, 'L' and 'R' stand for left and right. For example,  $(\backslash L)$  (resp.  $(\backslash R)$ ), derives sequents with  $\backslash$  on the *left* (resp. on the *right*) of the sequent arrow,  $\Rightarrow$ . For various purposes it is convenient to consider the addition of the 'Cut' rule, given below (in which  $x$  is referred to as the Cut formula, and  $T \Rightarrow x$  as the minor premise):

$$\frac{U, x, V \Rightarrow w \quad T \Rightarrow x}{U, T, V \Rightarrow w} \text{Cut}$$

Lambek [1958] establishes that  $L^{(/, \backslash)} + \text{Cut} \vdash r$  iff  $L^{(/, \backslash)} \vdash r$  (Cut elimination), and that  $L^{(/, \backslash)} \vdash r$  is decidable.

The proof of the decidability of  $L^{(\cdot, \setminus)} \vdash r$  proceeds as follows. First one reads the rules of  $L^{(\cdot, \setminus)}$  ‘backwards’, as a set of rewrites, growing a tree at its leaves ‘up the page’. Call the trees grown this way *deduction trees*.  $L^{(\cdot, \setminus)} \vdash r$  iff  $r$  is the root of a deduction tree whose leaves are all *axioms*. It remains to note that there are only finitely many deduction trees for a given sequent: a leaf can be grown in at most a finite number of different ways, and the added daughters have always a diminished complexity (complexity measured as number of occurrences of connectives). This decision procedure is improved upon somewhat if the rules of the calculus are expressed as a Prolog data base of conditionals concerning a binary predicate `seq`, holding between a list of categories and a single category. For later reference, let `Lam` stand for some such Prolog implementation of  $L^{(\cdot, \setminus)}$ .

A grammar,  $G$ , in this perspective is an assignment of categories to words. Reading  $G \vdash s \in y$  as ‘according to  $G$ ,  $s$  has category  $y$ ’, I will say  $G \vdash s \in y$ , if (i)  $s$  is lexically assigned  $y$ , or (ii)  $s = s_1 \dots s_n$  ( $n \geq 1$ ),  $G \vdash s_i \in x_i$ , and  $L^{(\cdot, \setminus)} \vdash x_1, \dots, x_n \Rightarrow y$ .

For any Lambek grammar,  $G$ , the question whether  $G \vdash s \in x$  is decidable. This is got by combining Cut elimination with the decidability of  $L^{(\cdot, \setminus)} \vdash r$ . Consider deciding whether  $G \vdash s_1 s_2 \in z$ , where  $s_1$  and  $s_2$  are lexically assigned the categories  $x$  and  $y$ . One can first check whether  $L^{(\cdot, \setminus)} \vdash x, y \Rightarrow z$ , which is decidable. If  $L^{(\cdot, \setminus)} \not\vdash x, y \Rightarrow z$ , then one should try a ‘non-flat’ categorisation possibility. That is, one should also consider derivable categorisations of the subexpressions, namely  $x'$  and  $y'$  such that  $L^{(\cdot, \setminus)} \vdash x \Rightarrow x'$ ,  $y \Rightarrow y'$ , and check whether they may be combined to give  $z$ . Here lurks a problem, because there are *infinitely* many  $x'$  and  $y'$  such that  $L^{(\cdot, \setminus)} \vdash x \Rightarrow x'$ ,  $y \Rightarrow y'$ . The way out of this problem is the relationship between the ‘non-flat’ categorisation strategy and Cut-based proofs, to illustrate which, note that if there were derivable categorisations,  $x'$  and  $y'$  of the subexpressions, which combined to give  $z$ , then  $L^{(\cdot, \setminus)} \vdash x, y \Rightarrow z$ :

$$(1) \quad \frac{\frac{\frac{\vdots}{x \Rightarrow x'} \quad \frac{\frac{\vdots}{y \Rightarrow y'} \quad \frac{\vdots}{x', y' \Rightarrow z}}{x', y \Rightarrow x'} \text{Cut}}{x, y \Rightarrow z} \text{Cut}}$$

So parsing with an  $L^{(\cdot, \setminus)}$  grammar comes to deciding the derivability of  $x_1, \dots, x_n \Rightarrow s$ , where  $x_i$  are the categories of the lexical items.

This Lambek style of grammar is associated also with a certain method for assigning meanings to strings. The idea is that a proof,  $\gamma$ , of  $L^{(\cdot, \setminus)}$  can be mapped into a semantic operation,  $\mathcal{G}$ . So, if there is a proof,  $\gamma$ , of  $x_1, \dots, x_n \Rightarrow y$ , then a sequence of expressions with categories  $x_1, \dots, x_n$  and meanings

$m_1, \dots, m_n$ , has a possible meaning  $\mathcal{G}(m_1, \dots, m_n)$ . As to which operation,  $\mathcal{G}$ , goes with which proof,  $\gamma$ , this is defined by a *term-associated* calculus. Representative parts of the (extensionally) term associated calculus,  $L_e^{(\cdot, \setminus)}$ , are given below:

$$\begin{aligned} (\text{Ax}) \quad & x : \alpha \Rightarrow x : \alpha \\ (/L) \quad & \frac{U, y : \alpha(\beta), V \Rightarrow w : \epsilon \quad T \Rightarrow x : \beta}{U, y/x : \alpha, T, V \Rightarrow w : \epsilon} /L \\ (/R) \quad & \frac{T, x : \zeta \Rightarrow y : \alpha}{T \Rightarrow y/x : \lambda\zeta\alpha} /R \end{aligned}$$

There are corresponding  $(\setminus L)$  and  $(\setminus R)$  rules.  $L_e^{(\cdot, \setminus)}$  derives sequents where in place of categories there are category:term pairs. If we start with an  $L^{(\cdot, \setminus)}$  proof of  $r$ , and add variables to the antecedent categories of  $r$ , there is a unique way to add terms to the rest of the proof so as to get a proof of  $L_e^{(\cdot, \setminus)}$ . When this is done the term,  $\alpha$ , associated with the succedent of  $r$ , represents the semantic operation. The above mentioned decision procedure can be embellished to develop trees featuring semantic terms, some of them unknown, together with an evolving set of equations in these unknowns. When a proof is discovered, the term for that proof can be obtained by solving the set of equations.

There is a semantic question to be asked about the acceptability of parsing simply by search through  $L^{(\cdot, \setminus)}$  proofs: are all term-associated proofs for a sequent in  $L^{(\cdot, \setminus)} + \text{Cut}$  equivalent to some term-associated proof in  $L^{(\cdot, \setminus)}$ , and vice-versa? The answer is yes [Hendriks, 1989], [Moortgat, 1989].

## 2 Polymorphism

Despite the great simplicity of Lambek grammars, a surprising amount of coverage is possible [Moortgat, 1988]. Two aspects of this are embryonic accounts of extraction, and scope-ambiguity, the latter arising from the fact that there may be more than one proof of a given sequent. However, the accounts possible have remained only partial. Non-peripheral extraction remained unaccounted for (eg. the  $(\text{man})_i$  who Dave told  $e_i$  to leave) and only the scope-ambiguities of peripheral quantifiers are covered (as in the structure QNP TV QNP). A simple account of cross-categorial coordination has also often been cited as an attractive feature of Lambek grammars ([Moortgat, 1988]). However, the analyses are never in a purely Lambek grammar. Belonging to Lambek grammar proper is a part assigning some category to the strings to be coordinated, and then lying without Lambek grammar, a coordination schema, such as  $x$ , and,  $x \Rightarrow x$ .

To overcome these deficits in coverage, I have proposed a *polymorphic* extension of the calculus. Added to the categorial vocabulary are category variables and their universal quantification, allowing such categories as:  $X, X/X, \forall X.X/(X \setminus np)$ . To  $L^{(\cdot, \setminus)}$  are added left and right rules for  $\forall$ , to give what I will call  $L^{(\cdot, \setminus, \forall)}$  (I given straightaway the term-associated calculus):

$$(\forall L) \frac{U, x[y/Z] : \alpha(a), V \Rightarrow w : \Phi}{U, \forall Z.x : \alpha, V \Rightarrow w : \Phi}$$

$$(\forall R) \frac{T \Rightarrow x : \alpha}{T \Rightarrow \forall Z.x : \Delta\omega\alpha} \quad [Z \text{ is not free in } T]$$

Notation: the terms are drawn from the language of 2nd Order Polymorphic  $\lambda$ -calculus [Girard, 1972], [Reynolds, 1974]. Here, terms carry their type as a superscript, and one can have variables in these types (eg.  $\lambda x^\pi.x^\pi$ ), one can abstract over such variable types, deriving terms of *quantified* type (eg.  $\Delta\pi.\lambda x^\pi.x^\pi$ , of type  $\forall\pi(\pi \rightarrow \pi)$ ), and terms of quantified type can be applied to types (eg.  $\Delta\pi.\lambda x^\pi.x^\pi(t)$ , of type  $(t \rightarrow t)$ ). In the  $(\forall L)$  rule above, the type,  $\alpha$ , that  $\alpha$  is applied to, is the type that corresponds to the category variable,  $Z$ .<sup>2</sup> An equivalent slight variant on  $L^{(\cdot, \setminus, \forall)}$  takes as axioms only those  $x \Rightarrow x$  sequents where  $x$  is basic or a variable, something I will call  $L_0^{(\cdot, \setminus, \forall)}$ . It is easy to show  $L_0^{(\cdot, \setminus, \forall)} \vdash r$  iff  $L^{(\cdot, \setminus, \forall)} \vdash r$  (see [Emms and Leiss, forthcoming]).

By assigning conjunctions to  $\forall X.((X \setminus X)/X)$ , negation to  $\forall X.X/X$ , and quantifiers to  $\forall X.X/(X \setminus np)$  and  $\forall X.X \setminus (X \setminus np)$ , one obtains coverage of cross-categorial coordination and negation, as well as a comprehensive account of quantifier scope ambiguity [Emms, 1989],[Emms, 1991]. Assigning relativisers to  $\forall X.((cn \setminus cn)/(s \setminus X)/(X \setminus np))$ , non-peripheral extraction can also be handled [Emms, 1992]. The meanings that go along with these categories are as follows. Where  $\mathcal{L}$  is  $\mathcal{Q}, \mathcal{J}$  or  $\mathcal{N}$ , let  $\mathcal{L}_G$  vary over the conventional meanings of quantifiers, junctions and negation, with  $\mathcal{L}_P$  the *polymorphic* version.

$$\begin{aligned} \mathcal{L}_P(t) &= \mathcal{L}_G \\ \mathcal{Q}(a \rightarrow b)(P^{e \rightarrow a \rightarrow b})(x^a) &= \mathcal{Q}(b)(y^e \rightarrow P y x) \\ \mathcal{J}(a \rightarrow b)(P_1^{a \rightarrow b})(P_2^{a \rightarrow b})(x^a) &= \mathcal{J}(b)(P_1 x)(P_2 x) \\ \mathcal{N}(a \rightarrow b)(P^{a \rightarrow b})(x^a) &= \mathcal{N}(b)(P x) \\ who(a)(P_1^{e \rightarrow a})(P_2^{a \rightarrow t})(Q^{e \rightarrow t})(x^e) &= P_2(P_1 x) \wedge Q x \end{aligned}$$

I will give two illustrations. The proof below would allow the embedded quantifier, *every man*, to be assigned a de-re interpretation in *John believes every man walks*. Note  $(s \setminus np) \setminus ((s \setminus np)/s) = \bar{X}$ .

<sup>2</sup>The  $(\forall R)$  given is a cut-down version of the ‘official’ version, which allows a change of bound variable

$$\frac{\frac{\frac{\vdots}{np, (s \setminus np)/s, \bar{X} \Rightarrow s} \setminus L}{np, (s \setminus np)/s, \bar{X}/(\bar{X} \setminus np) \setminus np \Rightarrow s} \setminus L}{np, (s \setminus np)/s, \forall X.X/(X \setminus np), s \setminus np \Rightarrow s} \forall L$$

Now assuming  $j, bel, em$  and  $walk$  were the terms associated with the antecedents of the root sequent, the term for the proof is:

$$\begin{aligned} &emp_P(tet, et)(\lambda x \lambda f \lambda y [f(walk(x))(y)])(bel)(j) \\ &\text{We obtain as a possible denotation for John believes every man walks:} \\ &emp_P(tet, et)(x, f, y \mapsto f(walk(x))(y))(bel)(j) \\ &= emp_P(et)(x, y \mapsto bel(walk(x))(y))(j) \\ &= emp_P(t)(x \mapsto bel(walk(x))(j)) \\ &= em_G(x \mapsto bel(walk(x))(j)) \end{aligned}$$

As an illustration of non-peripheral extraction, the proof below allows the string *who John told to go* to be recognised as a postmodifier of a common noun:

$$\frac{\frac{\frac{\frac{\vdots}{s/vpc, vpc \Rightarrow s} \setminus R}{r \quad vpc \Rightarrow s \setminus \bar{X}} \setminus L}{(cn \setminus cn)/(s \setminus \bar{X}), vpc \Rightarrow cn \setminus cn} \setminus L}{(cn \setminus cn)/(s \setminus \bar{X})/(\bar{X} \setminus np), np, V, vpc \Rightarrow cn \setminus cn} \setminus L}{\forall X.((cn \setminus cn)/(s \setminus X)/(X \setminus np)), np, V, vpc \Rightarrow cn \setminus cn} \forall L$$

Here  $r = cn \setminus cn \Rightarrow cn \setminus cn$ ,  $V = ((s \setminus np)/vpc)/np$ ,  $\bar{X} = s/vpc$ . Assuming  $who, j, told$ , and  $go$  were associated with the antecedents of the root, the term for the proof is:

$$\begin{aligned} &who((et, t)(\lambda x \lambda y [told(x)(y)(j)])(\lambda f [f(go)]) \\ &\text{We obtain for the denotation of the string who John told to go:} \\ &who((et, t)(x, y \mapsto told(x)(y)(j))(f \mapsto f(go)) \\ &= Q, z \mapsto ((f \mapsto f(go))(y \mapsto told(z)(y)(j))) \wedge Q(z) \\ &= Q, z \mapsto (told(z)(go)(j) \wedge Q(z)) \end{aligned}$$

For the further discussion of the analyses within an  $L^{(\cdot, \setminus, \forall)}$  grammar that cover a significant range of data, see the earlier references. I turn now to the main problem which this paper addresses: is there an automatic procedure able to find these analyses?

## 2.1 Cut Elimination for $L^{(\cdot, \setminus, \forall)}$

We want a procedure to decide whether  $G \vdash s \in x$ , where  $G$  is an  $L^{(\cdot, \setminus, \forall)}$  grammar. As with  $L^{(\cdot, \setminus)}$  grammars, this problem reduces to deciding  $L^{(\cdot, \setminus, \forall)} \vdash r$  if it can be shown both that Cut can be eliminated, and without the loss of any significant semantic diversity. This has recently been shown ([Emms and Leiss, forthcoming]). I make some remarks on the proof. The strategy of the proof of Cut elimination for  $L^{(\cdot, \setminus)}$  starts from the observation that a proof,  $\gamma$ ,

using Cut must contain at least one use of Cut which dominates no further uses of Cut – a ‘topmost’ use of Cut. Suppose this use of Cut derives  $r$ . Then one defines two things: a *degree* of the Cut leading to  $r$ , and a *transformation* taking the proof of  $r$  to an alternative proof of  $r$ , such that either the transformed proof of  $r$  is Cut-free, or it is a proof with 2 or less cuts of lesser *degree*. After a finite number of iterations of the transformation, one must have a cut free proof.

In the proof for  $L^{(\wedge)}$ , the *degree* of a Cut inference is simply the sum of the numbers of connectives in the two premises. This cannot be the degree for  $L^{(\wedge, \forall)}$ . For example, a case to be considered is where one has a cut of the kind shown in (2). The natural rewrite is (3) (that  $T \Rightarrow y[a/Z]$  is provable relies on the fact that  $Z$  is not free in  $T$  and substitution for free variables preserves derivability [Emms and Leiss, forthcoming])

$$(2) \quad \frac{\frac{T \Rightarrow y \quad \forall R \quad U, y[a/Z], V \Rightarrow w}{T \Rightarrow \forall Z.y \quad U, \forall Z.y, V \Rightarrow w} \forall L}{U, T, V \Rightarrow w} \text{Cut}$$

$$(3) \quad \frac{T \Rightarrow y[a/Z] \quad U, y[a/Z], V \Rightarrow w}{U, T, V \Rightarrow w} \text{Cut}$$

With degree defined by number of connectives, we need that the number of connectives in  $y[a/Z]$  is strictly less than the number in  $\forall Z.y$ , and that is often false. The proof goes through instead by taking the degree of a cut to be the sum of sizes of the proofs of its two premises, where the size is the number of nodes in the proof.<sup>3</sup>

## 2.2 Difficulties in deciding $L^{(\wedge, \forall)} \vdash T \Rightarrow x$

So the problem reduces to one of  $L^{(\wedge, \forall)}$  derivability. Whether  $L^{(\wedge, \forall)}$  derivability is decidable I do not know. The nearest to an answer to this that the logical literature comes is a result that quantified intuitionistic propositional logic is undecidable [Gabbay, 1974]. The difference between  $L^{(\wedge, \forall)}$  and logic of this result is the presence of the further connectives ( $\vee, \wedge$ ), and the availability of all structural rules. I will describe below some of the problems that arise when some natural lines of thought towards a decision procedure are pursued.

One might start by considering the logic that is  $L^{(\wedge)} + (\forall R)$ . This can be argued to be decidable in the same fashion as  $L^{(\wedge)}$ : read  $(\forall R)$  backwards as a rewrite, adding another way to build deduction trees. As for  $L^{(\wedge)}$  a sequent has only finitely many deduction trees, and provability is equivalent to the existence of a deduction tree with axiom leaves.

<sup>3</sup>In fact nodes above axiom form sequents are not counted in the size, and the proof relies on changes of bound variable and substitutions not changing the size of  $L^{(\wedge, \forall)}$  proofs

However, when  $(\forall L)$  is added this simple argument will not work: if  $(\forall L)$  is read backwards as a further clause in the definition of deduction trees, then a leaf containing an antecedent  $\forall$  could be rewritten *infinitely* many different ways. A natural move at this point is to redefine deduction trees, reading the  $(\forall L)$  rule as an instruction to substitute an *unknown*. One hopes then that: (i) the set of so-defined deduction trees for a given sequent,  $r$ , is finite (ii) there is some easy to check property,  $P$ , of these trees such that the existence of a  $P$ -tree in the set would be equivalent to  $L^{(\wedge, \forall)} \vdash r$ . Now, if we were considering the combination of *first-order* quantification with the Lambek calculus, this strategy works, but whether it works for  $L^{(\wedge, \forall)}$  remains unknown.

I will go through the application of the strategy in the first-order case to highlight why  $L^{(\wedge, \forall)}$  does not yield so easily. The first-order quantification plus the Lambek calculus, I will call  $L^{(\wedge, \forall^1)}$ . It is the endpoint of a certain line of thought concerning agreement phenomena. One first reanalyses basic categories, such as  $s$  and  $np$ , as being built up by the application of a predicate to some arguments, giving categories such as  $np(3rd, sing)$ ,  $s(fin)$ . It is natural then to consider quantification over the first order positions, such as  $\forall p$ .  $s(fin) \setminus np(p, pl)$ , which could be used when, as in English, the plural forms of a verb are not distinguished according to person. Now  $L^{(\wedge, \forall^1)}$  is decidable, which can be shown by adapting an argument that shows that when the contraction rule is dropped from classical predicate logic, it becomes decidable [Mey, 1992]. Deduction trees for a sequent,  $r$ , of  $L^{(\wedge, \forall^1)}$  are defined so that the rewrite associated with the  $(\forall L)$  rule substitutes an *unknown*. There are then only finitely many deduction trees (the absence of the structural rule of contraction is essential here). Now, if  $L^{(\wedge, \forall^1)} \vdash r$ , and  $r$  has a complex first order term, one can be sure that this term is present in an axiom, because no rules build complexity in the places in categories where a bound variable can occur. For this reason, the so-defined deduction trees for  $r$  cover all the possible *patterns* for a proof of  $r$ . Provability is therefore equivalent to the existence of a substitution making one of the deduction trees have axiom leaves, and this can be checked using resolution.

This situation does not wholly carry over to  $L^{(\wedge, \forall)}$ . The ‘substitute an unknown’ rewrite reading of  $(\forall L)$  defines only finitely many deduction trees for a sequent,  $r$ . However, these so-defined deduction trees for  $r$  do not cover all the possible *patterns* for a proof of  $r$ : unlike  $L^{(\wedge, \forall^1)}$ , there are rules that build complexity in the places in categories where a bound variable can occur. So, for example,  $L^{(\wedge, \forall)} \vdash np, \forall X.X/(X \setminus np)$ ,  $(s \setminus np) \setminus np$ , but none of the deduction trees represents the pattern of the proof. So to check for the existence of a deduction tree (as above defined) that by a substitution would have ax-

iom leaves is not sufficient to decide derivability. It seems we must defined the looked for property,  $P$ , of deduction trees recursively, so that a tree has  $P$  if (1) the leaves by a substitution become axioms, or (2) by hypothesising a connective in one of the unknowns, and extending the tree by rewrites licensed by this connective, one obtains a  $P$ -tree.

It would amount to the same thing if the definition of deduction tree was extended (by hypothesising a connective in an unknown), and the looked for property,  $P$ , kept simple: a tree whose leaves by a substitution become axioms. However, the extended definition of deduction tree now allows *infinitely* many trees for a sequent. This may seem surprising, but is seen one considers a leaf such as  $T \Rightarrow X$ . One can hypothesis  $X = Y/Z$ , extend the deduction tree by the rewrite associated with a slash Right rule, obtaining once again a leaf with a succedent occurrence of an unknown. By imposing a control strategy which would systematically consider all deduction trees of height  $h$ , before deduction trees of height  $h + 1$ , one can be sure that any provable sequent would sooner or later be accepted by the decision procedure (because its provability would entail the existence of a deduction tree of a certain finite height). However, there is no reason to expect the procedure to terminate when working on an undervivable sequent.<sup>4</sup>

### 3 A partial decision procedure for $L_{/,\wedge,\forall}$

While there are problems in the way of a general decision procedure for  $L_{/,\wedge,\forall}$ , I claim a *partial* decision procedure for  $L_{/,\wedge,\forall}$  is possible. Partial in the sense of covering only a certain class of sequents, but one sufficiently large, I claim, to cover all linguistically relevant cases. The procedure will be a partial decision procedure for  $L_{/,\wedge,\forall}$  via being a partial decision procedure for  $L_0^{/,\wedge,\forall}$ .

To describe the class of sequents that the procedure applies to I need definitions of the ‘polarity’ of an occurrence of a category. Let the *category polarity* of an occurrence of  $x$  in a category  $y$  ( $pol(x, y)$ ) be:

$$\begin{aligned} pol(x, x) &= + \\ \text{if } x \text{ occurs in } y, \quad pol(x, y/z) &= pol(x, \forall Z.y) = \\ pol(x, y) &= opp(pol(x, z/y)) \end{aligned}$$

Here  $opp(+)$  =  $-$ ,  $opp(-)$  =  $+$ . The *sequent polarity* of an occurrence of  $x$  in  $y$  in a sequent  $r$  is the same as the category polarity if  $y$  is an antecedent, and otherwise it is opposite. I use ‘polarity’ as short for ‘sequent polarity’. An example:

$$(4) \quad s \backslash (\forall^- X.X / (X \backslash np)) \Rightarrow s \backslash (\forall^+ X.X / (X \backslash np))$$

<sup>4</sup>I have found non-terminating consecutively bounded depth first search to happen on the Prolog implementation of the calculus that these paragraphs suggest

The decision procedure to be described is applicable to sequents whose *negative* occurrences of polymorphic categories are unlimited, but whose positive polymorphic categories are drawn from:

$$(5) \quad \begin{aligned} &\forall X.X / (X \backslash np), \forall X.X \backslash (X / np), \\ &\forall X.X / X, \\ &\forall X.((cn \backslash cn) / (s \backslash X)) / (X / np) \\ &\forall X.((X \backslash X) / X), \end{aligned}$$

I will now make three observations concerning proofs in  $L_0^{/,\wedge,\forall}$ , leading up to the definition of the procedure.

**Observation One** In the categories in (5) there is exactly one positive and one or two negative occurrence of the bound variable. This leads to the predictable occurrence of certain sequents. To help describe these I need to define some more terminology.

An *initial labelling* of a proof is the assignment of unique integers to some of the categories in some sequent of the proof. A *completed labelling* is got from an initial labelling by a certain kind of propagation up the tree: a label is passed up when a labelled category is simply copied upward, and in a  $(\forall L)$  inference the label is distributed to the occurrences of the categories chosen for the variable. In other inferences where a labelled category is active, the label is not passed up. For example:

$$(6) \quad \frac{\frac{s_1 \Rightarrow s \quad s \Rightarrow s_1}{s_1/s_1, s \Rightarrow s} \backslash L \quad \frac{np \Rightarrow np \quad s \Rightarrow s}{np, s \backslash np \Rightarrow s} \backslash R}{\forall_1 X.X/X, s \Rightarrow s} \forall L \quad \frac{\frac{\frac{s_1 \Rightarrow s \quad s \Rightarrow s_1}{s_1/s_1, s \Rightarrow s} \backslash L \quad \frac{np \Rightarrow np \quad s \Rightarrow s}{np, s \backslash np \Rightarrow s} \backslash R}{\forall_1 X.X/X, s/2(s \backslash np), s \backslash np \Rightarrow s} \backslash L}{\forall_1 X.X/X, s/2(s \backslash np), s \backslash np \Rightarrow s} \backslash L$$

I will say  $U, a_i, V \Rightarrow w$  is ‘positive for  $\forall_i$ ’ if the sequent occurs in a labelled  $L_0^{/,\wedge,\forall}$  proof and the label on  $a_i$  has been passed from a labelled occurrence of  $\forall_i$ . Correspondingly, call a sequent  $T \Rightarrow a_i$  ‘negative for  $\forall_i$ ’. Now note that in the above proof, the  $\forall_1$  in the root led to one  $\forall_1^+$  and one  $\forall_1^-$  branch. This is no accident: one can predict the existence of such branches in any proof of a sequent with a positive occurrence of  $\forall_i X.X/X$ . To see this, let me first define a notion reflecting how ‘embedded’ a category is:

$$path(a, a) = \langle \rangle.$$

$$\begin{aligned} \text{Where } a \text{ occurs in } x, \quad path(a, x/y) &= \langle /, path(a, x) \rangle, \\ path(a, y/x) &= \langle /, path(a, x) \rangle, \quad path(a, \forall Z.x) = \\ &\langle \forall, path(a, x) \rangle \end{aligned}$$

With the exception of bound variable, if a category occurs with a path  $\langle C, p \rangle$ , and a polarity  $\delta$ , in the conclusion of an inference, then it occurs in the premises of that inference with the same polarity, and with either the same path or with path  $p$ . Also, in leaves of a proof in  $L_0^{/,\wedge,\forall}$ , categories only occur with zero path. Therefore, if we have

a proof of a sequent with a positive occurrence of  $\forall_i X.X/X$  and with non-zero path, then there must occur higher in the proof, a sequent with  $\forall_i X.X/X$  occurring again positively and this time with with zero-path. In other words there must occur a node  $U, \forall_i X.X/X, V \Rightarrow w$ . Then if there were no ( $\forall L$ ) inference in this proof introducing the category  $\forall_i X.X/X$ , the category  $\forall_i X.X/X$  would be present in the leaves of the proof. Because the leaves can only feature basic categories, there must be a ( $\forall L$ ) inference, and therefore a node  $U', a_i/a_i, V' \Rightarrow w'$ . Reasoning in a similar vein concerning the category  $a_i/a_i$ , we can be sure there must be a ( $/L$ ) inference, with premises  $U'', a_i, V'' \Rightarrow w''$  and  $T'' \Rightarrow a_i$ . These are  $\forall_i^+$  and  $\forall_i^-$  sequents.

Provable sequents having a positive occurrence of one of the polymorphic categories from (5), labelled with  $i$ , will generate an  $L_0^{(/, \wedge, \forall)}$  proof such that corresponding to each of the positive and negative occurrences of the bound variable, there are (distinct)  $\forall_i^+$  and  $\forall_i^-$  branches.

**Observation Two** We just argued that in any proof of a sequent with a positive occurrence of quantified category, there must occur a node at which the quantifier is introduced by a ( $\forall L$ ) inference, and that for the categories in (5),  $\forall_i^\delta$  sequents must appear above this. For each of the  $\forall_i^\delta$  sequents, the minimum number of steps there can be between the conclusion of the ( $\forall L$ ) step and the  $\forall_i^\delta$  sequent is the length of the paths to the associated occurrence of the bound variable in the quantified category. Proofs featuring such minimum intervals between the quantified category and the associated  $\forall_i^\delta$  sequents I will call *orderly*. One can ask the question whether whenever there is a proof of a sequent whose positive quantifiers are drawn from the list in (5), there is also an (equivalent) orderly proof. And the answer is that there is.

*Proof sketch* We want to show that for any category  $x$  in (5), for each of the occurrence of a variable in it, that if there is a proof of  $U, x, V \Rightarrow w$ , then there is a proof in which the steps leading from the lowest occurrence of the relevant  $\forall_i^\delta$  sequent to the ( $\forall L$ ) inference correspond to the path to the bound variable in  $x$ .

Let me define the *spine* of a category as:  $sp(x/y) = \langle /, sp(x) \rangle$ ,  $sp(\forall Z.x) = \langle \forall, sp(x) \rangle$ ,  $sp(x) = \langle \rangle$ , where  $x$  is basic.

We will show first for categories such that  $sp(x) = \langle \forall, slash \rangle$ , and  $sp(x) = \langle slash_1, slash_2 \rangle$ , that when there is a proof such that the left inferences for the first two elements of the spine are separated by  $n$  steps there must be an equivalent proof where they are separated by  $n - 1$  steps.

One considers all the possibilities for the *last* intervening step,  $l$ , and shows that the step associated with the first element of the spine could have

been done before  $l$ , thus lowering by 1 the number of steps intervening between the first two elements of the spine. There is not the space to show all the cases. (7), (8) and (9), (10) are representative examples for  $sp(w) = \langle \forall, sp(x) \rangle$ . Note that in (9) and (10) there are side-conditions to the ( $\forall R$ ) inferences. Satisfaction of these for (9) entails satisfaction for (10). (11), (12) and (13), (14) show representative examples for  $sp(w) = \langle slash_1, slash_2 \rangle$ . In (14),  $X'$  is some variable chosen to be not free in  $U, x/y/z, T, V$  and  $w$ . The provability of the upper premise  $U, x/y, V \Rightarrow w[X'/X]$  follows from that of  $U, x/y, V \Rightarrow w$  by substitution for the variable  $X$  throughout.<sup>5</sup> As to the equivalence of the proofs, one can confirm that in the term-associated versions, the same term is paired with the succedent category in each case.

$$(7) \frac{U, a, V_2 \Rightarrow w \quad x'/y', V_1 \Rightarrow b}{U, a/b, x'/y', V_1, V_2 \Rightarrow w} /L$$

$$\frac{U, a/b, x'/y', V_1, V_2 \Rightarrow w}{U, a/b, \forall Z.x/y, V_1, V_2 \Rightarrow w} \forall L$$

$$(8) \frac{x'/y', V_1 \Rightarrow b}{U, a, V_2 \Rightarrow w \quad \forall Z.x/y, V_1 \Rightarrow b} \forall L$$

$$\frac{U, a/b, \forall Z.x/y, V_1, V_2 \Rightarrow w}{U, a/b, \forall Z.x/y, V_1, V_2 \Rightarrow w} /L$$

$$(9) \frac{U, x'/y', V \Rightarrow z}{U, x'/y', V \Rightarrow \forall Y.z} \forall R$$

$$\frac{U, x'/y', V \Rightarrow \forall Y.z}{U, \forall X.x/y, V \Rightarrow \forall Y.z} \forall L$$

$$(10) \frac{U, x'/y', V \Rightarrow z}{U, \forall X.x/y, V \Rightarrow z} \forall L$$

$$\frac{U, \forall X.x/y, V \Rightarrow \forall Y.z}{U, \forall X.x/y, V \Rightarrow \forall Y.z} \forall R$$

$$(11) \frac{U, a, V \Rightarrow w \quad x/y, T_2 \Rightarrow b}{U, a/b, x/y, T_2, V \Rightarrow w} /L$$

$$\frac{U, a/b, x/y, T_2, V \Rightarrow w \quad T_1 \Rightarrow z}{U, a/b, x/y/z, T_1, T_2, V \Rightarrow w} /L$$

$$(12) \frac{x/y, T_2 \Rightarrow b \quad T_1 \Rightarrow z}{U, a, V \Rightarrow w \quad x/y/z, T_1, T_2 \Rightarrow b} /L$$

$$\frac{U, a/b, x/y/z, T_1, T_2, V \Rightarrow w}{U, a/b, x/y/z, T_1, T_2, V \Rightarrow w} /L$$

$$(13) \frac{U, x/y, V \Rightarrow w}{U, x/y, V \Rightarrow \forall Y w[Y/X]} \forall R$$

$$\frac{U, x/y, V \Rightarrow \forall Y w[Y/X] \quad T \Rightarrow z}{U, x/y/z, T, V \Rightarrow \forall Y w[Y/X]} /L$$

$$(14) \frac{U, x/y, V \Rightarrow w[X'/X] \quad T \Rightarrow z}{U, x/y/z, T, V \Rightarrow w[X'/X]} /L$$

$$\frac{U, x/y/z, T, V \Rightarrow w[X'/X]}{U, x/y/z, T, V \Rightarrow \forall Y w[Y/X]} \forall R$$

<sup>5</sup>Here the 'full' version of ( $\forall R$ ) is being used, incorporating a change of bound variable. See earlier footnote.

This is enough to show orderly proofs for  $\forall X.X/X$  and  $\forall X.(X\backslash X)/X$ . For  $\forall X.X/(X\backslash np)$  and  $\forall X.((cn\backslash cn)/(s\backslash X)/(X\backslash np))$  we must further show that if there is a proof of  $T \Rightarrow x/y$  whose last step is not a  $(/R)$  inference introducing  $x/y$ , then there is an equivalent proof whose last step is a  $(/R)$  inference introducing  $x/y$ . One can show this by showing if there is a proof whose last two steps use  $(/R)$  followed by some rule  $*$ , then there is an equivalent proof reversing that order. (15) and (16) illustrate this.

$$(15) \frac{\frac{U, a, V, y \Rightarrow x}{U, a, V \Rightarrow x/y} /R \quad T \Rightarrow b}{U, a/b, T, V \Rightarrow x/y} /L$$

$$(16) \frac{U, a, V, y \Rightarrow x \quad T \Rightarrow b}{U, a, T, V, y \Rightarrow x} /L \quad \frac{U, a/b, T, V \Rightarrow x/y}{U, a/b, T, V \Rightarrow x/y} /R$$

So much by way of a sketch of a proof. I will put the fact that orderly proofs exist to the following use. For sequents whose positive quantifiers are drawn from the list in (5), one can be sure that if they have proofs at all, they have proofs which instantiate quantifiers 'one at a time'. One at time in the sense that once a there is a  $(\forall L)$  inference, one can suppose there will be no more  $(\forall L)$  on the branches leading to the first occurrences of a  $\forall_i^6$  sequents.

**Observation Three** Bearing in mind Observation One, the question whether a given choice,  $a_i$ , for the value of the quantified variable is a good one will come to depend, sooner or later, on the derivability of a certain set of  $\forall_i^6$  sequents, containing one  $\forall_i^+$  sequent and one or two  $\forall_i^-$  sequents. In relation to this consider the following:

**Fact 1 (Unknown elimination)** (i) and (ii) are equivalent

- (i) There is an  $x$  such that  $L^{(/, \backslash, \forall)} \vdash U, x, V \Rightarrow w, T_1 \Rightarrow x, \dots, T_n \Rightarrow x$   
(ii)  $L^{(/, \backslash, \forall)} \vdash U, T_1, V \Rightarrow w, \dots, U, T_n, V \Rightarrow w$

The proof of this, from left to right uses Cut and Cut-Elimination. For example, from  $L^{(/, \backslash, \forall)} \vdash U, x, V \Rightarrow w, L^{(/, \backslash, \forall)} \vdash T_1 \Rightarrow x$ , we deduce  $L^{(/, \backslash, \forall)} \vdash \text{Cut} \vdash U, T_1, V \Rightarrow w$ . Therefore by Cut elimination,  $L^{(/, \backslash, \forall)} \vdash U, T_1, V \Rightarrow w$ . For the right to left direction, let me say that  $(w\backslash U)/V$  is a shorthand for  $(w\backslash u_1 \dots \backslash u_n) / v_n \dots / v_1$ . We choose the  $x$  to be  $(w\backslash U)/V$ . Clearly for this  $x$ ,  $L^{(/, \backslash, \forall)} \vdash U, x, V \Rightarrow w$ . Also each of the claims  $L^{(/, \backslash, \forall)} \vdash T_i \Rightarrow x$ , follows from the assumed  $U, T_i, V \Rightarrow w$ , simply by sufficiently many slash Right inferences.

On the basis of these observations, I suggest the

following decision procedure: <sup>6</sup>

**Definition 1 (Decision procedure)** Where  $\Delta, \Gamma$  vary over possibly empty sequences of sequents, let a rewrite procedure  $\mathcal{R}$  be defined as follows

1.  $\Delta, x \Rightarrow x, \Gamma \rightsquigarrow \Delta, \Gamma$ , where  $x$  is atomic
2.  $\Delta, T \Rightarrow w, \Gamma \rightsquigarrow \Delta, \Theta, \Gamma$ , if  $T \Rightarrow w$  follows from  $\Theta$  by some rule of  $L^{(/, \backslash, \forall)}$  other than  $(\forall L)$
3.  $\Delta, U, \forall Z.x, V \Rightarrow w, \Gamma \rightsquigarrow \Delta, x[X/Z], V \Rightarrow w, \Gamma$ , where  $X$  is an unknown, and there are no other unknowns in  $\Delta, U, \forall Z.x, V \Rightarrow w, \Gamma$
4.  $\Delta, U, X, V \Rightarrow w, T_1 \Rightarrow X, \dots, T_n \Rightarrow X, \Gamma \rightsquigarrow \Delta U, T_1, V \Rightarrow w, \dots, U, T_n, V \Rightarrow w, \Gamma$

A sequent  $T \Rightarrow w$  is accepted iff the sequence consisting of just this sequence can be rewritten to the empty sequence by  $\mathcal{R}$ .

The fourth clause slightly oversimplifies what I intend in the two respects that (i) the rewrite can apply when the  $U, X, V \Rightarrow w, T_1 \Rightarrow X, \dots, T_n \Rightarrow X$  occur dispersed in any order through the sequence, and (ii) it can only apply if the unknown  $X$  does not occur in sequents other than those mentioned. Note because of clause 3, there will only ever be one unknown in the state of the procedure. This corresponds to Observation Two above. I will show that this procedure is terminating and correct when applied to sequents whose positive quantifiers are drawn from (5). By correctness of the procedure, I mean that the procedure accepts  $r$  iff  $L^{(/, \backslash, \forall)} \vdash r$ . The implication left to right I will call soundness, and from right to left completeness.

There is a term associated version of this decision procedure, rewriting a pair consisting of a set of equations, and a sequence of term-associated sequents. On the basis of the discussion earlier, for the most part the the reader should be able to easily imagine what embellishments are required to the clauses of the rewrite. I will just give the full version of the Clause 4 rewrite. The input will be:

Equations:  $\Sigma$

Sequence:  $\Delta, U : \vec{u}, X : \Phi_1, V : \vec{v} \Rightarrow w : \Phi_2, T_1 : \vec{t}_1 \Rightarrow X : \Psi_1, \dots, T_n : \vec{t}_n \Rightarrow X : \Psi_n, \Gamma$

The output will be:

Equations:  $\Sigma$  plus  $\Phi_2 = \Phi_1(\vec{v})(\vec{u}), \Psi_1 = \lambda v_1 \lambda u_1 \Psi'_1, \dots, \Psi_n = \lambda v_n \lambda u_n \Psi'_n$

Sequence:  $\Delta, U : \vec{u}_1, T_1 : \vec{t}_1, V : \vec{v}_1 \Rightarrow w : \Psi'_1, \dots, U : \vec{u}_n, T_n : \vec{t}_n, V : \vec{v}_n \Rightarrow w : \Psi'_n, \Gamma$

### 3.1 Termination

If there are any rewrites possible for a sequence there at most finitely many. So we require that no rewrite series can be infinitely long. Call the sequents featuring an unknown a *linked set*. At any one time

<sup>6</sup>Since writing this paper, I have discovered that the above observation concerning unknown elimination has been made before [Moortgat, 1988], [Bentham, 1990]. This will be further discussed at the end of the paper

there is at most one linked set. Let the degree,  $d$ , of a sequence be the total number of connectives. All rewrites on a sequence that has no linked set lower the degree. So rewriting can only go on finitely long before it stops or a linked set is introduced. A linked set is introduced by a clause 3 rewrite, introducing an unknown into some particular sequent. Call this the *input* sequent. While the sequence contains a linked set, either the degree of the whole sequence goes down, and the sequence remains one containing a linked set (clause 1, clause 2), or the sequence becomes one no longer containing a linked set (clause 4). So a rewrite can only go on finitely long before it either stops, or has a phase where a linked set is introduced and then eliminated. Call the sequents which result from the elimination of the unknown in a clause 4 rewrite, the *output* sequents. Now considering any such phase of unknown introduction followed by elimination, one can say that the count of positive quantifiers in the *input* sequent must be strictly greater than the count of positive quantifiers in any of the *outputs*. This, taken together with the fact that the maximum count of positive quantifiers is never increased outside of such phases, means that there can only be finitely many such phases in a rewrite.

### 3.2 Soundness

We show that if the procedure accepts a sequence of  $n$  sequents ( $n \geq 1$ ), then there is substitution for the unknowns such that there are  $n$  proofs of the  $n$  substituted for sequents. This subsumes soundness, which is where  $n = 1$  and there are no unknowns. I shall use  $sub(\Delta)$  to refer to the sequence of sequents got from  $\Delta$  by some substitution for the unknowns in  $\Delta$ , and  $L^{(\wedge, \vee)} \vdash \Delta$  for the claim that there are proofs of each of the sequents in  $\Delta$ .

The proof is by induction on the length of the shortest accepting rewrite. When the shortest accepting rewrite is of length 1, the sequence must consist simply of an axiom, and so there is a proof. Now suppose the statement is true for all sequences whose shortest accepting rewrite is less than  $l$ . Then for sequences whose shortest accepting rewrite is of length  $l$ , we consider case-wise what the first rewrite might be.

- clause 2 rewrite, for example:  $\Delta, U, x/y, T, V \Rightarrow w, \Gamma \rightsquigarrow \Delta, U, x, V \Rightarrow w, T \Rightarrow y, \Gamma. \Delta, U, x, V \Rightarrow w, T \Rightarrow y, \Gamma$  must have a shortest accepting rewrite of length  $< l$ , so by induction there is a substitution such that  $L^{(\wedge, \vee)} \vdash sub(\Delta), sub(U, x, V \Rightarrow w), sub(T \Rightarrow y), sub(\Gamma)$ . From this it follows that  $L^{(\wedge, \vee)} \vdash sub(\Delta), sub(U, x/y, T, V \Rightarrow w), sub(\Gamma)$ . The other possibilities for clause 2 rewrites work in a similar way

- clause 3 rewrite:  $\Delta, U, \forall Z.x, V \Rightarrow w, \Gamma \rightsquigarrow \Delta, U, x[x/Z], V \Rightarrow w, \Delta$ . By induction there is a substitution such that  $L^{(\wedge, \vee)} \vdash sub(\Delta), sub(U, x[x/Z], V \Rightarrow w), sub(\Delta)$ . Let  $sub'$  be the sub-

stitution that differs from  $sub$  simply by substituting nothing for  $x$ .  $sub'(\forall Z.x) = \forall Z(sub'(x))$ , and  $sub'(x[x/Z]) = sub'(x)[sub'(x)/Z]$ . It follows that  $L^{(\wedge, \vee)} \vdash sub'(\Delta), sub'(U, \forall Z.x, V \Rightarrow w), sub'(\Gamma)$

- clause 4 rewrite.  $\Delta, U, x, V \Rightarrow w, T_1 \Rightarrow x, \dots, T_n \Rightarrow x, \Gamma \rightsquigarrow \Delta, U, T_1, V \Rightarrow w, \dots, U, T_n, V \Rightarrow w, \Gamma$ . By induction:  $L^{(\wedge, \vee)} \vdash sub(\Delta), sub(U, T_1, V \Rightarrow w, \dots, U, T_n, V \Rightarrow w), sub(\Gamma)$ . Let  $sub'$  be the substitution that differs from  $sub$  simply by substituting for  $x$ ,  $sub(w \setminus U/V)$ . Clearly  $L^{(\wedge, \vee)} \vdash sub'(U, x, V \Rightarrow w)$ . Also for each  $T_i$ , it follows from  $L^{(\wedge, \vee)} \vdash sub(U, T_i, V \Rightarrow w)$  that  $L^{(\wedge, \vee)} \vdash sub'(T_i \Rightarrow x)$ . Hence  $L^{(\wedge, \vee)} \vdash sub'(\Delta), sub'(U, x, V \Rightarrow w), sub'(T_1 \Rightarrow x), \dots, sub'(T_n \Rightarrow x), sub'(\Gamma)$   $\square$

### 3.3 Completeness

I will now show completeness for sequents whose positive polymorphic categories are drawn from (5).

By a *frontier*,  $f$ , in a proof, I will mean either the leaves of that proof or the leaves of a subtree having the same root. Given a frontier  $f$  in a proof  $p$ , which has some completed labelling, the procedure will be said to be in a state  $s$  that *corresponds* to  $f$ , if the state and the frontier are identical except that (i)  $s$  may have some axioms deleted as compared with  $f$ , and (ii) the occurrences of labelled, non-quantified  $a_i$  in  $f$ , are transformed to occurrences of some unknown in  $s$ . Given a state  $s$ , I will say that a frontier,  $f$ , is accessible if there is a state corresponding to  $f$  that the procedure may reach from  $s$ .

I assume the procedure is complete for unknown-free sequents whose positive quantifier count is zero.<sup>7</sup> Now suppose the procedure is complete for unknown-free sequents whose positive quantifier count is less than some particular  $n$ , and consider a sequent  $r$ , of positive quantifier count  $n$ , with some proof,  $p$ , and one of the form remarked upon in Observation Two. There will be  $(\forall L)$  inferences in this proof, amongst which is a set lower than any others. Take the conclusion of one such  $(\forall L)$  inference,  $U, \forall X.y, V \Rightarrow w$  and from all other branches pick a point not above a  $(\forall L)$  inference. This set of points forms a frontier,  $f$ ; which is accessible if the procedure starts at  $r$ . Call the corresponding state  $s$ . The sequents in the state other than  $U, \forall X.y, V \Rightarrow w$  are unknown-free, have a positive quantifier count of less than  $n$ , and have a proof, and so by induction the procedure is complete for them. So there is a possible later state  $s'$  which consists solely of the sequent  $U, \forall X.y, V \Rightarrow w$ . We now focus on the subproof of  $p$  that is rooted in  $U, \forall X.y, V \Rightarrow w$ . Consider  $\forall X.y$  as labelled with  $i$ , and labelling to have been propagated up the tree. I want to define a certain accessible frontier,  $f'$ , in this tree. There are a certain finite number of branches ending in  $U, \forall X.y, V \Rightarrow w$ . A certain subset of those

<sup>7</sup>I am of course assuming that all these positive quantified categories are drawn from the list in (5)



branches lead to  $\forall_i^\delta$  sequents, and without any intervening ( $\forall$ L) inferences. Select for the frontier  $f'$  the lowest occurrences for the  $\forall_i^\delta$  sequents. From the other branches simply select a set of nodes,  $\Gamma$ , which is not preceded by a ( $\forall$ L). This frontier is accessible, and the corresponding state is:  $U, X_i, V \Rightarrow w, T_1 \Rightarrow X_i, \dots, T_n \Rightarrow X_i$ . By a clause 4 rewrite this leads to:  $U, T_1, V \Rightarrow w, \dots, U, T_n, V \Rightarrow w$ . This state is unknown free, each of the sequents has positive quantifier count less than  $n$ , and each has a proof. So by induction, the procedure is complete for each of the sequents, and the state may be rewritten to  $\langle \rangle$ .  $\square$

## 4 Implementation

We can with respect to the term-associated version of the decision procedure ask whether it is *semantically comprehensive*: whether the procedure assigns, up to logical equivalence, exactly the same terms to a sequent as are assigned to it by the declarative definition of an  $L^{(\wedge, \vee)}$  grammar. Some but not all parts of what is necessary for a proof of this are established - that Cut elimination for  $L^{(\wedge, \vee)}$  preserves readings, that restriction to orderly proofs loses no readings. However, for the moment, the claim rests ultimately on empirical evidence, drawn from the prolog implementation that I will now describe. I will describe the implementation as additions/alterations to the earlier mentioned Lam.

*First*, it was noted in Observation Two, that one can insist in proof search that Slash right rules are used as soon as their application become possible: this early use of Slash right rules is the first modification of Lam. For the sake of the discussion, assume it is done by adding to non Slash right rules a check on the absence of a slash in the succedent.

*Second*, a conditional for ( $\forall$ L) is added:

```
seq([U, pol(X, Y) : Term1, V], W : Term2) :-
    groundseq([U, pol(X, Y) : Term1, V],
W : Term2),
    substitute(X1, X, Y, Y1), % Y1 is Y[X1/X]
    mark(Y1, Y2),
    seq([U, Y2 : Term1(Ty), V], W : Term2),
    cattotype(X1, Ty).
```

Note, polymorphic categories appear as terms such as  $\text{pol}(x, x/x)$ . The code is in a simplified form, pretending that  $[U, X, V]$  matches any list that is the appending together of the lists  $U, [X]$  and  $V$ , where in reality there are further clauses taking care of this. The conditional basically substitutes an unknown for a quantified variable. Prior to the substitution there is a check, `groundseq`, that the categories in the goal do not already feature some syntactic unknown. Subsequent to substitution, the `mark` relation leads to the replacement of the positive occurrence of the un-

known  $X1$  with  $(X1, a)$ .

*Third*, a goal featuring a zero-path occurrence of  $(X1, a) : \text{Term}$  matches no standard sequent rule, because of the marking, matching instead an 'argument stacking' conditional:

```
seq([U :  $\vec{U}$ , (X, a) : F, V :  $\vec{V}$ ], W : Term) :-
X = ( $\mathcal{W} \setminus U$ ) / V, Term = F( $\vec{V}$ ) ( $\vec{U}$ )
```

*Fourth*, sequents featuring the marked version of the unknown are dealt with *before* sequents featuring the unmarked (negative) instances of the unknown, by ordering the *major* premise before the *minor* in the conditionals for the Slash Left rules.

To illustrate I will 'trace' the behaviour of the program on the goal given as 1 below (`tv` stands for  $(s \setminus np) / np$ )

1. `seq([np:f, tv:g, pol(x, x/(x/np)):h], s:T)`
2. `seq([np:f, tv:g, (X1, a) \ (X1/np):h(Ty)], s:T)`
3. `seq([np:f, (X1, a):h(Ty)(T1)], s:T)`
4. `X1 = s \ np, T = h(Ty)(T1)(f)`
5. `seq([(s \ np) / np : g], s \ np / np : T1)`
6. `T1 =  $\lambda x \lambda y g(x)(y)$`
7. `cattotype(s \ np, Ty)`
8. `Ty = (e, t)`
9. `T = h((e, t))( $\lambda x \lambda y gxy$ )(f)`

1 matches against the ( $\forall$ L) clause. The check that there are no syntactic unknowns around is successful, and after substitution and marking, we reach the subgoal shown as 2, which introduces the new unknowns  $X1$  and  $Ty$ . 2 matches against the ( $\setminus$ L) clause, the first subgoal of which is the *major* premise, shown as 3, with the new unknown  $T1$  (if we could pick the *minor* premise, we would have non-termination). 3 matches only the 'argument stacking' conditional, giving a solution for  $X1$  and solving  $T$  in terms of  $Ty$  and  $T1$ , as shown in 4. The second subgoal of 2 is then considered, under the current bindings, which is 5. 5 will solve via a combination of slash Left and slash Right rules, giving the solution for  $T1$  shown in 6. 2 is now satisfied, and the final subgoal of 1 is considered under the current bindings, which is 7. 7 solves with the solution for  $Ty$  shown in 8. 1 is now satisfied, and the solution for  $T$  is shown in 9 (recall in 4,  $T$  was expressed in terms of  $Ty$  and  $T1$ ).

Space precludes giving a formal argument that this Prolog implementation and the foregoing decision procedure correspond, in the sense that they succeed and fail on the same sequents, and assign the same terms. By way of indication of the behaviour of the implementation, and in particular its semantic comprehensiveness, I give below some examples of what the implementation does by way of assigning readings. In all but the last two cases the task is to reduce to `s`. For the last two it is to reduce to `cn`.

- (17) a. every man walks (1)  
 b. every man loves a woman (2)  
 c. John believes Mary thinks every man walks (3)  
 d. every man a woman 2 flowers (0)  
 e. every man loves a woman 2 flowers (0)  
 f. every man gave a woman 2 flowers (6)  
 g. (omdat) John gek en Mary dom is (1)  
 h. man who John told to go (1)  
 i. man who John told Mary to go (0)

## 5 Concluding remarks

To pick up on an earlier footnote, I have discovered since writing this paper that Benthem and Moortgat have shown decidable, by using what I have referred to as Unknown Elimination, the system which is  $L^{(\cdot, \cdot)}$  with an added rule of 'Boolean Cut':

$$\frac{U, x, V \Rightarrow w \quad T_1 \Rightarrow x \quad T_2 \Rightarrow x}{U, T_1, J, T_2, V \Rightarrow w} \text{Bool.Cut}$$

The question arises then of the relation between their work and what has been proposed in this paper. At the very least, I hope to have shown that there is lurking in this Unknown Elimination technique, an approach not only to coordination, but also to quantifier scope ambiguity and non-peripheral extraction. The main difference between the decision procedure for  $L_0^{(\cdot, \cdot, \forall)}$  and that for  $L^{(\cdot, \cdot)} + \text{Bool.Cut}$  is that the Unknown Elimination technique is put to work on sequents which do not arise from special purpose Cut rules, but simply by the elimination of categorial connectives from certain kinds of categories containing unknowns. This introduces some intricacies into the proof of completeness, which the observation concerning orderly proofs was used to deal with.

As to the scope of the decision procedure, this ought to have a more general specification than that which has been given here, though I have not yet found it. A plausible seeming idea is that there should be one positive and several negative occurrences of a bound variable. However, this includes a category such as  $\forall X.s/(X/X)$ , and a proof featuring this category is not guaranteed to produce *separate*  $\forall^b$  sequents.

A direction for future research would be to investigate the possibility of combining this approach to quantification, coordination and extraction with non-categorial accounts of other aspects of a language. The idea would be to use such a non-categorial grammar as an extended axiom base. If this turned out to be feasible then we would have an attractively *portable* account of quantification, coordination and extraction.

## References

[Benthem, 1990] Johan van Benthem. Categorial

Grammar meets unification. In *Unification formalisms: syntax, semantics and implementation*, J.Wedekind et al.(eds.).

[Emms, 1989] Martin Emms. Polymorphic Quantifiers. In *Proceedings of the Seventh Amsterdam Colloquium*, pages 139–163, Torenvliet, M. S. L. (ed.), Institute for Language, Logic and Information, Amsterdam, December 1989.

[Emms, 1991] Martin Emms. Polymorphic Quantifiers. In *Studies in Categorial Grammar* Barry, G. and Morrill, G. (eds.) , pages 65–112, Volume 5 of Working Papers in Cognitive Science, 1991, Edinburgh, Centre for Cognitive Science.

[Emms, 1992] Martin Emms. Logical Ambiguity. PhD Thesis, Centre of Cognitive Science, Edinburgh.

[Emms and Leiss, forthcoming] Martin Emms and Hans Leiss. Cut Elimination for Polymorphic Lambek Calculus. CIS Technical Report, forthcoming.

[Gabbay, 1974] Dov Gabbay. Semantical Investigations in Heyting's Intuitionistic Logic Dordrecht: Reidel.

[Girard, 1972] J. Y. Girard. Interpretation Fonctionnelle et Elimination des Coupres de L'Arithmetique d'Order Superieur. PhD Thesis.

[Hendriks, 1989] Herman Hendriks. Cut Elimination and Semantics in Lambek Calculus Manuscript available from University of Amsterdam. To appear in his PhD thesis 'Studied Flexibility'.

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

[Mey, 1992] Daniel Mey. Investigations on a Calculus Without Contractions. PhD Thesis, Swiss Federal Institute of Technology, Zurich.

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

[Moortgat, 1989] Michael Moortgat. Unambiguous proof representations for the Lambek Calculus. In *Proceedings of the Seventh Amsterdam Colloquium*, pages 389–401, Torenvliet, M. S. L. (ed.), Institute for Language, Logic and Information, Amsterdam, December 1989.

[Reynolds, 1974] J.C Reynolds. Towards a theory of type structure. In *Colloquium sur la programmation*, 1974, pages 408–423.