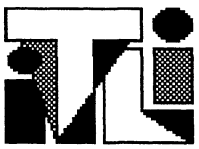


Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

LAMBEK GRAMMAR: AN INFORMATION-BASED CATEGORIAL GRAMMAR

Víctor Sánchez Valencia
Department of Mathematics and Computer Science
University of Amsterdam

ITLI Prepublications
for Logic, Semantics and Philosophy of Language
ISSN 0924-2082

Received January 1992

LAMBEK GRAMMAR: AN INFORMATION-BASED CATEGORIAL GRAMMAR

Víctor Sánchez

1. INTRODUCTION

1.1. PRELIMINARY REMARKS. In this report I would like to present and partially illustrate Lambek Grammar (**LG**). LG is a non-directed categorial grammar in which string generation and type combination are intimately connected in that, as we shall see, it is one process which combines types and generates strings. This grammar is specifically designed for the regulated transmission of *grammatical* information between semantical types and strings. So, Lambek Grammar contains principles guiding the transmission of information from the semantical types into the strings. For reasons we shall come to, Lambek Grammar also contains principles regulating the flow of information from the strings into the semantic types themselves.[1]

We have an intrinsic motivation for setting up LG because we think that this system is a convenient vehicle for the representation of temporal meaning. For ease and clarity of exposition this feature of Lambek Grammar will be considered in other place. In the hope of providing an initial orientation of the advantages of LG we offer in this report an example of its practical utility: we use Lambek Grammar to overcome the linguistic infelicity of the non-directed Lambek Calculus (**LP**) developed in van Benthem (1986).

1.2. LP AND OVERGENERATION. To put this report into perspective, let us look briefly at the sources of infelicity in LP. Students familiar with contemporary categorial literature know that this formalism is often seen as syntactically disturbing and semantically embarrassing.

1.2.1. SYNTACTICAL INADEQUACY. In LP an expression of category (a, b) and an expression of category a are allowed to combine in either order. As a result, if a string of English words is assigned to a category a, then any permutation of the members of the string also counts as member of the same category a. For instance, LP assigns the following strings to the category of sentences **loves heloise abelard*, ** loves abelard heloise*, **abelard heloise loves*, **heloise abelard loves*, *abelard loves heloise* and *heloise loves abelard.*

1.2.2. SEMANTICAL INADEQUACY. In LP the sentence *abelard loves heloise* is given the two non-equivalent readings:

(a) Love(abelard, heloise) and
(b) Love(heloise, abelard).

The sentence *every man admires a woman* is given the following non-equivalent four readings:

(a) $\exists y$ (woman(y) $\wedge$ $\forall x$ (man(x) $\rightarrow$ love(y, x))),
(b) $\exists y$ (woman(y) $\wedge$ $\forall x$ (man(x) $\rightarrow$ love(x, y))),
(c) $\forall x$ (man(x) $\rightarrow$ $\exists y$ (woman(y) $\wedge$ love(y, x))) and
(d) $\forall x$ (man(x) $\rightarrow$ $\exists y$ (woman(y) $\wedge$ love(x, y))).

As these examples indicate, LP is insensitive to external 'surface constituent order' and to internal 'semantical argument order'. In our presentation of LG we shall focus on the problem posed by the semantical inadequacy of LP but we shall also touch on the issue of syntactic overgeneration.

1. 3. CLAIM. We claim that LG gives one reading to the sentence *abelard loves heloise* and two readings to the sentence *every man admires a woman.*[2] In addition, we shall see that LG rules out such garbage as **abelard heloise loves, *heloise abelard loves.*

1.4. A BROAD DESCRIPTION OF LG. A central feature of the current theory of categorial grammar, carried over into LG, is that this linguistic formalism resembles a logical implication calculus. The elimination (Modus Ponens, application) and introduction (Conditionalization, withdrawal) rules for the implication define the ways in which types can be 'combined' and specify the type resulting from this 'combination'. A key feature of LG is that herein the rules for the implication have a dual character. From the logical point of view they can be seen as the introduction and elimination rules for the implication. From the linguistic point of view these rules can be seen as the string operations of deletion and concatenation.

1.5. ASSIGNMENT STATEMENTS. The format of LG is determined by our decision to take *statements* expressing the association of strings of symbols with semantical types as the items the grammar provides a *proof* of. Consequent upon this decision the formulas of LG are assignment statements of the form $\Sigma \in a$, where Σ is a string of symbols (the subject of the statement) and a is a category (the predicate of the statement).[3]

It will be observed at the outset that the symbols making up the subject of assignment statements are usually words of English but they can also be abstract linguistic items, i.e. items which do not appear in actual natural language expressions: variables, temporal

operators, case slots and tense slots. We shall conclude this section with a brief sketch of the role these abstract elements play in LG.

1.6. ABSTRACT ELEMENTS. In general, the role of the variables consists in permitting the postponement of the concatenation of a lexical item. By so doing, items which appear 'deep' in a string may have a wider scope than their position in the string might lead us to expect. For instance, a Noun Phrase (NP) in object position may be given a semantical scope over the whole sentence. This is achieved by processing a variable on the place the NP would have been processed had it not been postponed.

The slots are the places in strings in which information, stemming from the semantical types, can be stored. Typically, the NP combined with a transitive verb to form a Verb Phrase (VP) will carry a slot filled with the 'accusative' marker. Thus, in the current framework the type of transitive verbs carries information about grammatical relations.

In the linguistic literature the nominative marking is sometimes linked to tensed verbs, sometimes to temporal items -tense 'operators' or auxiliaries. In LG we choose the second alternative. Thus, an NP occurring in the *scope* of a tense operator will carry a slot filled with the 'nominative' marker. This means that the type of the tense operator will carry information about grammatical relations.

1.7. CLAIM. We shall show that the mechanism of information flow from strings into the semantical types allows LG to prove that tensed verbs are also nominative assigners.

1.8. FILTERS ON LG DERIVATIONS. On the top of the mechanism which generates strings with case and tense slots, we impose a double filter to eliminate spurious derivations. These filters take the form of linear precedence constraints and are similar in *spirit* to the Linear Precedence statements of GPSG -another linguistic formalism in which linear ordering is factored out from the 'concatenation' rules.[4] The filters hinge on information about grammatical relations carried by strings. We propose to determine the constituent and argument order of finite sentences by using the case information encoded in the NPs:

- a string assigned to the type of sentences is acceptable only if therein
  (a) nominative NPs precede tensed verbs and
  (b) transitive verbs precede accusative NPs.

So, ultimately, we shall fix order by using outputs of the case marking mechanism.

It is worthwhile to pause a moment and reflect on the role of these filters in the current theory. LP can be seen as a logical calculus providing an independent notion of semantical interpretability. It answers the question: which sequences of expressions are semantical in-

terpretable? However, LP is not intended to match a language-specific notion of syntactical correctness. Moreover, according to LP sentences of the form Noun Phrase + Transitive Verb + Noun Phrase are ambiguous as to whether which Noun Phrase is the object of the Transitive Verb. Obviously, the latitude of LP has to be restricted in some way. As van Benthem (1986) points out, there are at least two standard strategies. One is to add rule-specific constraints to the system so that strings are generated or recognized in accordance with principles which reflect language-specific facts of syntax and semantics. This strategy is followed in Sanchez (1991a). A result of this strategy is a weaker logical calculus with more complex inference rules. Another strategy is to keep the calculus as it is, adding filters to account for language-specific syntactical and semantical facts. These facts are language dependent and so are the filters. Thus, the filter strategy has the advantage that it leaves the logical part of the grammar undisturbed. For instance, the LP rules for English and Spanish are exactly the same. The differences between these languages could then be handled by language-specific filters.

1.9. SUMMARY. To sum up, LG is a non-directed categorial grammar in which information encoded in semantical types is allowed to flow into strings. Moreover, information encoded in strings is allowed to flow back into semantical types. Linguistic infelicity is prevented by using output constraints based upon grammatical information encoded in strings. The immediate goal of LG is to overcome the semantical inadequacy of unrestricted non-directed categorial grammars. However, a more global concern will be to underscore the utility of an information-based categorial grammar.

1. 10. OVERVIEW. In the following section I would like to make some general remarks about the main properties of LP. Everything I will mention there is discussed in van Benthem's categorial writings.[5] My aim is primarily to pave the way for LG. The third section is devoted to a formal presentation of the basic part of LG -which we call unrestricted LG. We shall argue that unrestricted LG is a notational variant of LP. The fourth section contains a preliminary discussion of the way in which LG can be constrained to overcome LP 's predicament. I hope that by this point it would have been made acceptable that LG proves to be fruitful. The fifth section contains a more complete presentation of the way in which LG copes with LP's problems. The sixth section considers a possible generalization of the strategy put forward in the foregoing sections. By this point, I hope that LG has proven to be promising for further study.

2. DESCRIPTION OF LP

2.1. SEMANTICAL TYPES. The basic idea in categorial grammar is to link grammatical categories with semantical types. The following correspondence is the most relevant for us here:

category	type
proper name	e
intransitive verb	(e, t)
common noun	(e, t)
transitive verb	(e, (e, t))
determiner	(e, t), ((e, t), t)

Semantical types are obtained by assuming that there is a set of atomic types and that

- every atomic type is a type.
- if a and b are types, then so is (a, b).

Each atomic type is intended to denote a particular set. For instance, the atomic types e, t are intended to denote, respectively, a set of entities and a set of truth-values. Each complex type (a, b) is intended to denote some function from the set denoted by a to the set denoted by b.

2.2. LP IN THE CATEGORIAL LANDSCAPE. Categorial grammars, seen as implication calculi, differ from each other according to the answers given to the following questions:

(a) should we keep track either of formulas or occurrences of formulas in derivations?

(b) in the definition of the sequent $\Delta \Rightarrow \phi$, should we say that Δ forms a set, a multiset or a list?

(c) in the definition of the sequent $\Delta \Rightarrow \phi$, should we say that all the members of Δ must be used in the derivation proving it, or should we say that the used premisses must be contained in Δ?

As it happens, LP is a natural deduction system in which we keep track of occurrences and in which the relation of entailment solely holds between the multiset of open assump-

tions and the conclusion of a derivation. More formally, van Benthem defines LP as follows:

- A sequent $X \Rightarrow b$ is LP-derivable if there exists a proof tree for b in which exactly the occurrence of the premisses mentioned in the sequent X remain in force, such that each Conditionalization has withdrawn exactly one occurrence of its antecedent.

2.2.1. EXAMPLES OF LP DERIVATIONS. Typical LP derivations are the following two:

(1) e ; e, (e, t) ; $e \Rightarrow t$ is LP-derivable:

$$\dfrac{e \qquad \dfrac{e,\ (e,t) \qquad e}{(e,t)}\ \text{elimination}}{t}\ \text{elimination}$$

(2) (e,t), t ; (e, (e, t) ; (e,t), $t \Rightarrow t$ is LP-derivable :

$$\dfrac{\dfrac{(e,t),\ t \qquad \dfrac{e,\ (e,t) \qquad e^1}{(e,t)}\ \text{elimination}}{t}\ \text{elimination}}{(e,t)}\ \text{introduction -1}^1 \qquad (e,t),\ t\ \text{elimination}$$

$$t$$

I will not give more examples of LP-derivations here, as we shall see many more 'similar' LG-derivations below.

2.3. TYPED LAMBDA TERMS AND LP DERIVATIONS.

2.3.1. THE TYPED LANGUAGE. Van Benthem (1986) shows that there is an effective correspondence between derivations in LP and terms in a logical type-theoretical language. This language has an infinite supply of variables $x_a, y_n, \ldots$ for each type a. The formation rules are

- if t_1 is of type (a, b), and t_2 is of type a, then $t_1(t_2)$ is a term of type b. (application)
- if t is of type b and x is a variable of type a, then $(\lambda x.\ t)$ is a term of type (a, b). (lambda abstraction)

The broad idea, due to Curry, behind the correspondence between derivations and terms is that the premisses of a derivation, i.e. types, correspond to distinct variables of those types, that Modus Pones corresponds to application and, finally, that Conditionalization corresponds to abstraction.

2.3.2. A FRAGMENT OF THE LAMBDA CALCULUS. Essential to van Benthem result is the definition of a *fragment* of the Lambda Calculus which corresponds to LP derivations. He defines a class Λ of typed terms as follows:

- Individual variables are in Λ.
- If t_1, t_2 are in this class, they have no terms in common and $t_1(t_2)$ is a term, then $t_1(t_2)$ is in this class as well.
- If t is in Λ, x occurs exactly once free in t and $\lambda x.t$ has at least one free variable, then $\lambda x.t$ is also in Λ.

2.3.3. VAN BENTHEM'S RESULT. Consequently, Van Benthem proves

- $a_1 \cdots a_n \Rightarrow a$ is provable in LP iff there exists a term in Λ with exactly the free variables $x_{a_1} \cdots x_{a_n}$.

The proof itself provides a method for obtaining the Λ-term associated with the derivation which proves $a_1 \cdots a_n \Rightarrow a$. It is also effective in the other direction: each term in Λ encodes an LP-derivation.

2.3.4. EXAMPLES OF THE CORRESPONDENCE. Below we show the terms corresponding to the derivations in LP given above. These examples illustrate the way in which the correspondence takes place.

- Derivation (1) corresponds to the term $x_{(e,(e,t))}\, y_e(z_e)$:

$$\dfrac{\dfrac{x_{(e,(e,t))} \qquad y_e}{xy} \qquad z_e}{xy(z)}$$

- Derivation (2) corresponds to the term $v_{(e,t),\,t}\,(\lambda y_e \cdot z_{(e,t),\,t}(x_{(e,\,(e,\,t))}\,y))$:

$$\dfrac{z_{(e,t),\,t} \qquad \dfrac{\dfrac{x_{(e,\,(e,\,t))} \quad y_e}{xy}}{z(xy)}}{\dfrac{\lambda y \,.\cdot z(xy) \qquad\qquad v_{(e,\,t),\,t}}{v(\lambda y \,.\, z(xy))}}$$

2.4. LP AND LANGUAGE RECOGNITION. Classical Categorial Grammar is a language recognition device. LP may be used with this purpose in mind in the following way:

- a string Σ of English words is assigned to the type a if some corresponding sequence of initially assigned types derives a in LP.

2.4.1. EXAMPLES OF LP AS RECOGNITION DEVICE.

- Example (1) shows that the string *abelard loves heloise* can be assigned to the type t by assigning *abelard* to type e; *loves* to type e, (e, t) and *heloise* to type e.

- Example (2) shows that the string *every man loves a woman* can be assigned to the type t by assigning *every man* to type (e, t), t ; *loves* to type e, (e, t) and *a woman* to type (e, t), t.

2.4.2. REMARK. It should be said that when we are interested in an LP-derivation as an instrument for language recognition we usually present the derivation with the words inserted above the relevant assumptions. For instance (1), used to recognize *abelard loves heloise*, is usually represented as follows:

$$\dfrac{\text{abelard} \qquad \dfrac{\text{loves} \quad \text{heloise}}{\text{e, (e, t)} \qquad \text{e}}\,\text{elimination}}{\text{e} \qquad\qquad \text{(e, t)}}\,\text{elimination}$$

$$\text{t}$$

2.5. THE INADEQUACY OF LP. The above description of LP as a device for linguistic recognition shows why this formalism counts as a crude linguistic device. The point is that in 2.4.1 nothing is said about the structure of the derivations that establish the validity of a sequent.

Notice that according to the characterization in 2.4.1 the derivation of example (1) can also be used to assign to the type t the strings mentioned in the introduction: **loves heloise abelard, * loves abelard heloise, *abelard heloise loves, *heloise abelard loves, abelard loves heloise, heloise loves abelard.*

2.6. MEANING RECIPES IN LP. Before turning to the question of semantical overgeneration, I would like to introduce within the framework just sketched a modification to van Benthem's association of LP derivations with Lambda terms.[6] First we define as follows the notion of *meaning recipe* :

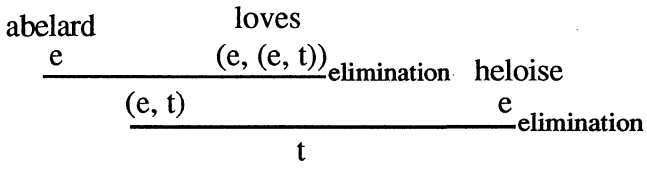
- Let $A_1, \ldots, A_n$ be a string of natural language expressions. Assume that each A_i has been assigned to a semantical type a_i. We shall say that if D is an LP derivation that proves $a_{i_1}, \ldots, a_{i_n} \Rightarrow a$, and S is the Λ-term that corresponds to D, then S is a *meaning recipe* of $A_1, \ldots, A_n$ (under D).

Van Benthem's association of LP derivations with Lambda terms requires the choice of different variables for the assumptions used in the derivation. I wish to follow a slightly different course here. We shall assume that the typed language contains the expressions $Aa_i, \ldots, Aa_n$ as primitive terms. In the construction of a meaning recipe we shall take different variables for assumptions *withdrawn* by Conditionalization. However, we shall use $Aa_i, \ldots, Aa_n$ for the assumptions which are in force after concluding the derivation.[7] By and large, this is the usual practice in the LP literature. When we are interested in a term that codifies a derivation used for recognition purposes we use words instead of the free variables of the term -these variables correspond to the open assumptions of the derivation. Under this modification

- (1) corresponds to the term $\text{loves}_{(e,\,(e,\,t))}\ \text{heloise}_e(\text{abelard}_e)$. And
- (2) corresponds to the term
  $\text{every man}_{(e,t),\,t}\ (\lambda y_e \cdot \text{a woman}_{(e,t),\,t}(\text{loves}_{(e,\,(e,\,t))}\ \ y))$.

2.7. SEMANTICAL INADEQUACY OF LP. Let us now go back to the question of semantical overgeneration. Consider the following LP derivation and the corresponding meaning recipe:

(3)



Meaning recipe: $\text{loves}_{(e,\,(e,\,t))}\ \text{abelard}_e(\text{heloise}_e)$.

If we accept that (3) proves that *abelard loves heloise* is a sentence then we have to conclude that this sentence has two meaning recipes. To have two recipes is a necessary condition for having two readings. But this is not sufficient. We can speak of a string having more than one reading only when we are speaking about the denotation of the associated terms. Readings are induced by the meaning recipes but these recipes are not yet the 'meaning' of the natural language expressions. There are two ways to derive two readings from the recipes associated with our string. One is based on the logical properties of the terms involved: $\text{loves}_{(e,\,(e,\,t))}\ \text{abelard}_e(\text{heloise}_e)$ and $\text{loves}_{(e,\,(e,\,t))}\ \text{heloise}_e(\text{abelard}_e)$ are not logically equivalent. Hence, it is possible to find an interpretation of the logical typed language by which the denotations of these terms differ from each other. And this is all we need to assert that, according to LP, the string *abelard loves heloise* has two readings.

The other way is to make use of the semantical machinery associated with the type language in combination with 'meaning' postulates to ensure that our two Lambda terms denote different objects. I will now try to describe this second approach since it is interesting in its own right.

2.8. THE DENOTATION OF EXPRESSIONS. The denotation of natural language expressions can be computed by using the meaning recipe which LG determines. To obtain this denotation one defines first the hierarchy of domains:

- D_e is a non-empty set.
- D_t is the set of truth-values $\{0,1\}$.
- $D_{(a,\,b)}$ is the set of functions from D_a into D_b.

One then establishes that if $t \in a$, then the denotation of t is a member of D_a.

2.8.1. EXAMPLES OF DENOTATIONS. In the following discussion we shall use λ-'s and first order logic to speak of functions. Notice that according to our initial assignment statements:

(a) the denotation of *love* will be the function (described by) $\lambda y \cdot \lambda x \cdot$ love(x, y) where x, y are objects of type e.

(b) the denotation of *man*, *woman* and *walks* will be defined by the functions $\lambda x \cdot$ man(x), $\lambda x \cdot$ woman(x), $\lambda x \cdot$ walks(x), etc where x is an object of type e.

(c) the denotation of the determiners *a* and *every* will be $\lambda P \cdot \lambda Q \cdot (\forall x$ (P(x) $\rightarrow$ Q(x)), $\lambda P \cdot \lambda Q \cdot (\exists x$ (P(x) $\wedge$ Q(x), respectively, where P and Q are objects of type (e, t).

Moreover,

(d) the denotation of a compound expression A(B) will be denoted by the result of applying the function denoted by A to the object denoted by B.

Finally,

(e) the denotation of an expression $\lambda x \cdot$ t will be defined as $\lambda x \cdot$ t', where t' is the object denoted by t.

2.8.2. ILLUSTRATION. The denotation of *abelard loves heloise* induced by the two LP derivations and the previous conventions are:

- $(\lambda y \cdot \lambda x \cdot$ love(x, y)(heloise))(abelard) =
  $\lambda x \cdot$ love(x, heloise)(abelard) =
  love(abelard, heloise)

- $(\lambda y \cdot \lambda x \cdot$ love(x, y)(abelard))(heloise) =
  $\lambda x \cdot$ love(x, abelard)(heloise) =
  love(heloise, abelard)

We assume that the reader can work out that the denotation of *every man loves a woman* under derivation (2) will be:

- $\exists y$ (woman(y) $\wedge \forall x$ (man(x) $\rightarrow$ loves(y, x))).

We have now described some of the properties of LP and we have sketched the difficulties and limitations of this formalism. The next section is devoted to an initial description of LG -the formalism with which we aim to overcome these difficulties. We start below by describing LG as a notational variant of LP. This basic version of LG will be called 'unconstrained LG'.

3. TOWARDS A DESCRIPTION OF LG

3.1. BASIC ASSIGNMENTS. In LG natural language expressions get a semantic type via the assignment statements:

$$A \in a$$

where A is a basic natural language expression and a is a category. It will necessary to assume the existence of an infinite supply of variables $X_1, \ldots, X_n, \ldots$ which might be used as (part of) subjects of assignment statements.

3.1.1. EXAMPLES OF BASIC ASSIGNMENTS.

- abelard, heloise $\in$ e.
- walks $\in$ (e, t).
- man, woman $\in$ (e, t).
- loves $\in$ e, (e, t).
- every, some $\in$ (e, t), ((e, t), t).

Strings of natural language expressions and variables get a type via the following rules:

3.1.2. ELIMINATION RULES (MODUS PONENS).

$$\frac{A \in a \quad B \in (a, b)}{AB \in b}\,R1 \qquad \frac{B \in (a, b) \quad A \in a}{BA \in b}\,R2$$

3.1.3. INTRODUCTION RULE (CONDITIONALIZATION).

$$\begin{array}{c} X \in a \\ \triangledown \\ F(X) \in b \end{array}$$

$$\frac{F(X) \in b}{F \in (a, b)}\,R3$$

where F(X) is a string containing exactly one occurrence of the variable X and F is the result of deleting X from F(X).

After an application of R3, the assignment statement $X \in a$ is called 'discharged'.

3.2. DEFINITION OF LG ANALYSES. An LG derivation D is an analysis of the string $A_1 \ldots A_n$ if

- $a_{j1}, \ldots, a_{jn} \Rightarrow b$ has an LP proof.
- No A_j is a variable.
- D proves $A_1 \in a_{j1}, \ldots, A_n \in a_{jn} \Rightarrow A_1 \ldots A_n \in b$.

3.2.1. EXAMPLES OF ANALYSES. The following derivations indicate that LG has LP's potential of semantical overgeneration.

(4) abelard loves heloise ∈ t.

$$\dfrac{\text{abelard} \in e \quad \dfrac{\text{loves} \in e, (e, t) \quad \text{heloise} \in e}{\text{loves heloise} \in (e, t)}}{\text{abelard loves heloise} \in t}$$

(5) abelard loves heloise ∈ t.

$$\dfrac{\dfrac{\text{abelard} \in e \quad \text{loves} \in (e, (e, t))}{\text{abelard loves} \in (e, t)} \quad \text{heloise} \in e}{\text{abelard loves heloise} \in t}$$

(6) every man loves a woman ∈ t.

$$\dfrac{\text{every man} \in (e, t), t \quad \dfrac{\dfrac{\dfrac{Y \in e \quad \text{loves} \in e, (e, t)}{Y \text{ loves} \in (e, t)} \quad X \in e}{Y \text{ loves } X \in t}}{\text{loves } X \in (e, t)}}{\dfrac{\text{every man loves } X \in t}{\text{every man loves} \in (e, t)} \quad \text{a woman} \in (e, t), t}$$

$$\text{every man loves a woman} \in t$$

(7) every man loves a woman ∈ t.

$$\dfrac{\text{every man} \in (e, t), t \qquad \dfrac{\dfrac{\text{loves} \in e, (e, t) \qquad X \in e}{\text{loves } X \in (e, t)}}{\text{every man loves } X \in t}}{\dfrac{\text{every man loves} \in (e, t) \qquad \text{a woman} \in (e, t), t}{\text{every man loves a woman} \in t}}$$

(8) every man loves a woman ∈ t.

$$\dfrac{\dfrac{X \in e \qquad \text{loves} \in e, (e, t)}{X \text{ loves} \in (e, t)} \qquad \dfrac{\text{a woman} \in (e, t), t}{X \text{ loves a woman} \in t}}{\dfrac{\text{every man} \in (e, t), t \qquad \text{loves a woman} \in (e, t)}{\text{every man loves a woman} \in t}}$$

(9) every man loves a woman ∈ t.

$$\dfrac{\dfrac{Y \in e \qquad \dfrac{\dfrac{\text{loves} \in (e, (e, t)) \qquad X \in e}{\text{loves } X \in (e, t)}}{Y \text{ loves } X \in t}}{Y \text{ loves} \in (e, t)} \qquad \dfrac{\text{a woman} \in (e, t), t}{Y \text{ loves a woman} \in t}}{\dfrac{\text{every man} \in (e, t), t \qquad \text{loves a woman} \in (e, t)}{\text{every man loves a woman} \in t}}$$

Unconstrained LG being a notational variant of LP takes over some of its virtues and most of its defects. In the rest of this section we shall make this point clear.

3.3. LG AND LAMBDA TERMS. In the first place, notice that the correspondence between LG derivations and Lambda terms can be extended to LG. One simply links each assumption $A \in a$ in an LG derivation to a typed term A_a and proceeds further as in van Benthem's proof treating the items A_a as variables of the typed logical language. For instance, the above examples induce the following terms:

- $love_{e, (e, t)}$ $heloise_e$ $abelard_e$:

$$\dfrac{\dfrac{love_{e, (e, t)} \quad heloise_{\ e}}{love\ heloise} \qquad abelard_e}{love\ heloise\ abelard}$$

Denotation: love(abelard, heloise)
Cf. (4).

- $love_{e, (e, t)}$ $abelard_e$ $heloise_e$:

$$\dfrac{\dfrac{love_{\ e, (e, t)} \quad abelard_e}{love\ abelard} \qquad heloise_e}{loves\ abelard\ heloise}$$

Denotation: love(heloise, abelard)
Cf. (5).

- $a\ woman_{(e, t), t}$ ($\lambda x_e \cdot every\ man_{(e, t), t}$ ($\lambda y_e \cdot$ $love_{e, (e, t)}$ y x)) :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{y_e \quad love_{e, (e, t)}}{love\ y} \qquad x_e}{love\ y\ x}}{\lambda y \cdot love\ y\ x}}{every\ man\ (\lambda y \cdot love\ y\ x)}}{\lambda x \cdot every\ man\ (\lambda y \cdot love\ y\ x) \qquad a\ woman_{(e, t), t}}$$
$$a\ woman\ (\lambda x \cdot every\ man\ (\lambda y \cdot love\ y\ x))$$

Denotation: $\exists y$ (woman(y) $\wedge$ $\forall x$ (man(x) $\rightarrow$ love(y, x))).
Cf. (6).

- $a\ woman_{(e, t), t}$ (λx_e . $every\ man_{(e, t), t}$ ($love_{e, (e, t)}$ x)) :

$$\dfrac{\dfrac{\dfrac{love_{\ e, (e, t)} \quad x_e}{love\ x}}{every\ man_{(e, t), t}}}{every\ man\ (love\ x)}$$
$$\dfrac{a\ woman_{(e, t), t} \qquad \lambda x \cdot every\ man\ (love\ x)}{a\ woman(\lambda x \cdot every\ man\ (love\ x))}$$

Denotation: $\exists y\,(\text{woman}(y) \wedge \forall x\,(\text{man}(x) \rightarrow \text{love}(x, y)))$.
Cf. (7).

- every man $_{(e, t), t}$ (λx_e . a woman $_{(e, t), t}$ (love $_{e, (e, t)}$ x)) :

$$\dfrac{\text{every man}_{(e, t), t} \qquad \dfrac{\dfrac{x_e \qquad \text{love}_{e, (e, t)}}{\text{love x}} \qquad \text{a woman}_{(e, t), t}}{\text{a woman love x}}}{\text{every man } (\lambda x \cdot (\text{a woman love x}))} \quad \lambda x \cdot (\text{a woman love x})$$

Denotation: $\forall x\,(\text{man}(x) \rightarrow \exists y\,(\text{woman}(y) \wedge \text{love}(y, x)))$.
Cf. (8).

- every man$_{(e, t), t}$ ($\lambda y_e \cdot$ a woman$_{(e, t), t}$ ($\lambda x_e \cdot$ love$_{e, (e, t)}$ x y)) :

$$\dfrac{\text{every man}_{(e, t), t} \qquad \dfrac{\dfrac{y_e \qquad \dfrac{\text{love}_{(e, (e, t))} \qquad x_e}{\text{love x} \in (e, t)}}{\text{love x y}} \quad \lambda x \cdot (\text{love x y})}{\text{a woman}(\lambda x \cdot (\text{love x y}))} \qquad \text{a woman}_{(e, t), t}}{\text{every man}(\lambda y \cdot (\text{a woman}(\lambda x \cdot (\text{love x y}))))} \quad \lambda y \cdot (\text{a woman}(\lambda x \cdot (\text{love x y})))$$

Denotation: $\forall x\,(\text{man}(x) \rightarrow \exists y\,(\text{woman}(y) \wedge \text{love}(x, y)))$.
Cf. (9).

3.4. LG AND SEMANTICAL INADEQUACY. Notice that unconstrained LG generates the same amount of readings as LP for sentences of the form NP TV NP. For instance the above examples show that LG generates two non-equivalent derivations proving that *abelard loves heloise* is a sentence. These derivations induce two readings of our sentence. Also observe that derivations (6)-(9) induce the four readings of the sentence *every man loves a woman* that we listed in 1.2.1.

This comment concludes our outline of unrestricted Lambek Grammar. So far,we have seen that LG is a notational variant of LP. The remainder of this report is devoted to a discussion of the way in which constrained Lambek Grammar differs from LP. To begin with, in the next section we describe our proposal to account for dispensing with (5) as a derivation of *abelard loves heloise*. After that, we shall describe the way in which the pro-

posal has to be modified to account for the rejection of (6) and (7) as derivations of *every man loves a woman.*

4. TOWARDS A CONSTRAINED DEFINITION OF LG

4. 1. TRANSITIVE VERBS AS CASE ASSIGNERS. As is well known, transitive verbs can be seen as case assigners. The NP that is combined with a transitive verb counts as the object of the verb and it is assigned the accusative case. The idea we want to exploit here is that the category assigned to the transitive verbs carries information about the case their object gets. For instance the initial assignment concerning the verb *love* could have the following form:

- loves $\in (e_{accu}, (e, t))$

Since we want to consider the transitive verbs as binders of accusative markers, we shall generate them as sharing an index (called *the transitive index*) with the case marker. Therefore, the *official* initial assignment concerning transitive verbs is the following:

- loves[i] $\in (e_{accu_i}, (e, t))$

At this stage we have the first point of difference between LP and constrained LG. Under the standard approach the semantical types encode only semantical information. Here the semantical type associated with transitive verbs encodes information referring to a grammatical relation.

4.1.2. INFORMATION FLOW. The transmission of case information between semantical types and strings rests on the idea that the combination of a transitive verb with a NP permits this information to flow from the semantical type into the NP. This process of information flow is regulated by the following

4.1.3. CASE RULES.

$$\frac{TV[i] \in (e_{acc_i}, (e, t)) \quad N \in e}{TV[i]\ N_{[acc_i]} \in (e, t)} \qquad \frac{N \in e \quad TV[i] \in (e_{acc_i}, (e, t)}{N_{[acc_i]}\ TV[i] \in (e, t)}$$

At the top of this mechanism of information flow we impose a linear precedence constraint filtering out undesirable derivations:

4.1.4. FILTER 1. An LG derivation D proving that A ∈ t is unacceptable if

• A is such that therein the case marker acc_i precedes the transitive index i.

4.1.5. THE READINGS OF *ABELARD LOVES HELOISE*. To make less hypothetical the question about the semantic inadequacy of restricted LG, let me consider the two derivations LG gives of *abelard loves heloise* ∈ t. There remains a number of things I have to go into in detail, but we have enough apparatus to exclude one of the above derivations. A fortiori, we exclude one of the readings we noticed before.

(10)

$$\dfrac{\text{abelard} \in e \qquad \dfrac{\text{loves[i]} \in (e_{acc_i}, (e, t)) \qquad \text{heloise} \in e}{\text{loves[i] heloise}_{[acc_i]} \in (e, t)}}{\text{abelard loves[i] heloise}_{[acc_i]} \in t}$$

Cf. derivation (4).

(11)

$$\dfrac{\dfrac{\text{abelard} \in e \qquad \text{loves[i]} \in (e_{acc_i}, (e, t))}{\text{abelard}_{[acc_i]} \text{ loves[i]} \in (e, t)} \qquad \text{heloise} \in e}{\text{abelard}_{[acc_i]} \text{ loves[i] heloise} \in t}$$

Cf. derivation (5).

Notice that our marking mechanism and the linear precedence constraint excludes (11) as an admissible derivation, since acc_i precedes i in the terminal string. By contrast, (10) satisfies the linear precedence constraint. Thus, as we claimed in the introduction LG supplemented with a case filter reduces to one the number of derivations that LP associates with *abelard loves heloise*. A fortiori, LG reduces the number of reading of this sentence.

4.1.6. SOME PERMUTATIONS OF *ABELARD LOVES HELOISE*. There is, however, something more. This simple mechanism has some impact on the syntactic inadequacy of LG since we cannot accept as admissible a proof of **abelard*$_{[acc_i]}$ *heloise loves [i]* and **heloise abelard*$_{[acc_i]}$ *loves[i]*. This reduces the number of inadmissible strings mentioned in 1.2.1.

On the basis of these observations we can conclude that by taking seriously the mechanism of information flow along categorial derivations we have the beginning of a linguistic felicitous non-directed categorial grammar.

4.2. NOMINATIVE MARKING. One of the things I have not gone into in detail yet, concerns the expression *abelard* in the non rejected string *abelard loves[i] heloise*$_{[acc_i]}$.[8] In the linguistic literature any sentence containing an NP unmarked for case is considered ill-formed. Therefore we have to explain how LG assigns case to NPs that do not combine directly with transitive verbs. To do this we need to take a brief look at the way in which LG handles tense.

In our treatment of tense we follow the practice of tense logicians by interpreting temporal items as operators. However, we think that the *syntactical* effect of such operators in natural language resembles as much the work of binding operators as it resembles the work of unary Boolean ones.[9] In a way to be explained below, LG generates strings containing base of verbs carrying an empty temporal slot. We exploit the analogy between temporal items and binders by asking that such an item must bind as least one temporal slot. For instance, LG will generate the tenseless matrix *abelard love[i] heloise*$_{[acc_i]}$. Then we shall apply the present operator to this string obtaining in this way *Present abelard love[present, i] heloise*$_{[acc_i]}$.

It is at this point in which our treatment of tense can be used to limit the syntactic overgeneration of LG. It is generally agreed that in English the subject of tensed clauses uniformly appears in nominative case. In our framework this will be accounted for by extending the binding properties of the tense operators. We shall say that tense operators fill the temporal slot of the verb base and that they assign the nominative case to unmarked NPs. In addition, we shall say that the temporal operator binds together the tense slot and the case marker by using an index shared by the markers *present* and *nom.*

4.2.1. TENSE OPERATORS AS CASE ASSIGNERS. Let me turn to the specification of the manner in which LG implements these ideas. In the first place the basic assignment for the tense operator and the base of verbs take now the following form:

- Present $\in$ (t_{nom_j}, $pres_j$, t)

We assume that *Present* denotes the identity function in $D_{(t,t)}$.

- love[i] $\in$ (e_{accu_i}, (e, t))
- walk[] $\in$ (e, t)

Once again, the innovation consists in encoding extra information in the semantical types and in the symbols from which the subject of assignment statements is made up.

4.2.3. INFORMATION FLOW. The propagation of information from the semantical type into the strings is regulated by the:

4.2.4. TENSE INTRODUCTION RULE

$$\frac{\text{Present} \in (t_{\text{nom}_j,\ \text{pres}_j},\ t) \quad F(N, V[*]) \in t}{\text{Present}\ F(N_{[\text{nom}_j]},\ V[\text{pres}_j,\ *]) \in t}$$

where $F(N, V[*])$ is any string containing the items N, $V[*]$ in any order. Moreover, N must be the subject of a statement of the form $N \in e$. Finally, $*$ is the empty string or a transitive index.

4.2.5. TENSE DELETION RULE. The following rule will guarantee that the temporal operator is discharged correctly -i.e. after it has filled the tense and the nominal slot:

$$\frac{\text{Present}\ F(Y_{[\text{nom}_j]}\ V[\text{pres}_j]) \in t}{F(Y_{[\text{nom}_j]}\ V_{\text{pres}_j}) \in t}$$

Now we can add the following new linear condition in order to rule out undesirable derivations:

4.2.6. FILTER 2. An LG derivation D proving that $A \in t$ is unacceptable if

• A is such that the index *$pres_j$* precedes the index *nom_j*.

4.2.7. EXAMPLE OF NOMINATIVE MARKING

(12)

$$\dfrac{\dfrac{\text{Present} \in (t_{\text{nom}_j,\ \text{pres}_j},\ t) \qquad \dfrac{\text{abelard} \in e \qquad \dfrac{\text{love[i]} \in (e_{\text{acc}_i},\ (e,\ t)) \quad \text{heloise} \in e}{\text{love[i]}\ \text{heloise}_{[\text{acc}_i]} \in (e,\ t)}}{\text{abelard love[i] heloise}_{[\text{acc}_i]} \in t}}{\text{Present abelard}_{[\text{nom}_j]}\ \text{love}[\text{pres}_j,\ i]\ \text{heloise}_{[\text{acc}_i]} \in t}}{\text{abelard}_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ \text{heloise}_{[\text{acc}_i]} \in t}$$

Cf. derivations (4) and (10).

4.2.8. THE OTHER PERMUTATIONS OF *ABELARD LOVES HELOISE*. It should be apparent that the effect of this filter goes beyond the question of semantical inadequacy since it excludes garbage such as *$\text{love}_{\text{pres}_j}$ $\text{heloise}_{[\text{nom}_j]}$ $\text{abelard}_{[\text{acc}_i]}$ and *$\text{love}_{\text{pres}_j}$ $\text{abelard}_{[\text{acc}_i]}$ $\text{heloise}_{[\text{nom}_j]}$.

4.3. THE SITUATION SO FAR. Thus far, we have envisaged LG as a grammar containing

- rules of type combination and string construction
- a mechanism of case marking
- linear precedence constraints

This grammar is arguably more felicitous than LP. Thus, as far as the combination of proper names and transitive verbs is concerned LG fares better than LP does. Observe that from the list of 1.2.1. we have been able to eliminate the four unacceptable strings. Moreover, we have been able to exclude derivation (5) as an LG derivation of the string *abelard loves heloise.*

This is, of course, not sufficient. Several points of detail that were glossed over require further consideration. First, our description does not exclude the possibility that a variable gets a case marker. But then our mechanism of information flow does not explain what happens when a marked variable is deleted. Therefore we have still to explain how the case information can flow from the strings into the semantical types.

Second, in the case and tense rules as stated above, we have described the flow of informations as passing from the 'functional' type to the string linked to the 'argument' type. But if we want to assign the accusative case to complex NPs then the rules have to be extended to cover all the NPs. This means that we must allow that the information flows from the argument type into the string linked to the functional type. In the next section I shall give a precise formulation of this process of information flow. I shall be arguing there that Lambek Grammar reduces the derivations associated with the string *every man loves a woman.*

Finally, the subject of the statements our system provides a proof of, is not English since it carries abstract elements that do not appear in surface English expressions. To put the last point in a slightly different way: as it stands LG generates structures that do not look quite English sentences. In fact, we need additional principles to regulate the deletion of these abstract elements.

In the next section we will modify our picture of LG to take these three points into account.

5. THE CONCRETE MECHANISM OF INFORMATION FLOW

5.1. GENERAL TRANSMISSION RULES. Several attempts to give a characterization of grammatical relations boil down to the assertion that the object of a sentence is the NP which combines with the rest of the string before the subject NP. Given the way in which LG (and LP) has been defined it is not possible for us to adopt this characterization. Our account of case marking is a generalization of the pilot example of the previous section. First, we use the expression *mark* to stand for the information items nom_j and acc_i. Then we define as follows the transmission of information from types to strings:

5.1.1. INFORMATION FROM FROM TYPES INTO STRINGS.

5.1.1.1 INFORMATION FROM FROM FUNCTIONAL TYPES INTO ARGUMENT STRINGS.

• TENSE

$$\frac{\text{Present} \in (t_{nom_j,\ pres},\ t) \quad F(N, V[*]) \in t}{\text{Present}\ F(N_{[nom_j]},\ V[pres_j, *]) \in t}\ T$$

where

• F(N, V[*]) is a string containing the items V[*], N in any order.
• $N \in (e, t), t$ or $N \in e$.
• * is the empty string or the transitive index.

• CASE MARKING

$$\frac{V \in (a_{mark},\ b) \quad N \in a}{VN_{[mark]} \in b}\ MR1 \qquad \frac{N \in a \quad V \in (a_{mark},\ b)}{N_{[mark]}V \in b}\ MR2$$

5.1.1.2 INFORMATION FROM FROM ARGUMENT TYPES INTO FUNCTIONAL STRINGS.

• CASE MARKING

$$\frac{V \in (e_{mark},\ t) \quad N \in (e, t), t}{VN_{[mark]} \in t}\ MR3 \qquad \frac{N \in (e, t), t \quad V \in (e_{mark},\ t)}{N_{[mark]}\,V \in t}\ MR4$$

5.1.1.3. REMARK. If we look back at the previous case marking rules we note that while the first two apply to all semantical types, the second pair applies only to Noun Phrase types. The point is the type of a verb can be the argument of a functional type which is not allowed to carry case marking, e.g. adverbs.

Next, we turn to the rules regulating transmission of marking information from strings to categories:

5.1.2. INFORMATION FROM STRINGS INTO TYPES

$$\frac{Y \in b \qquad \triangledown \qquad F(Y_{[mark]}) \in a}{F \in (b_{mark}, a)}\ MR5$$

where $F(Y_{[mark]})$ is a string containing exactly one occurrence of $Y_{[mark]}$ and F is the result of deleting $Y_{[mark]}$ from $F(Y_{[mark]})$.

Finally, we introduce the rules that delete abstract elements:

5.1.3. DELETION OF ABSTRACT ELEMENTS

• TENSE OPERATOR DELETION

$$\frac{\text{Present } F(Y_{[nom_j]}\ V[pres_j]\ ) \in t}{F(Y_{[nom_j]}\ V_{pres_j}\ ) \in t}\ D1$$

• SLOTS DELETION

$$\frac{M_{[nom_j]}\ V[pres_j]\ P_{[acc_i]} \in t}{M'\ V'\ P' \in t}\ D2$$

where

- $M_{[nom_j]}\ V[pres_j]\ P_{[acc_i]}$ contains no variables
- V' is the present tense inflexion of V[] which agrees in number with $M_{[nom_j]}$, and

- M' and P' are the nominative and accusative inflexion of M and P if they exist. Otherwise they are M and P themselves.

The two filters introduced in the previous section remain unchanged.

In the next part we illustrate the way in which these rules work. The derivations worked out there bear on claims made in the introduction.

5.2. VERBS AS NOMINATIVE ASSIGNERS. First we show that tensed verbs can also be seen as case assigners with regard to the nominative case. As we pointed out in the introduction judgments differ as to which linguistic item is the assigner of the nominative case. We have chosen for tense as assigner but we are able to show that tensed verbs are also nominative assigners.

5.2.1. INTRANSITIVE VERBS AS NOMINATIVE ASSIGNERS. Below we display the derivation proving $\text{dance}_{\text{pres}_j} \in (e_{\text{nom}_j}, t)$:

$$\dfrac{\dfrac{\dfrac{\text{Present} \in (t_{\text{nom}_j,\, \text{pres}_j}\ t) \qquad \dfrac{X \in e \quad \text{dance}[\,] \in (e, t)}{X\ \text{dance}[\,] \in t}}{\text{Present}\ X_{[\text{nom}_j]}\ \text{dance}[\text{pres}_j] \in t}}{X_{[\text{nom}_j]}\ \text{dance}_{\text{pres}_j} \in t}}{\text{dance}_{\text{pres}_j} \in (e_{\text{nom}_j}, t)}$$

5.2.2. TRANSITIVE VERBS AS NOMINATIVE ASSIGNERS. The next derivation proves that $\text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, (e_{\text{nom}_j}, t))$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Present} \in (t_{\text{nom}_j,\, \text{pres}_j,\ t}) \qquad \dfrac{Y \in e \qquad \dfrac{\text{love}[i] \in (e_{\text{acc}_i}, (e, t)) \qquad X \in e}{\text{love}[i]\ X_{[\text{acc}_i]} \in (e, t)}}{Y\ \text{love}[i]\ X_{[\text{acc}_i]} \in t}}{\text{Present}\ Y_{[\text{nom}_j]}\ \text{love}[\text{pres}_j, i]\ X_{[\text{acc}_i]} \in t}}{Y_{[\text{nom}_j]}\ \text{love}[\text{pres}_j, i]\ X_{[\text{acc}_i]} \in t}}{Y_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ X_{[\text{acc}_i]} \in t}}{\text{love}_{\text{pres}_j}[i]\ X_{[\text{acc}_i]} \in (e_{\text{nom}_j}, t)}}{\text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, (e_{\text{nom}_j}, t))}$$

One might interpret the result of these derivations as showing the advantages of having a categorial grammar in which grammatical information flows from semantical types into strings and vice versa: two conflicting theories about case assignment are seen to be closely related.

5.3. THE READINGS OF *EVERY MAN LOVES A WOMAN*. Second, we show the effect of the filters on the LG derivations (6)-(9). We pointed out previously that unconstrained LG, like LP, generates the following four readings of the string *every man loves a woman*::

(a) $\exists y\ (\text{woman}(y) \wedge \forall x\ (\text{man}(x) \rightarrow \text{love}(y, x)))$.
(b) $\exists y\ (\text{woman}(y) \wedge \forall x\ (\text{man}(x) \rightarrow \text{love}(x, y)))$.
(c) $\forall x\ (\text{man}(x) \rightarrow \exists y\ (\text{woman}(y) \wedge \text{love}(y, x)))$.
(d) $\forall x\ (\text{man}(x) \rightarrow \exists y\ (\text{woman}(y) \wedge \text{love}(x, y)))$.

However, we intend to show here that constrained Lambek Grammar enables us to exclude (a) and (c) as admissible readings.[10] The remainder of this report deals with the elimination of these spurious readings generated both by LP and unconstrained LG. This is one of the facts we wanted constrained LG to account for.

5.3.1. ELIMINATING (a). The marked LG derivation which generates this reading is the following one :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Y \in e \qquad \text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, (e_{\text{nom}_j}, t))}{Y_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i] \in (e_{\text{nom}_j}, t)} \qquad X \in e}{\dfrac{Y_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i]\ X_{[\text{nom}_j]} \in t}{\text{love}_{\text{pres}_j}[i]\ X_{[\text{nom}_j]} \in (e_{\text{acc}_i}, t)}} \qquad \text{every man} \in (e, t), t}{\text{every man}_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i]\ X_{[\text{nom}_j]} \in t}}{\text{every man}_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i] \in (e_{\text{nom}_j}, t)} \qquad \text{a woman} \in (e, t), t}{\text{every man}_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{nom}_j]} \in t}$$

Cf. Derivation (6).

This derivation induces the reading according to which there is a woman such that she loves every man. However, this derivation has to be filtered out since acc_i precedes *[i]* in the resulting string. Notice that the deletion rules can not even be applied: the subject of the conclusion has the wrong configuration.

5.3.2. ELIMINATING (c). The following derivation illustrates the way in which LG discharges the spurious reading according to which every man is loved by a woman:

$$\dfrac{\text{every man} \in (e, t), t \qquad \dfrac{\dfrac{X \in e \qquad \text{love}_{\text{pres}_j}[i] \in e_{\text{acc}_i}, (e_{\text{nom}_j}, t)}{X_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i] \in (e_{\text{nom}_j}, t)} \qquad \text{a woman} \in (e, t), t}{X_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{nom}_j]} \in t}}{\text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{nom}_j]} \in (e_{\text{acc}_i}, t)}}{\text{every man}_{[\text{acc}_i]}\ \text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{nom}_j]} \in t}$$

Cf. Derivation (8).

This derivation has to be rejected because, once again, in the resulting string the accusative marker precedes the transitive index.

The above illustration of the effects of the marking mechanism and the filters excludes the derivations we wanted to eliminate. By contrast, below we show admissible derivations related to (b) and (d).

5.3.3. GENERATING (b). The derivation below corresponds to the reading in which there is a woman loved by every man :

$$\dfrac{\dfrac{\text{every man} \in (e, t), t \qquad \dfrac{\text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, (e_{\text{nom}_j}, t)) \qquad X \in e}{\text{love}_{\text{pres}_j}[i]\ X_{[\text{acc}_i]} \in (e_{\text{nom}_j}, t)}}{\text{every man}_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ X_{[\text{acc}_i]} \in t}}{\text{every man}_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, t)} \qquad \text{a woman} \in (e, t), t}{\text{every man}_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{acc}_i]} \in t}$$
$$\text{every man loves a woman} \in t$$

Cf. Derivation (7).

5.3.2. GENERATING (d). Finally, the next derivation corresponds to the reading in which for every man there is a woman who he loves:

$$\dfrac{\dfrac{\text{every man} \in (e, t), t \qquad \dfrac{\dfrac{Y \in e \qquad \dfrac{\text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, (e_{\text{nom}_j}, t)) \qquad X \in e}{\text{love}_{\text{pres}_j}[i]\ X_{[\text{acc}_i]} \in (e, t)}}{Y_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ X_{[\text{acc}_i]} \in t}}{Y_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i] \in (e_{\text{acc}_i}, t)} \qquad \text{a woman} \in (e, t), t}{Y_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{acc}_i]} \in t}}{\text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{acc}_i]} \in (e_{\text{nom}_j}, t)}}{\text{every man}_{[\text{nom}_j]}\ \text{love}_{\text{pres}_j}[i]\ \text{a woman}_{[\text{acc}_i]} \in t}$$
$$\text{every man loves a woman} \in t$$

Cf. Derivation (9).

This concludes the presentation of arguments showing that restricted Lambek Grammar avoids the overly generosity of its unrestricted counterparts: unrestricted LG and LP itself.

6. SUMMARY. We have shown how restricted Lambek Grammar deals with sentences of the form NP TV NP. The results of the previous sections allow us to conclude that LG overcomes the semantic overgeneration of LP and unrestricted LG. As for the general form of the approach outlined here, a categorial grammar enriched with

- a mechanism regulating the flow of linguistic information

and constrained by

- linear precedence statements

turned out to be a convenient vehicle.

The theory outlined here embodies an important strategy which admits generalization. We shall conclude this report by sketching the way in which LG can be restricted to assure that the sentence *every man walks* gets only one reading. First, we demonstrate that in the unrestricted LG this sentence has two derivations. One corresponds to the natural reading of the sentence, the other corresponds to the reading in which only men dance. Next we show how LG could be modified, in *line* with the approach described in this report, to exclude the second reading.

6.1. THE NATURAL READING.

$$\dfrac{\dfrac{\text{every} \in (e, t), ((e, t), t) \qquad \text{man} \in (e, t)}{\text{every man} \in (e, t), t} \qquad \text{walks} \in (e, t)}{\text{every man walks} \in t}$$

Denotation: $\forall x\,(\text{man}(x) \rightarrow \text{walk}(x))$.

6.2. THE UNNATURAL READING.

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{every} \in (e, t), ((e, t), t) \quad X \in (e, t)}{\text{every } X \in (e, t), t} \quad \text{man} \in (e, t)}{\text{every } X \text{ man} \in t}}{\text{every man} \in (e, t), t} \quad \text{walks} \in (e, t)}{\text{every man walks} \in t}$$

Denotation: $\forall x$ (walk(x) $\rightarrow$ man(x)).

6.3. SKETCH OF A SOLUTION. Now, suppose that the basic assignment concerning the determiner *every* takes the following form:

- every[k] $\in$ $((e, t)_{cn_k}, ((e, t), t))$.

Moreover, suppose that we introduce the following very natural precedence constraint:

- NP filter

  A derivation D is unacceptable if therein cn_k does not immediately follow the determiner index *[k]*.

Then, if we use our mechanism of information flow the first of the above derivations takes the following shape:

$$\dfrac{\dfrac{\text{every[k]} \in (e, t)_{cn_k}, ((e, t), t) \quad \text{man} \in (e, t)}{\text{every[k] man}_{cn_k} \in (e, t), t} \quad \text{walks} \in (e, t)}{\text{every[k] man}_{cn_k} \text{ walks} \in t}$$

The second derivation, on the other hand, takes now the following shape:

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{every[k]} \in (e, t)_{cn_k}, ((e, t), t) \quad X \in (e, t)}{\text{every[k] } X_{cn_k} \in (e, t), t} \quad \text{man} \in (e, t)}{\text{every[k] } X_{cn_k} \text{ man} \in t}}{\text{every[k] man} \in ((e, t)_{cn_k}, t)} \quad \text{walks} \in (e, t)}{\text{every[k] man walks}_{cn_k} \in t}$$

But our NP-filter rules out the last derivation, in addition it admits the previous one. And this is the goal we wanted to meet. So, we can conclude from the foregoing considerations that the strategy described in the main part of this report go at least part way toward limiting the semantical overgeneration of undirected categorial grammars.

7. CONCLUDING REMARKS. It should have become clear thorough the foregoing sections that

(1) Lambek Grammar is an undirected categorial grammar in the format of type assignment statements.

And that

(2) this format allows us to define rules that built up grammatical and semantical information in close interaction so that we can account for syntactical and semantical facts.

In addition,

(3) This mechanism suggest several interesting logical questions

- Which kind of rules are admissible in this framework? For instance, D2 is in fact a replacement rules along the lines of Montague extended categorial grammar. Is this kind of rule necessary?
- What is the effect of the rules and the filters on the recognition power of the underlying grammar? For instance, LG has a deletion rule. But it is well-known that such a rule is responsible for the fact that transformation grammars recognize all 0-type languages. Can this rule have the same effect on Lambek Grammar?
- Recall Van Benthem's result: each LP derivation corresponds to a Λ-term and each Λ-term corresponds to an LP derivation. We have seen that each LG derivation corresponds to term in Λ. But, holds it now that such a term corresponds to a *correct* LG derivation?

Finally,

(4) this mechanism also suggest several interesting linguistical questions:

- LG contains an indexing-strategy. Are there general principles guiding this strategy? Could these indices be used to explain anaphorical phenomena along the lines of the Government and Binding theory?
- What is the relationship between LG and the directed Lambek Calculus in which grammatical relations are encoded in the direction of the type-forming operators?[11]
- What is the relationship between LG and the directed Lambek Calculus in which special operators have the effect of ensuring a certain amount of permutation?[12]

The result obtained in this report underscores the utility of constructing an information-based categorial grammar in which type combination and string generation are two sides of the same coin. The immediate goal of the research reported here has focused on overcoming the semantical inadequacy of unrestricted non-directed categorial grammars. However, a more global concern has been to emphasize the utility of a categorial grammar able to transmit information from categories into strings and vice versa. Although there are numerous details left untouched, and we have made no attempt to answer the questions listed above the theory described here proved to be fruitful and promising for further study.

---

[1]This process of information flow along categorial derivations was first introduced in van Benthem(1988). Sanchez (1991a, b) use this mechanism in the construction of a categorial natural logic.

[2] Of course, there is nothing special about these sentences. They are used in the categorial literature to illustrate the inadequacy of LP.

[3]This notation and terminology is taken over from the system of type assignment to lambda terms first described in Curry (1958).

[4] See Gazdar (1985). In connexion with linear precedence statements it is worthwhile to mention here the following fact. Van Benthem (1988) shows that LP recognizes the permutation closure of the regular language (abc)* consisting of the set of all strings with equal number of a, b, c's. But if one add to LP the linear precedence statements

a precedes b; b precedes c

then LP recognizes the context-sensitive language $\{a^n b^n c^n\}$. There is then a prima facie motivation for using precedence statements in order to restrict the overgeneration of LP.

[5] See, for instance, van Benthem(1986), van Benthem(1988), van Benthem(1991).

[6] This move represents an attempt to make the discussion of semantical overgeneration more easy but it is not necessary. For theoretical reasons van Benthem's approach is to be preferred.

[7] This is a very natural assumption to make and one with a historical pedigree. In the logical books co-authored by Hilbert, Hilbert(1929) and Hilbert(1934), we find a logical language with two kinds of individual variables: variables which appear only as bound variables and variables which appear only as free variables. In the theory under consideration here, we can think of natural language expressions as analogue to Hilbert only-free variables.

[8] It should be mentioned before turning to a detailed description of how this case is assigned that the mechanism by which this is done has no impact on the semantical inadequacy of LG. At least, it has no impact on the examples we are concerned with. However, as we shall show, it has some effect on the syntactical inadequacy of the grammar.

[9] I shall not attempt to defend this view here but simply refer the reader to Sanchez(forthcoming).

[10] In fact only one of the two filters is necessary but this is not relevant here.

[11] See Moortgat(1988).

[12] See Moortgat (1991).

REFERENCES

Curry (1958) : *Combinatory Logic*, H. B. Curry and R. Feys, North Holland, Amsterdam.

Hilbert(1928): *Grundzüge der theoretischen Logik*, D. Hilbert and W. Ackermann. Springer-Verlag, Berlin.

Hilbert (1934): *Grundlagen der Mathematik*, D. Hilbert and P. Bernays. Springer Verlag, Berlin.

Gazdar (1985) : *Generalized Phrase Structure Grammar*, G. Gazdar, E. Klein and I. Sag. Basil Blackwell, London.

Moortgat (1988) : *Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calcul*us, M. Moortgat. Foris, Dordrecht.

Moortgat (1991): 'Head and Phrases. Type Calculus for Dependency and Constituent Structure', M. Moortgat and G. Morrill. To appear in *Journal of Logic, Language and Information.*

Sanchez (1991a): *Studies on Natural Logic and Categorial Grammar*, Víctor Sánchez, dissertation, University of Amsterdam.

Sanchez (1991b): 'Categorial Grammar and Natural Reasoning', Víctor Sánchez, report LP 91-08, Institute for Language, Logic and Information, University of Amsterdam.

Sanchez (forthcoming): 'Lambek Grammar: a Vehicle for Temporal Meaning Representation', Víctor Sánchez.

van Benthem (1986) : *Essays in Logical Semantics*, J. van Benthem, D. Reidel, Dordrecht.

van Benthem (1988): 'The Lambek Calculus', J. van Benthem in *Categorial Grammar and Natural Language Structures,* R. T Oehrle, E. Bach and D. Wheeler (eds), D. Reidel, Dordrecht.

van Benthem (1991): *Language in Action. Categories, Lambdas and Dynamic Logic.* J. van Benthem, North Holland, Amsterdam.

The ITLI Prepublication Series

1986 86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem Logical Syntax
1987 87-01 Jeroen Groenendijk, Martin Stokhof Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem Polyadic quantifiers
87-05 Víctor Sánchez Valencia Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem Categorial Grammar and Type Theory
87-08 Renate Bartsch The Construction of Properties under Perspectives
87-09 Herman Hendriks Type Change in Semantics: The Scope of Quantification and Coordination
1988 LP-88-01 Michiel van Lambalgen *Logic, Semantics and Philosophy of Language:* Algorithmic Information Theory
LP-88-02 Yde Venema Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 Year Report 1987
LP-88-04 Reinhard Muskens Going partial in Montague Grammar
LP-88-05 Johan van Benthem Logical Constants across Varying Types
LP-88-06 Johan van Benthem Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe A Blissymbolics Translation Program
ML-88-01 Jaap van Oosten *Mathematical Logic and Foundations:* Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra Remarks on Intuitionism and the Philosophy of Mathematics
CT-88-01 Ming Li, Paul M.B.Vitanyi *Computation and Complexity Theory:* Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas Towards implementing RL
X-88-01 Marc Jumelet *Other prepublications:* On Solovay's Completeness Theorem
1989 LP-89-01 Johan van Benthem *Logic, Semantics and Philosophy of Language:* The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem Language in Action
LP-89-05 Johan van Benthem Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang Dependency of Belief in Distributed Systems
ML-89-01 Dick de Jongh, Albert Visser *Mathematical Logic and Foundations:* Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen The Axiomatization of Randomness
ML-89-07 Dirk Roorda Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone Provable Fixed points in $I\Delta_0+\Omega_1$
CT-89-01 Michiel H.M. Smid *Computation and Complexity Theory:* Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas Towards Functional Classification of Recursive Query Processing
X-89-01 Marianne Kalsbeek *Other Prepublications:* An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke The Modal Theory of Inequality
X-89-06 Peter van Emde Boas Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project
1990 *Logic, Semantics and Philosophy of Language*
LP-90-01 Jaap van der Does A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar
LP-90-03 Renate Bartsch Concept Formation and Concept Composition
LP-90-04 Aarne Ranta Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn Nominal Tense Logic
LP-90-06 Gennaro Chierchia The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks Flexible Montague Grammar
LP-90-09 Paul Dekker The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen Models for Discourse Markers
LP-90-11 Johan van Benthem General Dynamics
LP-90-12 Serge Lapierre A Functional Partial Semantics for Intensional Logic
LP-90-13 Zhisheng Huang Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker Existential Disclosure. Implicit Arguments in Dynamic Semantics

The ITLI Prepublication Series

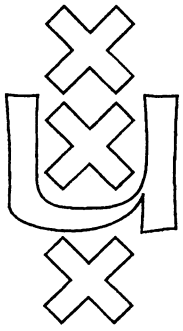
ML-90-01 Harold Schellinx *Mathematical Logic and Foundations* Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema Relational Games
ML-90-04 Maarten de Rijke Unary Interpretability Logic
ML-90-05 Domenico Zambella Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen Randomness in Set Theory
ML-90-11 Paul C. Gilmore The Consistency of an Extended NaDSet
CT-90-01 John Tromp, Peter van Emde Boas *Computation and Complexity Theory* Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas Physiological Modelling using RL
CT-90-09 Roel de Vrijer Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting
X-90-01 A.S. Troelstra *Other Prepublications* Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04 Annual Report 1989
X-90-05 Valentin Shehtman Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra Lectures on Linear Logic
1991 LP-91-01 Wiebe van der Hoek, Maarten de Rijke *Logic, Semantics and Philosophy of Langauge* Generalized Quantifiers and Modal Logic
LP-91-02 Frank Veltman Defaults in Update Semantics
LP-91-03 Willem Groeneveld Dynamic Semantics and Circular Propositions
LP-91-04 Makoto Kanazawa The Lambek Calculus enriched with additional Connectives
LP-91-05 Zhisheng Huang, Peter van Emde Boas The Schoenmakers Paradox: Its Solution in a Belief Dependence Framework
LP-91-06 Zhisheng Huang, Peter van Emde Boas Belief Dependence, Revision and Persistence
LP-91-07 Henk Verkuyl, Jaap van der Does The Semantics of Plural Noun Phrases
LP-91-08 Víctor Sánchez Valencia Categorial Grammar and Natural Reasoning
LP-91-09 Arthur Nieuwendijk Semantics and Comparative Logic
LP-91-10 Johan van Benthem Logic and the Flow of Information
ML-91-01 Yde Venema *Mathematical Logic and Foundations* Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra History of Constructivism in the Twentieth Century
ML-91-06 Inge Bethke Finite Type Structures within Combinatory Algebras
ML-91-07 Yde Venema Modal Derivation Rules
ML-91-08 Inge Bethke Going Stable in Graph Models
ML-91-09 V.Yu. Shavrukov A Note on the Diagonalizable Algebras of PA and ZF
ML-91-10 Maarten de Rijke, Yde Venema Sahlqvist's Theorem for Boolean Algebras with Operators
ML-91-11 Rineke Verbrugge Feasible Interpretability
ML-91-12 Johan van Benthem Modal Frame Classes, revisited
CT-91-01 Ming Li, Paul M.B. Vitányi *Computation and Complexity Theory* Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitányi How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitányi Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan Census Techniques on Relativized Space Classes
CT-91-07 Karen L. Kwast The Incomplete Database
CT-91-08 Kees Doets Levationis Laus
CT-91-09 Ming Li, Paul M.B. Vitányi Combinatorial Properties of Finite Sequences with high Kolmogorov Complexity
CT-91-10 John Tromp, Paul Vitányi A Randomized Algorithm for Two-Process Wait-Free Test-and-Set
CT-91-11 Lane A. Hemachandra, Edith Spaan Quasi-Injective Reductions
CT-91-12 Krzysztof R. Apt, Dino Pedreschi Reasoning about Termination of Prolog Programs
CL-91-01 J.C. Scholtes *Computational Linguistics* Kohonen Feature Maps in Natural Language Processing
CL-91-02 J.C. Scholtes Neural Nets and their Relevance for Information Retrieval
CL-91-03 Hub Prüst, Remko Scha, Martin van den Berg A Formal Discourse Grammar tackling Verb Phrase Anaphora
X-91-01 Alexander Chagrov, Michael Zakharyaschev *Other Prepublications* The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem Temporal Logic
X-91-06 Annual Report 1990
X-91-07 A.S. Troelstra Lectures on Linear Logic, Errata and Supplement
X-91-08 Giorgie Dzhaparidze Logic of Tolerance
X-91-09 L.D. Beklemishev On Bimodal Provability Logics for Π_1-axiomatized Extensions of Arithmetical Theories
X-91-10 Michiel van Lambalgen Independence, Randomness and the Axiom of Choice
X-91-11 Michael Zakharyaschev Canonical Formulas for K4. Part I: Basic Results
X-91-12 Herman Hendriks Flexibele Categoriale Syntaxis en Semantiek: de proefschriften van Frans Zwarts en Michael Moortgat
X-91-13 Max I. Kanovich The Multiplicative Fragment of Linear Logic is NP-Complete
X-91-14 Max I. Kanovich The Horn Fragment of Linear Logic is NP-Complete
X-91-15 V. Yu. Shavrukov Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic, revised version
X-91-16 V.G. Kanovei Undecidable Hypotheses in Edward Nelson's Internal Set Theory
X-91-17 Michiel van Lambalgen Independence, Randomness and the Axiom of Choice, Revised Version
X-91-18 Giovanna Cepparello New Semantics for Predicate Modal Logic: an Analysis from a standard point of view
X-91-19 Papers presented at the Provability Interpretability Arithmetic Conference, 24-31 Aug. 1991, Dept. of Phil., Utrecht University
1992 LP-92-01 Víctor Sánchez Valencia Lambek Grammar: an Information-based Categorial Grammar
ML-92-01 A.S. Troelstra Comparing the theory of Representations and Constructive Mathematics

Institute for Language, Logic and Information

LAMBEK GRAMMAR: AN INFORMATION-BASED CATEGORIAL GRAMMAR

Víctor Sánchez Valencia

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language LP-92-01



University of Amsterdam